QUALITY

ASSURANCE

GUIDELINES

Version 1.0

U.S. Department of Housing and Urban Development

June 2000

DOCUMENT CHANGE REQUEST (DCR)

Document Title:	Tracking Number:
Name of Submitting Organization:	
Organization Contact:	Phone:
	Email:
Mailing Address:	
Short Title:	Date:
Change Location: (use section #, figure #, table #, etc.)	
Proposed change:	
Rationale for Change:	

RECORD OF CHANGES

Change Number	Date	Number of Figure, Table or Paragraph	Status of Changes*	Title or Brief Description	Change Request Number
Rev. 0	5/1/00	25 Tables, 2 Figures		Quality Assurance Guidelines	
Rev. 1	6/26/00	Same as above	M	Minor changes from Office of Administration	
Rev. 2	7/5/00	Same as above	M	Minor changes from SEO&PMD	

* **Status of Changes:** **A** - ADDED **M** - MODIFIED **D** - DELETED

QUALITY ASSURANCE GUIDELINES

TABLE OF CONTENTS

QUALITY ASSURANCE GUIDELINES

LIST OF TABLES

QUALITY ASSURANCE GUIDELINES

LIST OF FIGURES

Page #

1.0 PURPOSE

The purpose of these guidelines is to define the Quality Assurance (QA) organization, tasks and responsibilities; provide reference documents and guidelines to perform the QA activities; provide the standards, practices and conventions used in carrying out QA activities; and provide tools, techniques, and methodologies to support QA activities and reporting.

1.1 Scope

These guidelines describe the QA function, show that the QA group has an independent reporting channel, and relate QA to a project's software engineering group and related groups including Configuration Management (CM).

The goal of the QA program is to verify that all software and documentation to be delivered meet all technical requirements and that a repeatable, measurable process has been followed in their creation. QA procedures defined herein may be used to examine all deliverable software and documentation, as well as determine compliance with technical and performance requirements and HUD's System Development Methodology (SDM). QA will assist in tailoring these guidelines for development of a project's Quality Assurance Plan (QA Plan) to meet the project's size and scope requirements.

1.2 Document Overview

These guidelines identify organizations and procedures to be used to perform activities related to HUD's quality assurance program.

Section 1 summarizes the purpose and contents of the Guidelines and describes the relationship of QA to management plans and other organizations.

Section 2 lists documents referenced in these guidelines.

Section 3 describes major elements of the QA organization, tasks involved in establishing and maintaining the QA function, and resource and organizational roles and responsibilities.

Section 4 describes the product review process.

Section 5 describes reviews and audits.

Section 6 identifies metrics.

Section 7 describes QA participation in testing.

Section 8 describes problem reporting and corrective action.

Section 9 describes QA's activities in the evaluation of project control.
The Appendix provides a list of acronyms and glossary of terms.

1.3 Relationship to Configuration Management Plans

QA evaluation of the software development processes throughout the lifecycle is based on the processes defined in the SDM. These guidelines provide a baseline QA Plan and work in conjunction with a division, office, or project Configuration Management (CM) Plan.

2.0 REFERENCE DOCUMENTS

This section lists documents referenced or used as a source for these guidelines.

2.1 Government Documents

a) Department of Housing and Urban Development (HUD) System Development Methodology (SDM), Release 6.01

b) HUD SDM Documentation Standards (Handbook 2400.15)

2.2 Other Documents

a) IEEE Standard for Software Quality Assurance Plans, IEEE-Standard-730-1998, 25 June 1998

b) IEEE Guide for Software Quality Assurance Planning, IEEE-Standard-730.1-1995, 12 December 1995

c) CMU/SEI-94-HB-01, Carnegie-Mellon University Software Engineering Institute, A Software Process Framework for the SEI Capability Maturity Model (CMM), September 1994

d) CMU/SEI-93-TR-24, Capability Maturity Model for Software, Version 1.1, February 1993

e) IEEE Standard 1298/A3563.1, Software Quality Management System

f) Industry Implementation of International Standard ISO/IEC 12207:1995 Standard for Information Technology – Software lifecycle processes, IEEE Standard 12207, March 1998

3.0 MANAGEMENT

This section describes major elements of the organization that influence the quality of the software.

3.1 Organization

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

Figure 3-1 shows where QA functions in the overall organization.

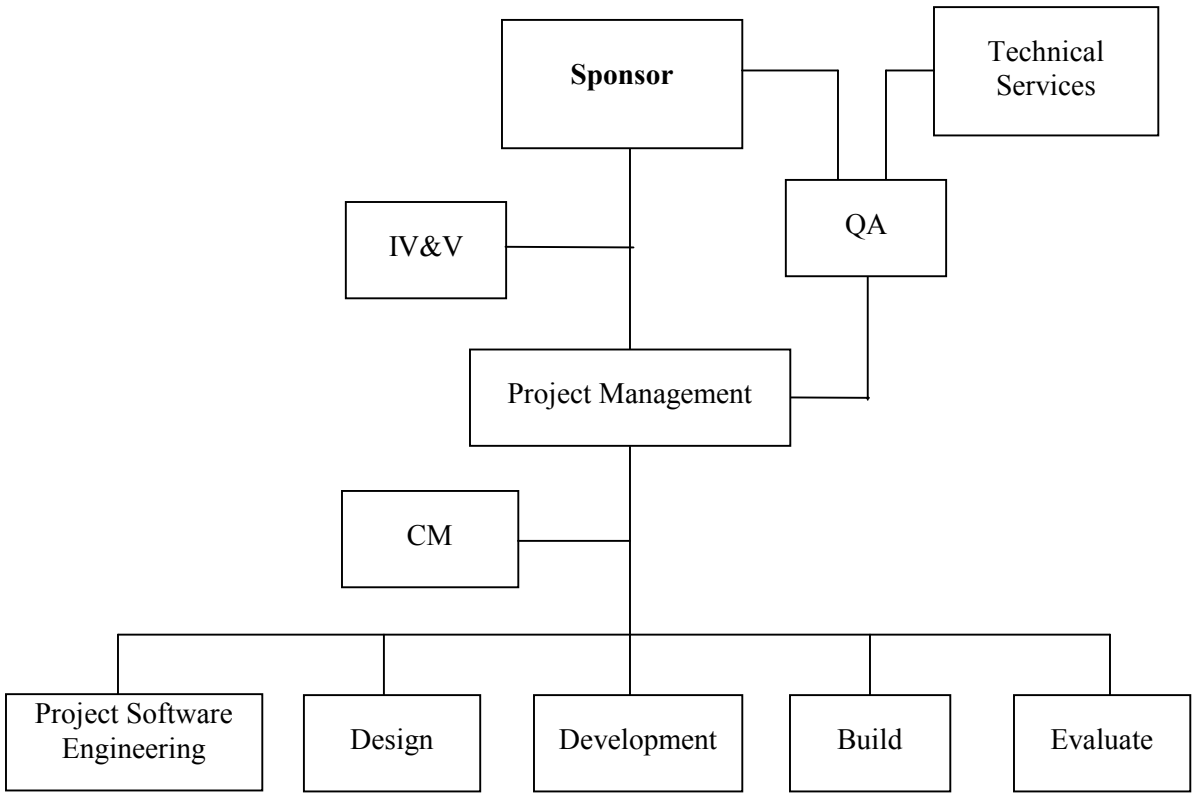

Figure 3.0-1. QA Organizational Relationship

QA is responsible for ensuring compliance with QA requirements as delineated in these guidelines. The QA organization assures the quality of deliverable software and its documentation, non-deliverable software, and the engineering processes used to produce software. QA may monitor project staff activities and review products for compliance to applicable standards, procedures, and the SDM. The results of QA monitoring and analysis, along with QA's recommendations for corrective action, may be reported to the Project Manager, and, as required, to responsible parties above the project level. All

documents and software approved by the Project Manager for release should have been reviewed and approved by QA personnel.

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

a) Sponsor may be responsible for:

(1) Establishing a quality program by committing the project team to implement the SDM and the QA Guidelines.

(2) Reviewing and approving the project's Quality Assurance Plan (QA Plan).

(3) Resolving and following-up on any quality issues raised by QA personnel.

(4) Identifying an individual or group independent from the project to audit and report on the project's QA function.

(5) Identifying the quality factors to be implemented in the system and in the software.

b) Technical Services may be responsible for:

(1) Establishing and maintaining QA guidelines.

(2) Establishing and maintaining a QA group.

(3) Assigning a QA individual to a project.

(4) Resolving and following up on any organization-level quality issues raised by a project team.

(5) Providing direct reporting direct lines of communication to the QA staff.

c) Independent Verification and Validation (IV&V) may be responsible for:

(1) Reviewing and commenting on the project's QA Plan.

(2) Implementing the quality program in accordance with the project's QA Plan.

(3) Resolving and following-up on any quality issues raised by QA staff.

(4) Verifying the quality factors are implemented in the system software, data, and hardware.

(5) Implementing the practices, processes, and procedures as defined for IV&V in the SDM and related project planning documents.

d) Quality Assurance may be responsible for:

(1) Keeping the SDM and QA Guidelines current with technology and state-of-the-practice development approaches.

(2) Maintaining the SDM template.

(3) Ensuring QA training availability.

(4) Monitoring project staff activities and processes for compliance with standards and policies.

(5) Reviewing products for compliance with standards.

(6) Providing assistance in software process engineering and software process improvement.

(7) Determining and reporting corrective actions to project management and evaluating corrective actions taken.

e) Project Management may be responsible for:

(1) Writing QA Plan

(2) Implementing the quality program in accordance with the SDM and the QA Guidelines.

(3) Identifying the QA activities to be performed by QA staff.

(4) Reviewing and approving the project's QA Plan.

(5) Identifying and funding an individual or group independent of the project to perform the project's QA functions.

(6) Resolving and following-up on any quality issues raised by QA.

(7) Identifying and ensuring that the quality factors to be implemented in the system and software are fulfilled.

(8) Identifying, developing and maintaining planning documents such as the Project Plan, the project's Quality Assurance Plan, the project's Configuration Management Plan, Test Plans (Unit and Integration), the Independent Validation, Verification and Test Plan, and the QA guidelines.

f) Configuration Management (CM) may be responsible for:

(1) Reviewing and commenting on the project's QA Plan.

(2) Implementing the quality program in accordance with these QA guidelines.

(3) Resolving and following-up on any quality issues raised by QA staff related to the project's CM.

(4) Ensuring that the quality factors are implemented in the system.

(5) Implementing the CM practices, processes, and procedures as defined in the SDM and related project planning documents.

g) Project Software Engineering may be responsible for:

(1) Implementing the quality program in accordance with these QA guidelines.

(2) Reviewing and commenting on a project's QA Plan.

(3) Resolving and following-up on any quality issues raised by QA staff related to software development activities.

(4) Identifying and evaluating the quality factors to be implemented in a system (software and hardware).

(5) Implementing software engineering practices, processes, and procedures as defined in the SDM and related project planning documents.

h) Software Design/Development/Build may be responsible for:

(1) Reviewing and commenting on the project's QA Plan.

(2) Implementing the quality program in accordance with the project's QA Plan.

(3) Resolving and following-up on any quality issues raised by QA staff related to software design and development.

(4) Identifying and implementing the quality factors to be implemented in the software.

(5) Implementing the software design/development practices, processes, and procedures as defined in SDM and related project planning documents.

i) Software Build/Evaluate may be responsible for:

(1) Reviewing and commenting on the project's QA Plan.

(2) Implementing the quality program in accordance with these QA guidelines.

(3) Resolving and following-up on any quality issues raised by QA staff related to testing.

(4) Verifying that the quality factors are implemented in the system's software, data, and hardware.

(5) Implementing test practices, processes, and procedures as defined in the SDM and related project planning documents.

3.2 Resources

3.2.1 Facilities and Equipment

QA staff will have access to facilities, equipment, computer resources, personnel resources, documentation, etc., to perform such QA functions as process and product evaluations, as well as audits.

3.2.2 Personnel

The QA Manager will be familiar with and will be able to apply QA standards. Additionally, the QA Manager will be familiar with software quality, software development related activities, and structured analysis, design, coding, and testing.

Table 3-1 provides a matrix that identifies required skills to perform QA tasks.

Table 3-1. QA Skills Matrix

Task	Skill Requirements
Code Reviews	Source Language, Peer Reviews
Documentation Reviews	Software Development, Documentation Standards and Guidelines, Peer Reviews
Process Audits	Software Development Lifecycle Processes, Audit Techniques
Testing	Testing Methodologies
QA Management	Project Management
Metrics	Data Collection and Analysis
Problem Reporting and Correction Action	Configuration Management
Tools	Vendor Supplied Training
Code, Media, and Supplier Control	Configuration Management
Risk Management and Analysis	Risk Management Process

3.3 QA Tasks and Responsibilities

The following section presents a range of tasks that cover QA activities. Depending on the scope, size, and duration of a project, a specific project may have a QA Plan developed to address the tasks applicable to the project. A small project with minor impact on HUD operations might specify fewer QA tasks, while a large project would implement a complete QA program. Projects implementing a tailored version of the QA Plan should consult the QA staff for help with deviations and waivers.

Scheduling of QA tasks is driven by the specific software development schedule. Therefore, a QA task is performed in relation to the particular software development activities that are taking place. One or more QA tasks may be performed concurrently until a task is completed. A task is considered complete when the required report e.g., QA Reports, Process Audits Reports, etc. is satisfactorily completed and action items have been closed.

The following tasks, requiring coordination and cooperation with the project team, may be performed by QA staff.. Table 3-2 presents the broad scope of QA activities and QA involvement across the project. Specific activities will be delineated in the sections following.

Table 3-2. QA Activities

Quality Assurance	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Appoint an Independent Quality Assurance Process Auditor		X						
Assist Quality Assurance Audits	X			X	X	X	X	X
Evaluate/Report Quality Assurance Audit Process	X							
Resolve Audit Findings	X	X	X					

3.3.1 Identify Standards and Guidelines

QA staff may assist the project in identifying the standards or guidelines to be followed in developing the software product. Findings will be documented and provided to the project manager.

Table 3-3 presents the broad scope of QA activities and involvement across the project in the identification and implementation of standards and guidelines.

Table 3-3. Identification of Standards and Guidelines

Identify Standards and Guidelines	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Review Standards	X							
Resolve Guidelines		X	X					
Approve Standards and Guidelines		X	X					
Implement Standards and Guidelines		X	X	X	X	X	X	X

3.3.2 Evaluate Software Tools

QA staff may conduct evaluations of tools, both existing and planned, used for software development and support. Tools are evaluated for applicability by assessing whether tool capabilities are needed for the software development or support and for adequacy by assessing whether they perform the functions for which the tools are intended. Planned tools are evaluated for feasibility by assessing whether they can be developed with the techniques and computer resources available or whether they will need to be acquired.

Documentation for the development tools evaluated will include the criteria used for the evaluation and provide the evaluation results. Any actions needed will be identified and reported to the project manager.

Table 3-4 presents the broad scope of QA activities and involvement across the project in the evaluation of software tools.

Table 3-4. Evaluate Software Tools

Evaluate Software Tools	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Evaluate Tool	X							
Resolve Audit Findings		X	X					
Approve Tool		X	X					
Integrate Tool into Environment					X			

3.3.3 Evaluate Facilities

QA staff may evaluate facilities, both existing and planned, for adequacy by assessing whether they provide the needed equipment and space used for software development and support.

QA staff will document evaluations of development facilities to include the criteria used for the evaluation and provide the evaluation results. Any actions needed will be identified and reported to the project manager.

Table 3-5 presents the broad scope of QA activities and involvement across the project in the evaluation of facilities and resolution of audit issues.

Table 3-5. Evaluate Facilities

Evaluate Facilities	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Evaluate Facilities	X							
Resolve Audit Findings		X	X					

3.3.4 Evaluate Software Products Review Process

This QA task assures that quality review processes are in place for all development 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.

QA staff may check that software products are reviewed, verify that results are reported, and that issues or problems reported are resolved in accordance with the SDM and the project's procedures.

QA staff's recommendation for corrective action requires a project management's approval and will be processed in accordance with a corrective action process.

Table 3-6 presents the broad scope of QA activities and involvement across the project in the evaluation and audit issue resolution of the software products review process.

Table 3-6. Software Products

Evaluate Software Review Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Review Products	X	X	X	X	X	X	X	X
Evaluate Review Process	X							
Resolve Audit Findings		X	X					

3.3.5 Evaluate Project Planning, Tracking, and Oversight Processes

Project planning, tracking, and oversight involves project management in developing and documenting plans as defined in the SDM and appropriate to the scope of the project. For project documents to be developed, QA staff will assist in identifying the appropriate guidelines and standards, and will assist in tailoring guidelines and standards, to meet the project's needs.

QA staff may determine whether the project conducts the relevant activities stated in the Program and Project plans. To verify that these activities are performed as planned, QA will audit the processes that

define the activity, and will use the SDM or planning document as the measure of whether those activities are being accomplished.

The results of this task will be documented using an approved process audit form and provided to project management. Any recommended changes to those plans will require update and approval by project management in accordance with the CM procedure as described in the project's CMP.

Table 3-7 presents the broad scope of QA activities and involvement across the project in the evaluation of the project planning, tracking, and oversight processes.

Table 3-7. Project Planning, Tracking, and Oversight

Project Planning & Oversight (PP&O) Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Develop/Document SDM Project-Related Plans		X	X	X				
Review Plans	X	X	X	X	X	X	X	X
Approve Plans		X	X					
Evaluate PP&O Process	X							
Resolve Audit Findings		X	X					

3.3.6 Evaluate Requirements Analysis Process

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

Requirements analysis establishes a common understanding of the requirements between sponsor, user, and software project team. An agreement with the user on the requirements for the project is established and maintained. This agreement is known as "allocating system requirements" to software and hardware.

QA staff may perform the following:

a) Verify that the correct participants are involved in the requirements definition and allocation process to ensure that all user needs are identified.

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 requirements definition and allocation 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 CM process is in place to control and manage the baselines.

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

k) Verify that agreed upon requirements are addressed in the project plan and in the requirements documents.

l) Verify that the software requirements definition and analysis process, and associated requirements reviews are conducted in accordance with the standards and procedures established by the project and as described in the SDM.

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

The results of this task may be documented using a process audit form and provided to project management. The QA staff's recommendation for corrective action requires project management's approval and will be processed in accordance with a corrective action process.
Table 3-8 presents the broad scope of QA activities and involvement across the project in the evaluation of the requirements analysis process.

Table 3-8. Requirements Analysis

Requirements Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Develop/Document Requirements					X	X	X	
Perform CM on Requirements				X				
Review Requirements	X	X	X		X	X	X	X
Approve Requirements		X	X					
Maintain Software Development Library (SDL) and Software Development Folders (SDFs)				X	X	X		
Evaluate/Report Requirements Process	X							
Resolve Audit Findings		X	X					

3.3.7 Evaluate Design Process

Design activity determines the overall structure of the system and software to be built. Based on requirements, the software is partitioned into modules, and the function(s) of each module and relationships among these modules are defined.

A goal of design is to define how the software will satisfy the allocated requirements. The level of detail of this design must be such that someone other than the original designer can accomplish the coding of the computer program.

QA may perform the following:

a) Verify that lifecycle documents and a traceability matrix are prepared and kept current and consistent.

b) Verify that relevant lifecycle documents are updated and that updates are based on approved requirements change.

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

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

e) Identify defects, verify resolution of previously 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 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.

l) 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 SDM.

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

n) 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 by its constituent parts.

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

The results of this task will be documented using a process audit form and provided to project management. The QA staff's recommendation for corrective action requires project management's approval and will be processed in accordance with a corrective action process.

Table 3-9 presents the broad scope of QA activities and involvement across the project in the evaluation of the design process.

Table 3-9. Design

Design Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Develop/Document Design					X	X	X	
Perform CM on Design				X				
Review Design	X	X	X		X	X	X	X
Approve Design		X	X					
Maintain SDL and SDFs				X		X		
Evaluate/Report Design Process	X							
Resolve Audit Findings		X	X					

3.3.8 Evaluate Code and Unit Testing Process

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

QA staff may perform the following:

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 SDM.

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

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

The results of this task may be documented using a process audit form and be provided to project management. The QA staff's recommendation for corrective action requires project management's approval and will be processed in accordance with a corrective action process.

Table 3-10 presents the broad scope of QA activities and involvement across the project in the evaluation of the code and unit testing process.

Table 3-10. Code and Unit Testing

Code & Unit Testing Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Develop/Fix Code						X		
Code Review	X					X	X	
Unit Test						X	X	
Maintain SDL and SDFs				X		X	X	
Maintain STR Process				X				
Evaluate/Report Code & Unit Test Process	X							
Resolve Audit Findings		X	X					

3.3.9 Evaluate Integration Testing Process

Software integration and test activities combine individually developed components in the development environment to verify that they work together to complete software and system functionality. Integration requires synchronization to meet designated integration and test milestones.

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

QA staff may perform the following:

a) Verify that software test activities are identified, test environments have been defined, and guidelines for testing have been designed. The QA staff may verify that the software integration process, software integration testing activities, and software performance testing activities are being performed in accordance with the SDM, the software design, the plan for software testing, and established software standards and procedures.

b) Verify that any transfer of control of code to personnel performing software is being accomplished in accordance with established software standards and procedures.

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

d) Verify that discrepancies discovered during software tests are identified, analyzed, 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 are updated based on results of software integration testing, software performance testing, and the corrective action process.

e) Verify that software performance tests produce results which 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 (Unit and Integration) and the Independent Verification, Validation, and Test Plan 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 certification or calibration of all support software or hardware used during tests.

The results of this task will be documented using a process audit form and will be provided to project management. QA's recommendation for corrective action requires project management's approval and will be processed in accordance with a corrective action process.

Table 3-11 presents the broad scope of QA activities and involvement across the project in the evaluation of the integration testing process.

Table 3-11. Integration Testing

Integration Testing Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Integrate Software (Software)						X		
Test Integrated Software							X	
Fix Errors						X		
Maintain SDL and SDFs				X		X	X	
Maintain Trouble Reporting Process				X				
Evaluate/Report Software Integration Test Process	X							
Resolve Audit Findings		X	X					

3.3.10 Evaluate Acceptance Testing Process

This activity ensures that the product is ready to be released to end users. It evaluates the "as built" system against functional, data, and performance requirements, and examines the utility of user, operations, and maintenance documentation.

Table 3-12 presents the broad scope of QA activities and involvement across the project in the evaluation of the acceptance testing process.

Table 3-12. Acceptance Testing

Acceptance Testing Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Performance Test Software						X		
Test System Functions Against Requirements	X						X	X
Fix Errors						X		
Maintain SDL and SDFs				X		X	X	
Maintain Trouble Reporting Process				X				
Evaluate/Report Software Integration Test Process	X							
Resolve Audit Findings		X	X					

3.3.11 Evaluate Release Process

This activity applies to all deliverables released into the HUD IT environment.

QA staff may evaluate activities in preparation for end-item delivery to verify that program or project requirements for functional and physical audits of the end-item products are being satisfied. The QA organization may be allowed to prohibit delivery of certain items, such as documentation, code, or a system, if the project fails to meet requirements or standards.

The results of this task will be documented using a process audit form and provided to project management. The QA staff's recommendation for corrective action requires project management's approval and will be processed in accordance with a corrective action process.

Table 3-13 presents the broad scope of QA activities and involvement across the project in the evaluation of the release process.

Table 3-13. Release

Release Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Prepare/Document Version Release Documentation				X				
Review Version Release Documentation	X				X	X	X	X
Approve Version Release Documentation			X					
Evaluate/Report End-Item Delivery Process	X							
Resolve Audit Findings		X	X					

3.3.12 Evaluate the Corrective Action Process

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

QA staff may perform the following:

a) Periodically review the corrective action processes and their results against the CMP to assess the effectiveness of the correction action process.

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

The results of this task may be documented using a process audit form and provided to project management. QA's recommendation for corrective action requires project management's approval and will be processed in accordance with a corrective action process.

Table 3-14 presents the broad scope of QA activities and involvement across the project in the evaluation of the corrective action process.

Table 3-14. Corrective Action

Corrective Action Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test

Follow Corrective Action Process	X	X	X	X	X	X	X	X
Maintain Corrective Action Process				X				
Evaluate/Report Correction Action Process	X							
Resolve Audit Findings		X	X					

3.3.13 Evaluate Media Certification Process

QA staff may verify that CM certifies media containing source code and media containing object code delivered to the procuring agency correspond to one another. QA staff may verify 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.

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

Table 3-15 presents the broad scope of QA activities and involvement across the project in the evaluation of the media certification process.

Table 3-15. Media Certification

Media Certification	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Follow Certification Process	X			X			X	X
Certify Software	X			X				
Evaluate/Report Certification Process	X							
Resolve Audit Findings		X	X					

3.3.14 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 deliverable software after delivery do not depend on the non-deliverable software, or provision is made to verify that the same software may be obtained. QA staff may 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 the acquirer can obtain the same software. QA staff may verify that all non-deliverable software used on the project performs its intended functions.

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

Table 3-16 presents the broad scope of QA activities and involvement across the project in the evaluation of the code and unit testing process.

Table 3-16. Non-deliverable Software Certification

Non-Deliverable Software Certification	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Follow Certification Process	X			X			X	X
Certify Software	X			X				
Evaluate/Report Certification Process	X							
Resolve Audit Findings		X	X					

3.3.15 Evaluate Storage and Handling Process

QA staff may verify that there is an established plan, methodology, or set of procedures for storage and handling of the media. They may evaluate 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.

The results of this task may be documented using an approved process audit form and provided to project management. QA staff's recommendation for corrective action requires project management's approval and will be processed in accordance with a corrective action process.

Table 3-17 presents the broad scope of QA activities and involvement across the project in the evaluation of the storage and handling process.

Table 3-17. Storage and Handling

Storage & Handling Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Follow Storage and Handling Process	X			X		X	X	X
Evaluate/Report Storage and Handling Process	X							
Resolve Audit Findings		X	X					

3.3.16 Evaluate Deviations and Waivers Process

QA staff may 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 CMP and approved by the approving organization.

Table 3-18 presents the broad scope of QA activities and involvement across the project in the evaluation of the deviations and waivers process.

Table 3-18. Deviations and Waivers

Deviations & Waivers	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Document Deviations & Waivers		X	X					

Recommend Approval			X					
Approve		Major	Minor					
Evaluate/Report Deviation & Waiver Process	X							
Resolve Audit Findings		X	X					

3.3.17 Evaluate Configuration Management Process

CM is the discipline that applies technical and administrative direction and surveillance in order to:

a) Identify and document functional and physical characteristics of Configuration Items (CIs),

b) Control changes to CIs and their related documentation,

c) Record and report information needed to manage CIs effectively, including status of proposed changes and the status of approved changes, and

d) Audit the CIs to verify conformance to specifications, interface control documents, and other contract requirements.

QA staff may evaluate the following:

a) Verify that configuration identification of documents, code, and computer data have established standards for titling 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 that configuration control of changes to baseline documents and software are being managed in accordance with CM requirements as stated in the CMP.

d) Verify that 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 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 CMP.

f) Verify that for document control that approved project personnel are using up-to-date documentation and 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 may also determine that control of access to software products is being properly exercised and that unauthorized changes to master files cannot occur.

The results of this task may be documented using an approved process audit form and may be provided to project management. QA's recommendation for corrective action requires project management's approval and will be processed in accordance with a corrective action process.

Table 3-19 presents the broad scope of QA activities and involvement across the project in the evaluation of the configuration management process.

Table 3-19. Configuration Management

Configuration Management Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Develop/Document CMP				X				
Review CMP	X	X	X		X	X	X	X
Approved CMP		X	X	X				
Follow CM Processes	X	X	X	X	X	X	X	X
Document CM Procedures				X				
Evaluate/Report CM Process	X							
Resolve Audit Findings		X	X					

3.3.18 Evaluate Software Development Library Control Process

The Software Development Library (SDL) functions as the main control point for software CM. An 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, as well as other technical and management material and documentation deemed necessary in the CMP. The SDL also contains previous versions of the operational software system.

QA staff may perform the following:

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 establishment of formal release procedures for CM 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.

The results of this task will be documented using an approved process audit form and will be provided to project management. QA's recommendation for corrective action requires project management's approval and will be processed in accordance with a corrective action process.

Table 3-20 presents the broad scope of QA activities and involvement across the project in the evaluation of the software development library control process.

Table 3-20. Software Development Library Control

Software Development Library Control Process	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Establish SDL				X				
Follow SDL Procedures	X		X	X	X	X	X	X

Evaluate/Report SDL Process	X							
Resolve Audit Findings		X	X					

3.3.19 Evaluate Non-developmental Software

Non-developmental Software (NDS) is software provided by the Government, a third party, or made available commercially (Commercial-off-the-shelf (COTS)). NDS may be referred to as reusable software, Government-furnished software, or commercially available software, depending on its source. QA staff may verify that non-developmental software performs its intended functions.

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

Table 3-21 presents the broad scope of QA activities and involvement across the project in the evaluation of non-developmental software.

Table 3-21. Evaluate Non-developmental Software

Evaluate Non-Developmental Software	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Evaluate Non-Development Software	X				X	X	X	X
Evaluate/Report Non-Developmental Software Process	X							
Resolve Audit Findings		X	X					
Integrate Non-Development Software					X	X	X	X
Resolve Integration Errors					X	X	X	X

3.3.20 Perform Configuration Audits

After acceptance testing for a CI release, a Functional Configuration Audit (FCA) is performed to determine if test results demonstrate that the CI meets its allocated requirements and user needs. QA staff may perform the following activities:

a) Review test and analysis results against test plans and procedures to ensure that testing is adequate and properly performed.

b) Review test and analysis results to verify that actual performance of the system complies with its requirements and that sufficient test results are available to ensure that the system will perform in its operational environment.

c) Review all waivers to ensure that they were approved.

A Physical Configuration Audit (PCA) is performed to determine if the CI's documentation is complete and consistent with the "as-built" CI. QA staff may perform the following activities:

a) Compare the tested system configuration with the operational system delivery to ensure that the appropriate components were tested.

b) Verify that the system complies with all applicable standards.

c) Ensure that system documentation accurately reflects the tested configuration.

QA staff may be required to perform or assist in a formal configuration audit in accordance with the project CMP. QA staff will perform or assist in the Function Configuration Audit (FCA) and the Physical Configuration Audit (PCA) as detailed in the CM Plan.

The results of this task may be documented using an approved process audit form and will be provided to project management. QA's recommendation for corrective action requires project management's approval and will be processed in accordance with a corrective action process.

Table 3-22 presents the broad scope of QA activities and involvement across the project in the evaluation of the performance of configuration audits.

Table 3-22. Perform Configuration Audits

Configuration Audits	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Assist/Perform Configuration Audits	X			X	X	X	X	X
Evaluate/Report Configuration Audit Process	X							
Resolve Audit Findings		X	X					

3.3.21 Evaluate Risk Management Process

Risk management provides an approach for identifying and mitigating loss, compromise, or damage to implemented systems and/or systems under development. Risks can be natural (e.g., earthquake), caused by human intervention (e.g., disgruntled employee), environmental (e.g., oil spill), facility-related (e.g., power outage), technical (e.g., insufficient bandwidth), etc. A risk assessment should be completed to determine the type and probability of a particular type of risk, and risk management and contingency plans should be in place to provide guidance as to how to mitigate risks. Risks should be periodically re-evaluated. QA staff may perform the following activities

a) Verify that a risk assessment has been performed and that is reflects the actuality of risk for the particular project and the project's environment.

b) Verify that a risk management plan is in place and is being kept current.

c) Ensure that system documentation accurately reflects the tested configuration.

The results of this task may be documented using an approved process audit form and may be provided to project management. QA's recommendation for corrective action requires project management's approval and will be processed in accordance with a corrective action process.

Table 3-23 presents the broad scope of QA activities and involvement across the project in the evaluation of the risk management process.

Table 3-23. Risk Management

Risk Management	QA Mgr	Prog Mgr	Proj Mgr	CM	Sys Eng	SW Dev	SW Test	Sys Test
Review Risk Assessment and Risk Mitigation Plans	X							
Evaluate/Report Risk Assessment and Management Process	X							
Resolve Audit Findings		X	X					

3.4 Schedule

QA schedules are closely coordinated with the software development lifecycle described in the SDM and the schedule presented in the project plan. Process audits will be performed at the beginning of each lifecycle phase 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 lifecycle phase to verify that the processes and procedures are being followed. At the completion of a software development phase, QA staff will review and report whether all steps required to transition to the next phase have been accomplished.

4.0 PRODUCT REVIEWS

All products will undergo a peer review (walkthrough or formal inspection) in accordance with the Peer review process detailed in the project's plans. The purpose of peer reviews is to remove defects from the work products early and efficiently. QA staff may review and/or audit the activities and work products for peer reviews and report the results.

The following provides an overview of the peer review process:

a) Walkthrough *(can also be called Non-Author Review or Technical Review)* - the item being reviewed is presented by the primary author in an informal setting with his or her peers. Defects noted are recorded and the author is obligated to address.

b) Formal Inspections - the item being reviewed is formally presented and discussed in a group meeting conducted by a facilitator rather than by the item's primary author. Errors discovered are rigorously recorded, categorized, and analyzed for trends. The formal inspection is performed in accordance with a formal inspection process.

The following criteria apply to a peer review:

a) Item Completeness - Determine whether the item fully meets the intended objectives.

b) Problem Identification - Identify problems as early as possible to correct them before they are compounded in subsequent project phases.

c) Compliance with Standards - Verify that the item complies with established or required standards, or that waivers are sought where it does not meet the standards.

d) Risk Identification - Identify potential risk areas in the project so that they can be managed and mitigated as the project progresses.

e) Traceability - Ensure that the item is traceable, possibly through a matrix, which will help verify that the item satisfies its requirements.

A peer review requires one the following decisions by the peer review attendees:

a) Accept the product without further modification,

b) Reject the product due to severe errors (once corrected, another review must be performed), or

c) Accept the product provisionally (minor errors have been encountered and must be corrected, but no additional review will be required).

Upon completion of a peer review, QA records and reports peer review metrics. The report consists of the item reviewed, the number of errors detected, when the peer review was conducted, the number of closed error reports, and the number of open error reports and then given to the project manager or designated representative.

Upon completion of a peer review, the product will be submitted to CM and placed under CM control. The product will then be processed in accordance with the CM product approval and release process as described in the project's CMP.

5.0 LIFECYCLE REVIEWS AND AUDITS

Table 5-1 identifies the recommended reviews and audits for the system and software development phases.

Table 5-1. Lifecycle Reviews and Audits

Development Phase	Products	Audits And Reviews
Initiation	1. Needs Statement 2. Feasibility Study 3. Cost/Benefit Analysis 4. Risk Analysis 5. Project Plan 6. Configuration Management Plan 7. Quality Assurance Plan	1. Management Review 2. Peer Review
Definition	1. Functional Requirements Document 2. Data Requirements Document 3. System Support Plan 4. System Security Plan 5. Internal Audit Plan	1. Requirements Review 2. Design Review 3. Process Audits 4. Management Review 5. Peer Review
Design	1. System/Subsystem Specification 2. Database Specification 3. Program Specification 4. Training Plan	1. Specification Review 2. Preliminary Design Review 3. Critical Design Review 4. Process Audits 5. Management Review 6. Peer Review
Build	1. Database 2. Computer Programs 3. User's Manual 4. Operations Manual 5. Maintenance Manual 6. Test Plan (Unit and Integration) 7. Verification, Validation & Test Plan 8. Training Plan 9. Installation & Conversion Plan (initial)	1. Test Readiness Review 2. Formal Qualification Review 3. Functional Configuration Audit 4. Physical Configuration Audit 5. Process Audits 6. Managerial Review 7. Peer Review
Evaluate	1. Test Results & Evaluation Reports 2. Installation & Conversion Plan (final)	1. Production Readiness Review 2. Physical Configuration Audit 3. Review Process Audits 4. Management Review 5. Peer Review
Operate	1. Pilot Test Results 2. User's Manual (updated) 3. Maintenance Manual (updated) 4. Training Manual 5. Production System	1. Process Audits 2. Management Review 3. Peer Review

5.1 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 should include persons with technical knowledge of the products to be reviewed. The purpose of the technical review is to focus on in-progress and final products, rather than materials generated especially for the review. QA staff will assure that technical reviews are accomplished and will selectively participate them in accordance with approved sampling techniques. The list below summarizes the types of technical reviews. Each type of review should be done regardless of a project's size. However, the structure and detail of the review may vary depending on a project's scope, size, duration, and impact.

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

Design Review (DR) - the objectives are to evaluate optimization, correlation, completeness, and risks associated with allocated technical requirements, and to include 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.

Specification Review (SR) - the objective is to review the finalized CI requirements and operational concept. A successful SR shows that there is a satisfactory basis for proceeding into preliminary design.

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

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

Test Readiness Review (TRR) - the objective is to determine whether the test procedures are complete and to assure that the developer is prepared for formal CI testing.

Formal Qualification Review (FQR) - the objective is to determine that a group of configuration items comprising the system are verified to have met specific program or project management performance requirements through tests, inspections, or analytical processes.

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 decision to go forward.

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

Various metrics are collected as part of technical reviews to help determine the effectiveness of the review process itself, as well as the effectiveness of process steps, that are used to produce the item being

reviewed. These metrics, reported to the project manager, may include the amount of time spent by each person involved in the review, including preparation for the review.

5.2 Management Reviews

QA staff's periodic management review of project status, progress, problems, and risk provides an independent assessment of project activities. QA provides 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 risk based on participation and evaluation of project progress and trouble areas.

Because the QA function is integral to the success of the project, QA staff should 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. These reports will be followed-up and tracked to closure.

5.3 Process Audits

Software development processes are audited according to the tasks specified in Section 3.3, QA Tasks, and performed in accordance with the software development schedule specified in the project plan or the SDM.

5.4 Configuration Audits

5.4.1 Functional Configuration Audit

The Functional Configuration Audit (FCA) is held prior to product delivery to compare the software as built (including its executable forms and available documentation) with the software requirements as stated in the baseline requirements specification. The purpose is to assure that the code addresses all, and only, the documented requirements and functional capabilities as stated in the requirements. QA staff will participate as a member of the FCA team with other FCA team members to be assigned by the project manager. QA will assist in the preparation of the FCA findings. The reported FCA findings will be monitored and tracked to closure.

5.4.2 Physical Configuration Audit

The Physical Configuration Audit (PCA) is held to verify that the product and its documentation are internally consistent and ready for delivery, as well as to assure that the documentation to be delivered is

consistent and correctly describes the code. QA staff will participate as a member of the PCA team, with other PCA team members to be assigned by the project manager and will assist in the preparation of the PCA findings. The reported PCA findings will be monitored and tracked to closure.

6.0 METRICS

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

Standards for logic structure, coding, and code comments are described in the project plan. QA will verify that the source code complies with these standards and with HUD-wide development standards.

Standards and practices for testing are described in the test plans. QA staff will verify that testing activities comply with these plans and the SDM.

6.1 Metrics

The following are examples of measurements that may be made and used to determine the cost and schedule status of the project's activities:

a) Planned milestone dates

b) Completed milestone dates

c) Planned work scheduled

d) Actual work completed

e) Planned effort expended

f) Actual effort expended

g) Planned funds expended

h) Actual funds expended

QA staff is responsible for reporting these metrics, in text or graphic form, and providing these metric reports to the Project Manager.

7.0 TEST

Testing activity includes unit level testing, integration testing (at Unit and CI level), performance testing and acceptance testing. Figure 7-1 provides a sample Test Process Flow. QA staff will audit testing activities, and may verify that the software and test documentation are subject to CM control. QA staff will witness the tests and verify that test results are recorded and evaluated. QA staff may coordinate the maintenance of trouble report logs with CM and may verify that software changes are controlled according to CM procedures. QA staff may witness all regression testing resulting from trouble reports to verify the effectiveness of the correction.

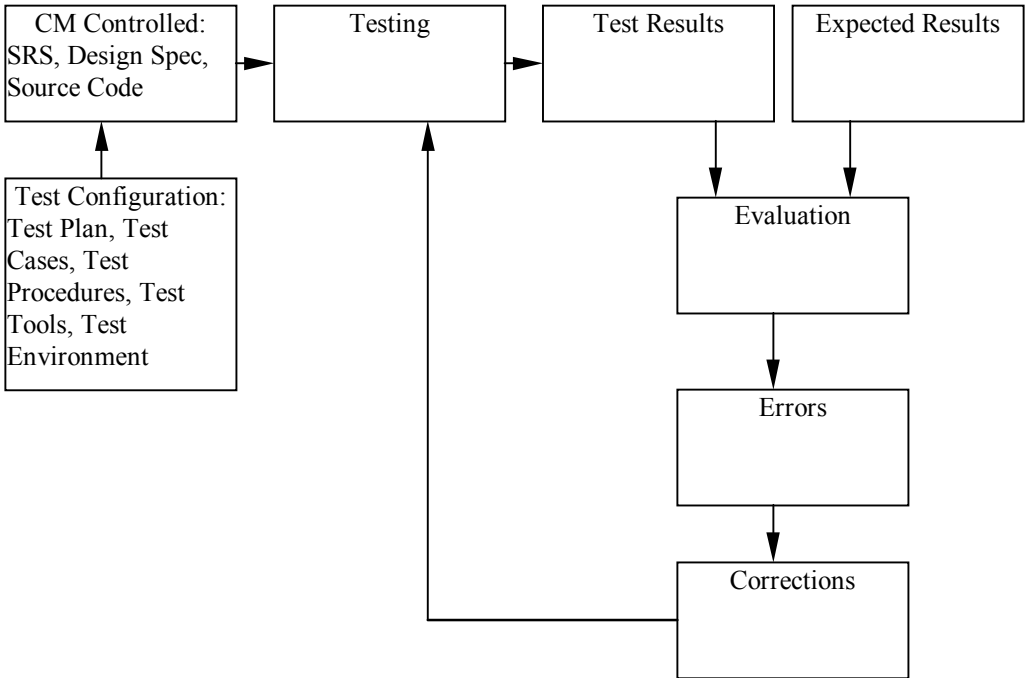

Figure 7-1. Test Process Flow Diagram

8.0 PROBLEM REPORTING AND CORRECTIVE ACTION

8.1 Process Audit Reporting

QA staff reports the results of a process audit and provides recommendations, if necessary, using a process audit report. The process audit report is used to record that the process is satisfying the following:

a) Being followed correctly and is working effectively,

b) Being followed but is not working effectively, or

c) Not being followed.

The process audit report is provided to the project manager. The report provides the project manager with insight into whether there is compliance with the development process and how effective it is 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. Additionally, the process audit report may be provided to senior management along with other project status information to guide senior management's attention to identify and mitigate project risks at the organizational level.

8.2 Problem/Change Reporting

Problems found in the software or software documentation that is under CM must be recorded by means of a problem/change report regardless of how or who discovered the problem. Problem/change reports generated by QA staff may be processed in accordance with the project's CMP. QA staff will analyze problem/change reports for problem trends in an effort to prevent recurring discrepancies, as well as report the results of problem/change report trend analyses along with suggestions for problem resolution and prevention. The format of the problem/change report may be specified in the project's CMP.

9.0 CONTROLS

9.1 Product Controls

Product control includes:

a) Identifying, labeling, and cataloging the products to be controlled.

b) Identifying the physical location of the products under control.

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

d) Distributing copies of the products, including software code.

e) Identifying documentation that is affected by a change.

f) Establishing a new version.

g) Providing user access to the products.

The control method is described in the project's CMP. QA staff will conduct ongoing evaluations of the product control process to verify that the process of control is effective and in compliance with the project's CMP.

9.2 Media Control

Media control includes:

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 media control methods and facilities are described in the project's CMP. QA staff will conduct ongoing evaluations of the media control process to verify that the process of control is effective and in compliance with the project's CMP.

9.3 Supplier Control

Prior to any purchase of software to support the development effort, QA staff and project members will define and provide complete requirements to the supplier/vendor. The software tool evaluation process will follow. Part of the evaluation process will require the supplier/vendor to describe the technical support, handling of user questions and problems, and software product upgrades.

All supplier software should be operationally tested in the target system.

9.4 Records Control

Records and reports that provide a history of product quality throughout the software lifecycle document QA activities. Metric data collected will be reviewed for trends and process improvement. All QA records will be collected and maintained in the SDL or archival storage in accordance with HUD standards and guidelines.

APPENDIX A
ACRONYMS AND GLOSSARY

APPENDIX A: ACRONYMS AND GLOSSARY

A-1 Acronyms and Abbreviations

Table 1, *Acronym and Abbreviation Definition*, lists the acronyms and abbreviations that are used in this document.

Table A-1. Acronym and Abbreviation Definition

Acronym	Abbreviation Definition
CDR	Critical Design Review
CI	Configuration Item
CM	Configuration Management
CMM	Capability Maturity Model
CMP	Configuration Management Plan
CMU	Carnegie-Mellon University
FCA	Functional Configuration Audit
FQR	Formal Qualification Review
IEEE	Institute of Electrical and Electronics Engineers
IV&V	Independent Verification and Validation
NDS	Non-Developmental Software
PCA	Physical Configuration Audit
PP&O	Project Planning and Oversight
QA	Quality Assurance
QA Plan	Quality Assurance Plan
RR	System Requirements Review
RS	Software Requirements Specification
SDF	Software Development File
SDL	Software Development Library
SDM	System Development Methodology
SDP	System Decision Paper
SDR	System Design Review
SEI	Software Engineering Institute
SEO&PMD	Systems Engineering, Oversight, and Program Management Division
SSDD	System/Subsystem Design Description
SSR	Software Specification Review
SSS	System/Subsystem Specification
SW	Software
TRR	Test Readiness Review
UDF	Unit Development Folder

A-2 Glossary of Terms

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

Design Review (DR) - the objective is to evaluate optimization, correlation, completeness, and risks associated with allocated technical requirements, as well as 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.

Formal Inspections - the item being reviewed is formally presented and discussed in a group meeting conducted by a facilitator rather than the item's primary author. Errors discovered are rigorously recorded, categorized, and analyzed for trends. The formal inspection is performed in accordance with a formal inspection process.

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

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.

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

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 decision to go forward.

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

Specification Review (SR) - the objective is to review the finalized Configuration Item (CI) requirements and operational concept. A successful SR shows that there is a satisfactory basis for proceeding into preliminary design.

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

Test Readiness Review (TRR) - the objective is to determine whether the test procedures are complete and to assure that the developer is prepared for formal CI testing.

Tools - QA 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.

Walkthrough *(can also be called **Non-Author Review** or **Technical Review***) - the item being reviewed is presented by the primary author in an informal setting with his or her peers. Defects noted are recorded and the author is obligated to address them.