SPAWAR
Systems Center
San Diego

TECHNICAL DOCUMENT 3107
March 2000

Process Control Document for the SSC San Diego Product Quality Engineering Group

R. P. Stone

SSC San Diego

ADMINISTRATIVE INFORMATION

This document was prepared for Space and Naval Warfare Systems Command by Product Quality Engineering (D4222), SSC San Diego.

Released by
M. D. Glorioso, Head
Product Quality Engineering Group

Under authority of
R. J. Jaffe, Head
Command and Intelligence
Systems Division

SB

CONTENTS

APPENDICES

Figures

Tables

1.0 PURPOSE

This Process Control Document defines the processes used within the Product Quality Engineering (PQE) Group at Space and Naval Warfare Systems Center, San Diego, CA (SSC San Diego). Initiatives within SSC San Diego (D42), as well as within SSC San Diego, have mandated progress toward Capability Maturity Model (CMM) Level 3. This requires that the PQE Group attain CMM Level 2. The primary key to reaching this level is making the processes of the PQE Group repeatable: "*The organization has achieved a stable process with a repeatable level of statistical control by initiating rigorous project management of commitments, costs, schedules and changes.*[1]" This document provides guidance within the PQE Group and for outside groups, agencies, and developers on the high-level processes that will be used by the PQE Group and all its team members. This creates an initial framework for starting a "repeatable level of statistical control by initiating rigorous project management." This document is not intended to be a testing handbook, but a guide to implement the various testing methods used in industry today.

1.1 Scope

This document includes information pertaining to testing and reporting done by the PQE Group for all its customers. With each customer comes a different level of funding and requirements; therefore, not all functions described in this document will be applicable to all customers. Customer requirements not addressed in this document may be addressed independently.

The following figure illustrates a traditional view of a system lifecycle.

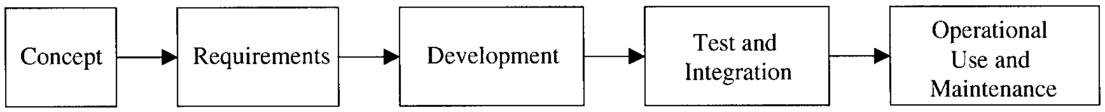


Figure 1-1. Traditional System Maintenance Lifecycle.

This figure provides a template of a system lifecycle to illustrate where the PQE efforts fit in. This diagram is not intended to show the many process concepts, only to provide an understanding of the overall process.

Traditionally, system or software test groups were only involved in the Test and Integration phase of the lifecycle. It is the intent of the PQE Group to become involved in all phases of the lifecycle since each portion of the lifecycle contains products, and all products are testable.

1. Humprey, Watts. Managing the Software Process. Addison-Wesley Publishing Company. 1990.

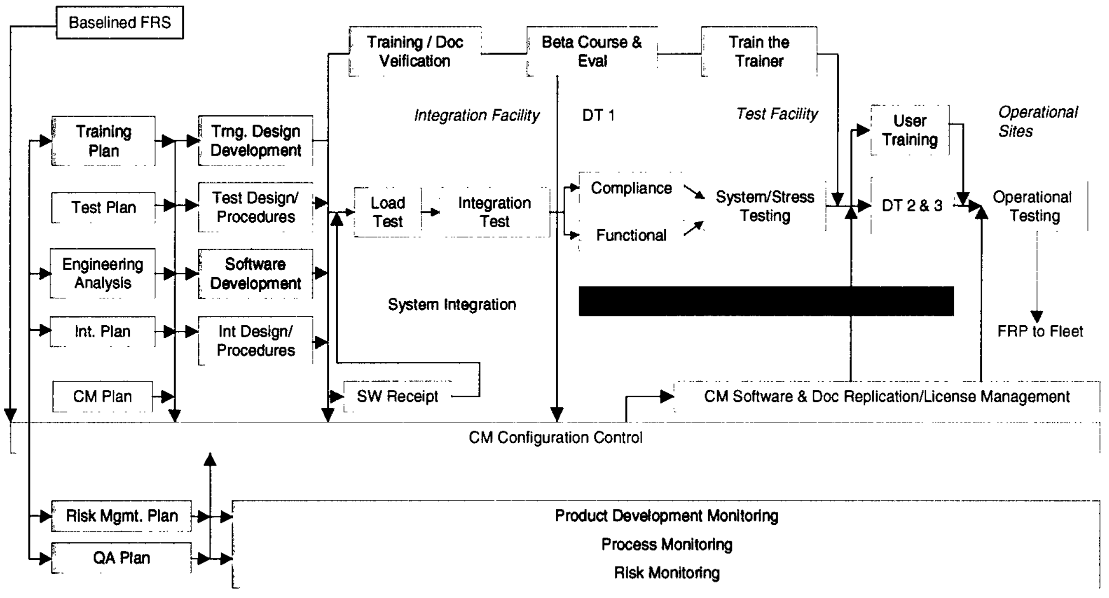


Figure 1-2. System Development Process.

Figure 1-2 depicts the process used by SSC San Diego D4222, C4I Systems, to manage the testing, configuration management (CM), integration training, and engineering. The green boxes (Test Plan, Test Procedures, Compliance, Functional, System/Stress Testing, and Developmental Test [DT] 2 and 3) indicate the PQE processes. This diagram illustrates how the PQE processes fit in with the overall process and their basic interaction in the process.

2.0 REFERENCED DOCUMENTS

This document was created in part with the use of the following referenced documents:

Humprey, Watts. *Managing the Software Process.* Addison-Wesley Publishing Company. 1990.

Hetzel, Bill. *The Complete Guide to Software Testing.* 2nd Edition. Wiley-QED. 1988.

Industry Implementation of International Standard ISO/IEC 12207: 1995, Software lifecycle processes – Lifecycle data. IEEE/EIA 12207.2-1997. April 1998.

Industry Implementation of International Standard ISO/IEC 12207: 1995, Software lifecycle processes. IEEE/EIA 12207.0-1996. March 1998.

Industry Implementation of International Standard ISO/IEC 12207: 1995, Software lifecycle processes – Implementation considerations. IEEE/EIA 12207.1-1997. April 1998.

Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification (I&RTS) Version 3.1, October 1998

Kit, Edward. *Software Testing in the Real World.* Addison-Wesley. 1997.

Sanders, Joc. *Software Quality, A Framework for Success in Software Development and Support.* Addison-Wesley. 1994.

Software Development and Documentation. MIL-STD-498. 5 December 1994.

User Interface Specifications for the Defense Information Infrastructure (DII). Version 2.0. 1 April 1996.

The Encyclopedia of Computer Science and Engineering, Second Edition, Anthony Ralston, Editor, Edwin D. Rielly, Jr., Associate Editor.

Microsoft® Bookshelf® Computer and Internet Dictionary© 1997 Microsoft Corporation.

Product Quality Engineering Document Format and Style Guide, Version 1.0, June 23, 1999.

Glenford J. Myers. *The Art of Software Testing.* John Wiley & Sons, Inc., 1979.

3.0 GENERAL INFORMATION

The information contained in this section applies to the PQE processes related to product testing. This information is only general and does not detail these processes. The processes in this section may not apply to all customers. Customer processes not addressed in this section may be addressed independently.

3.1 The Purpose of Testing

Testing is an organized process of verifying and validating that a system works as expected. It involves identifying discrepancies between the actual results and the expected results. The objective of testing is to systematically uncover errors within the software while using minimal time and effort.

3.2 Mission Statement for PQE

To engineer quality into the software products requires that the PQE Group inspect/test and remove defects from requirements, design, documentation, code, test plans, and tests. Software is intangible, and intangibility complicates the building in and monitoring of quality. The goal is to reduce this intangibility through a series of common-sense interim checkpoints, with the results of each task undergoing a review as it is completed.

The mission of the PQE Group is to ensure that software systems and products are delivered to the customer in the best possible working condition, using established standard procedures to measure defects, determine their root causes, and take action to prevent their future insertion.

3.3 Limitations of PQE Testing

The PQE Group will conduct tests in an environment that is as representative as possible of the appropriate operational system. However, because the majority of testing will be conducted in the laboratory, there may be some artificial conditions that affect test results: Examples of these artificialities include:

- Non-representative numbers of workstations and servers
- Non-representative number of simultaneous data feeds
- Non-representative network loading

Any limitations or conditions that cause the Software Test Environment (STE) to depart significantly from the intended operational environment should be identified in the "Risks" section of the Software Test Plan (STP) and Software Test Report. Industry studies have demonstrated that the use of testing as a debugging process is costly and time-consuming. To ensure software testing is conducted in a timely and efficient manner, PQE resources will not be used as debugging tool. To reduce the risk associated with testing immature software, all software delivered to the PQE Group should meet the following minimum criteria:

- Software delivered to the PQE Group shall be of such quality that it will load and execute properly.
- All defects listed in the Version Description Document (VDD) as corrected shall be validated and documented as corrected by the developer before delivery.
- Software shall be certified as meeting any required style and development specifications in the specifications, including:
    - Defense Information Infrastructure (DII) Common Operating Environment (COE) Compliance
    - DII COE Style Guide
    - JAVA Compliance
    - Microsoft Logo Compliance
    - Year 2000 (Y2K) compliance certification

Deficiencies in the aforementioned criteria should be documented in the "Risks" section of the STP and Software Test Report.

3.4 PQE Process Flow

Figure 3-1 illustrates the flow or relationship among processes in the PQE Organization. Detailed processes can be found in Section 4 and Appendix D.

This diagram does not restrict how the PQE process shall work, but is a guideline for how work is intended to progress throughout the organization. This process can vary with the needs of each PQE customer.

3.5 Defining a Product

Throughout this document, all items that are testable are referred to as "products." A product can be anything delivered that has potential defects. For all products that can be submitted to the PQE Group for testing, the PQE Group can generate defect-tracking reports. Products can be made up of one or more of the following:

- Requirements Specifications
- Functional Design Specifications
- Product Requirements
- User Manuals
- Installation Manuals
- Software Version Descriptions (SVDs) or VDDs
- Build Lists
- Test Plans
- Test Procedures
- Software Source Code
- Software Applications (executables) or Segments
- Hardware
- Firmware

Products also include both draft and final versions of the different types of products.

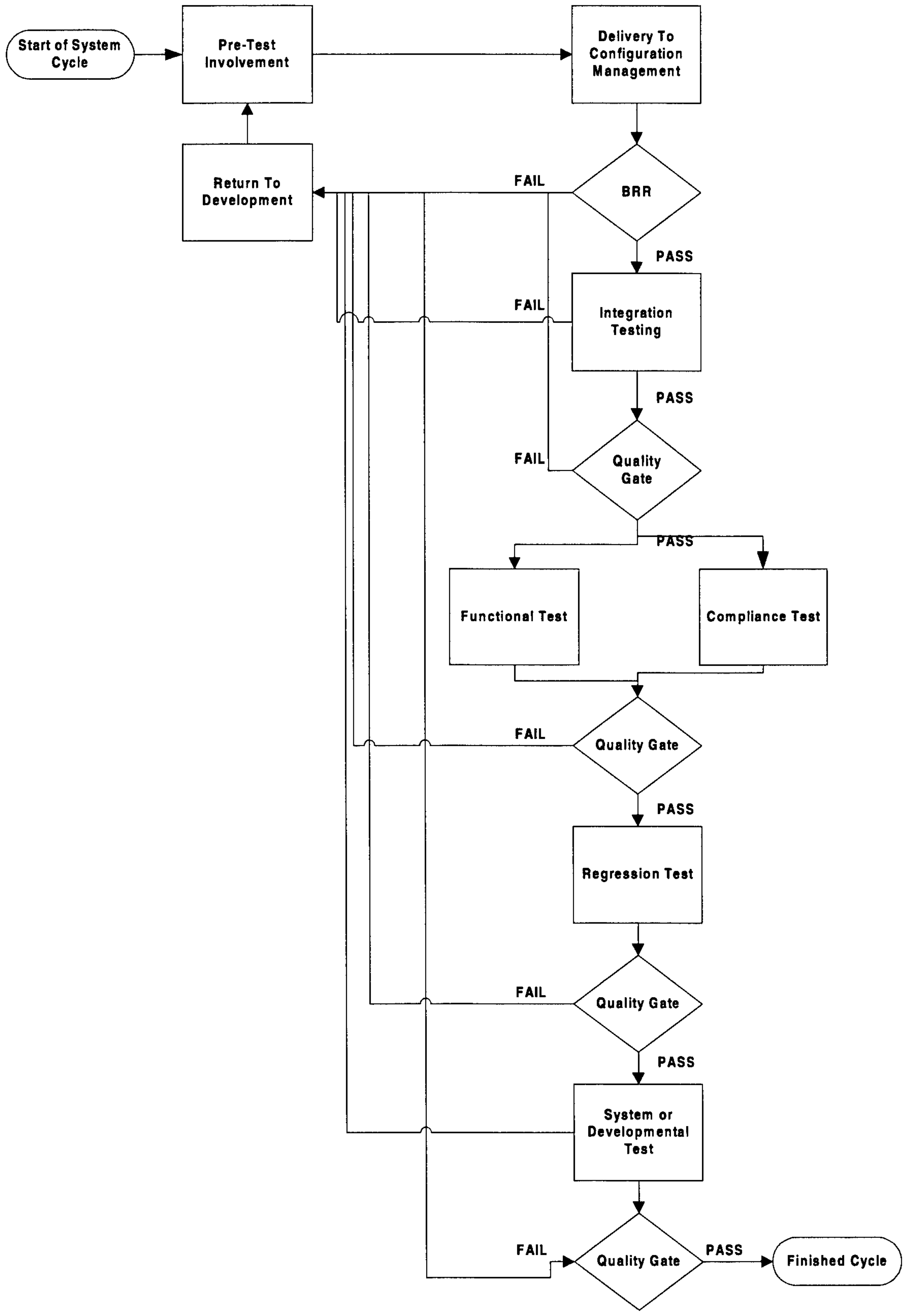


Figure 3-1. Process Overview.

3.6 Defining a Defect

Throughout this document, all problems that are found in products are referred to as "defects." There are many other terms that different groups use to identify problems (e.g., Software Trouble Report [STR], Trouble Report [TR], Global Software Problem Report [GSPR], bug). The following paragraph describes a defect as used in this document:

Software production can be seen as a series of imperfect translation processes. Each of these translations produces a work project or deliverable. Software errors [defects] are introduced when there is a failure to completely and accurately translate one representation to another, or to fully match the solution to the problem.[2]

3.7 Software Defect Tracking and Prioritization

Software defect reports shall be generated whenever defects are found, regardless of priority, and submitted to CM as soon as possible. When appropriate, the PQE Group will provide software enhancement reports. These reports do not track the defects in the system, but are used to input suggestions on improving the product or its capabilities.

Defect Prioritization shall be in accord with the latest approved standard for Department of Defense use. The current standard is IEEE/EIA 12207.0-1996. Table 3-1 problem classification priorities from that document[3]:

2. Edward, Kit. *Software Testing in the Real World – Improving the Process.* Addison-Wesley. 1997.

3. *Industry Implementation of International Standard ISO/IEC 12207: 1995, Software lifecycle processes – Lifecycle data.* IEEE/EIA 12207.2-1997. April 1998.

Table 3-1. Problem Classification Priorities.

Priority	Applies if a problem could:
1	a. Prevent the accomplishment of an essential capability b. Jeopardize safety, security, or other requirement designated "critical"
2	a. Adversely affect the accomplishment of an essential capability and no work-around solution is known b. Adversely affect technical, cost, or schedule risks to the project or to lifecycle support of the system, and no work-around solution is known
3	a. Adversely affect the accomplishment of an essential capability but a work-around solution is known b. Adversely affect technical, cost, or schedule risks to the project or to lifecycle support of the system, but a work-around solution is known
4	a. Result in user/operator inconvenience or annoyance, but does not affect a required operational or mission essential capability b. Result in inconvenience or annoyance for development or support personnel, but does not prevent the accomplishment of those responsibilities
5	Any other effect

3.8 Certification/Recommendation of Software Readiness

Each component of PQE as illustrated in figure 3-1 (Integration Testing, Functional Testing, Compliance Testing, and Operational Testing) shall provide a recommendation/certification for suitability to move to the next phase of its lifecycle. This recommendation will be provided in a required form or, if no form is specified, within the test report for the software tested. The basis for the recommendation is as follows:

- If the software does not meet the minimum requirements (i.e., does not load, or VDD is incorrect [e.g., problems listed as corrected were not]), it shall not be recommended as suitable for release until an updated VDD and/or updated software is received by PQE via CM.
- During any phase of testing, software defects that result in a Priority 1 or Priority 2 defect may result in a recommendation as not suitable for operational use or testing at an operational site.
- A Priority 1 problem may result in a recommendation as unsuitable for further distribution or testing if the problem results in an unstable environment (memory leaks, frequent crashes or halts, or database contamination/corruption).
- Software that does not meet a minimum of DII COE Compliance (normally Level 5 for legacy software, and Level 7 for new software) certification level will also receive a negative recommendation for further testing. For further details on

compliance, refer to the current version of the Integration and Runtime Specifications (I&RTS).

The intent of these recommendations is to save the Program Managers (PMs) money and permit the PQE Group to focus its resources on other efforts. Software that does not meet the minimum standards (Compliance Level 5/7, adequate documentation, and stability) indicates that the developer did not perform adequate internal testing before delivery and should be returned to the developer for further testing, debugging, and/or coding changes.

Even if the PQE Group provides a recommendation to halt testing, it will be up to the responsible PM and the PQE Test Manager to weigh issues such as safety, security, schedule, costs, or critical operational needs to determine if testing or delivery must continue regardless of the software. It is not the function of the PQE Group to decide to halt software deliveries. Halting a delivery is a business decision that is the responsibility of the PM.

3.9 Configuration Management

Throughout this document, references to internal PQE and external documents are listed. For each of these documents, the responsible PM to must provide a CM function to track, control, and distribute these documents. To ensure accurate information, the CM function must provide control numbers on all documents, software, and other related items. The CM team must also provide a defect tracking system to control, track, and provide metrics on defects. Metrics should include industry standard metrics items, such as the following:

- Number of defects by priority
- Number of defects by component
- Number of days to close a defect
- Number of defects by developer

3.9.1 Software Development Folders

In most PQE processes, the required inputs have Software Development Folders (SDFs) as the first item. The PQE Group uses SDFs for multiple purposes, including:

- Providing documentation for loading, testing, and training
- Ensuring that proper developer unit testing has been performed on the product, but may also include limitations in applications
- Ensuring that all required hardware and software products are available for testing

Availability of these materials is critical to successful and timely execution of the PQE processes. Although testing can still occur if some of these items are not available, their absence will significantly reduce testing effectiveness. SDFs consist of, but are not limited to:

Critical Items

- Applications software—all application software that is required to install and operate the system.
- Tools essential to the program—this includes compilers, libraries, debuggers, or other items necessary to run or use the applications.
- Commercial off-the-shelf (COTS) software—applications that were purchased and utilized to support the program or are otherwise necessary to fully exercise the software. If the software is not deliverable to the Government for reasons such as licensing restrictions or the Government already has the required software, this will be documented either separately or in the VDD.
- System Administrator's Manuals (SAMs) —documentation that describes how the underlying system must be configured, customized, modified, and maintained by the system's administrator. Installation procedures and any software dependencies must also be included. This document should also include appropriate troubleshooting instructions as necessary. The SAM will contain any hardware-specific information (i.e., central processing unit [CPU], memory, hard drive space, etc.) or software dependencies required to load or operate the software.
- Operator's Manual (OM) —documentation that describes how the system functions from the operator's perspective.
- VDDs— documents that describe a version of software, including any enhancements or changes that were made from previous versions, with a description of the implementation.
- Request/Certification for Deviation & Waiver—Used for any items that are delivered that do not meet specifications or requirements, the developer must provide a request to permit the discrepancy. If the deviation and waiver have already been granted, then the certification shall be provided. Requests are normally submitted for items like security and compliance issues.
- Compliance Certification—certification that the software has passed any necessary compliance requirements (e.g., Y2K, JAVA, MS logo certifications).
- Test Plan—a document describing the process used by the developer for testing the software to ensure it meets the appropriate requirements. The preferred format is IEEE 829.
- Test Report—a document reporting the results of the testing and making recommendations, if any, for software improvement. All defects found as a result of the testing shall be included. The preferred format is the IEEE 12207 series.
- Open defect reports— trouble reports that were not fixed by the newer version of the software or that were written on the delivered software. This may be submitted as part of other documents, such as the VDD or the Test Report.
- Test cases and data—the test scripts and accompanying test files that were used during execution of testing by the developer that form the basis of procedures for testing the software's functionality. If no changes were made to the test suite since the last delivery, or only minor modifications (less than 10%), then only the modified files need to be re-submitted.

Optional Items

- Statements of functionality—short documentation (normally one page) that describes the functional changes since the last version of the software. This may be mandatory depending on the PM.
- System generation software—all scripts or programs necessary to compile the source code.
- Engineering Change Proposals (ECPs)—proposals for making changes to the software as required or necessary.
- Program source—a computer program written in a language one or more steps removed from the machine language of a given computer[4]
- Object code—the output of a translating program, such as an assembler or a compiler, which converts a source program written in one language into another language, such as machine language that can be executed on a given computer[5]
- Interface Requirements Documents (IRDs)/Interface Design Specifications (IDSs) —documentation listing the requirements and specifications for the software's interface with other applications and/or hardware used to develop the software
- Data Base Design Document (DBDD)—all documents that describe the database schema.
- Quality Assurance (QA) Plan—documentation detailing how the developer ensured that quality methods were used during the design, development, and testing of the software.
- Configuration Management Plan—the plan used by the developer to maintain control of all information, such as documentation and source code, during the system development.

Each PQE customer has different requirements; therefore, documents may be substituted, combined, or deleted with the agreement of the PQE Manager and the PM.

3.10 Quality Gates or Entry/Exit Criteria

Quality Gates, also known as Entry and Exit Criteria, are checkpoints in the development cycle that provide a processing juncture at which the normal operation of a program or system is momentarily suspended to determine its environmental status.[6]. As a result of the processes contained in this document, all stages of software development have Quality Gates that are implemented and easily monitored. In all of the processes contained in this document, there are entry/exit criteria for required "inputs" and "outputs." These criteria serve as a Quality Gate by instituting a point in the testing processes that is controlled and monitored before proceeding to another phase of testing.

4. From the *Encyclopedia of Computer Science and Engineering*, Second Edition, Anthony Ralston, Editor, Edwin D. Rielly, Jr., Associate Editor.
5. Ibid.
6. *Microsoft® Bookshelf® Computer and Internet Dictionary*© 1997 Microsoft Corporation. All rights reserved.

3.11 Software Test Environment

This paragraph and the following subparagraphs deal with identification and management of the STE. The STE is a major component of scope and limitations, so it must be accurately documented in the Software Test Report. How closely the test site conforms to the software development plan (SDP) and STP, and how closely the SDP and STP conform to the end-user's environment, affect the validity of the conclusions drawn from testing. Deviations from the planned STE should be discussed in the "Risks" section of the STP and Software Test Report.

3.11.1 Defining the STE

The STE consists of the software application, operating system, system hardware, software and hardware configurations, firmware, type and number of data sources, network configuration and loading, and other factors. STE hardware items used during testing are defined in the SDP. The STE will use current versions of operating systems and DII COE (where applicable) on all installed computer hardware. When the application is expected to be used on multiple versions of the OS or DII COE, each version should be tested. Failure to test in all expected operating environments constitutes a risk that should be identified in the "Risks" section of the STP and Software Test Report.

3.11.2 Managing the STE

From a quality assurance perspective, the STE requires active management by the PQE test team. Any deviation from the planned environment should be conspicuously documented. Overall responsibility for managing the STE lies with the PQE Test Director, who will ensure that each required STE component is properly installed and will verify the component operation prior to the start of testing. Standard configuration management practices, as defined in the Software Configuration Management Plan (SCMP), will be followed to control and maintain each item of the STE. The PQE Test Director is supported by Software Engineers, Hardware Technicians, Software Quality Assurance (SQA) Observers, Software Configuration Management (SCM) Staff, and Software Test Engineers and Technicians.

3.11.3 Responsibilities

The STP identifies the tasks of the test team and other personnel designated to conduct and support application software testing. SSC San Diego will identify a customer representative to witness formal testing at the TRR, if desired. Any given test evolution may require the cooperation and support of organizations external to the PQE Group. It is beyond the purview of this document to direct tasking outside of the PQE Group, and there may be valid reasons that prevent support from other organizations. Lack of required support constitutes a risk to the test effort, and should be documented in the "Risks" section of the STP and Software Test Report. Unless otherwise specified by the PQE Test Director, responsibilities will be assigned as follows:

- SCM staff will provide the test team with the Development Baseline updates and the Product Baselines, as appropriate, and maintain copies of the STD, test logs, artifacts from test preparations, TRR documentation, and configuration-controlled project documents (SRS, SVD, and SUM) to support formal application software testing.

- Hardware technicians will configure and verify proper operation of the STE hardware items; observe operation of the STE; and provide hardware maintenance support, as appropriate, during dry run and formal testing.
- Software engineers may assist the Test Engineers in observing the operation of the application software, collecting software performance data, analyzing software failures, and providing software maintenance support, as appropriate, during dry run and formal testing.
- SQA observers will witness dry run and formal application software testing for compliance with the procedures defined in the STDs and verify application software performance to the allocated requirements. SQA will verify the correct STE hardware and software configuration before commencing formal application software testing.
- The Test Engineers and Technicians will install the application software, record the specific conditions of the STE in a log before the start of each test (this log will be used to verify the proper STE configuration and operation at the start of each test event), conduct the test; record the test results, and draft the final report.

3.12 Test Reporting Formats

SSC San Diego generates three types of test reports to detail the results of performed tests: (1) a Full Report, (2) a Condensed Report, or (3) a Test Progress Report. It shall be at the discretion of the PQE Manager to decide which report format to use unless otherwise dictated by the responsible PM. The following are default guidelines:

- Reports of large full functional tests, such as for new releases, shall use a full Test Report.
- All tests of patches to existing operational systems shall be reported using a Condensed Report.
- All tests that validate critical operational failures reported by the users of an operational system shall be documented in a Condensed Report.
- New system functionality that may go to a user site for Beta testing will be documented with a Condensed Report.
- Test Progress Reports are used during long testing cycles to report the status of testing events.
- Software tested that is not one of the above items, or otherwise included in a build plan, will be tested and the appropriate defect tracking reports will be closed or new ones generated; no report will be generated.

3.12.1 Full Report

A Full Report contains all the information gathered during testing. The report is derived from the IEEE 12207 Series. It shall contain all of the following items:

- Executive Summary, which includes a component description, top-level requirements, analyst's summary comments, list of any new STRs and/or SCPs, and a point of contact for more information on the testing discussed in the report
- Introduction
- Referenced Documents
- List and Description of Tested Segments
- Software Test Environment
- Overview of Test Results, including an overall assessment, impact, if any, of the test environment, and recommended improvements
- Detailed Test Results
- Test Log
- List of Major Risks, including the impact, limitations, and mitigation of each risk
- Conclusion
- Any Notes
- Test Procedures (in an appendix)
- Any Additional Data

3.12.2 Condensed Report

A Condensed Report is a shortened version of a Test Report. It uses the foundation of the Full Report; however, some detail is removed to expedite the report and enable effective use of testing resources. A Condensed Report is shall include the following items:

- Introduction
- Software Test Environment
- Test Objective
- Test Description (brief)
- Overview of Test Results, including an overall assessment, impact of the test environment, and recommended improvements
- Any Additional Data

3.12.3 Test Progress Report

Test progress reports are used during long testing cycles to report the status of testing events. The test director will decide on the report format. The report can be a single page breakdown or a short e-mail. Some of the following conditions would require a Test Progress Report:

- To advise of critical failures, schedule delays, safety or security concerns
- To provide interim status during an extended test event
- To provide a quick synopsis at completion of a test event beforeo delivery of the test report

3.13 Defect Validation

The Defect Validation Process validates all defects that were listed in the VDD (or other similar document) as being corrected. For each of the Testing Processes, only defects in the VDD, which apply to that phase of testing, will be tested. For example, in Integration testing, only defects that apply to Integration testing will be tested. Other defects, which are not part of that testing cycle, but are noted or tested as correct in the course of testing, will also be closed.

Every attempt shall be made to use available resources to validate that defects and/or enhancements have been correctly fixed. However, it should be understood that depending on the software dependencies, test environment limitations, and current tasking, it is possible that not all defects can be tested functionally. In such cases, PQE will close the defect as untestable by PQE unless otherwise agreed upon with the PM.

4.0 PQE PROCESSES

The following sections discuss the test processes of the PQE Group. The processes in this section may not apply to all customers. Customer processes not addressed in this section may be addressed independently.

4.1 Pre-Test Involvement Process

More than half of the errors encountered in a product are usually introduced in the requirements phase of a project. In previous system lifecycle evolutions, the testing organization (now PQE) was not included in activities, such as requirements definition and planning. These evolutions are critical to proper planning and execution of the testing processes. The involvement of the PQE Group will help ensure that testing issues are addressed and that the requirements are testable. This helps establish that when all testing is completed, it is clear that all requirements were met.

4.1.1 Purpose

Pre-test involvement is the process of having the PQE Group analyze requirements to ensure they are clearly documented, can be implemented, and testable. It also brings the group into the process early to ensure they understand future requirements and can begin planning to test them as early as possible.

It is beyond the purview of the PQE Group to identify requirements or drive the requirements process. The PQE Group can provide early, meaningful feedback during the process to aid in drafting requirements that are quantifiable and measurable. Well-written requirements are essential to a successful test program and, in many cases, can be used as a genesis for creating test procedures well before the software is delivered for testing.

4.1.2 Required Inputs

The following inputs from outside the PQE Group are necessary to start the pre-test involvement process:

- All requirements documents from CM (this includes documents such as Functional Requirements Specifications [FRS], Software Requirements Specifications [SRS], White Papers, or other requirements documentation as developed by the program)
- Latest Operational Requirements Documents
- Access to the latest specifications that are listed as the requirements documents (e.g., United States Message Traffic Format [USMTF], OS-OTG, and other program Operational Requirements Documents [ORDs])
- Access to Developer Unit Testing

Whenever possible and practical, the PQE Group will participate in any software engineering meetings or In-process Reviews (IPRs) so that they may provide inputs from the software testing perspective. Representation at these meetings is critical to the success of PQE. If representation is not possible, then minutes from the meetings should be forwarded to the PQE Group for review, analysis, or input.

4.1.3 Activities

The following paragraphs describe the interaction of the PQE Group with specific outside events and testing evolutions.

4.1.3.1 Requirements Review

The PQE Group shall review and provide inputs on requirements documents such as SRS, FRS, Functional Design Specifications (FDS), load plans, etc., to ensure testable detail and clarity of specifications (requires attendance at design reviews and delivery of CM-controlled specifications). At the request of the PM, inputs on requirements other than the testability of the requirements will be provided.

4.1.3.2 Unit Testing Verification

This process is for the review of test procedures, test plans, and test execution performed by the developer. When possible and practical, the PQE Group will witness developer execution of their unit testing. As stated previously in this document, ensuring that the developer performs unit testing correctly can provide enormous time, money, schedule, and resource savings. This also assists in allowing PQE personnel to become familiar with new product functionality before delivery, which can also significantly save lifecycle time and reduce cost.

4.1.3.3 In-Process Reviews

PQE needs to participate in IPRs to ensure that product quality is addressed and tracked. Input into the meetings will be performed as necessary. As with other reviews performed or participated in by PQE, it ensures that testing issues are monitored. This also assists in allowing PQE personnel to become familiar with new product functionality before delivery, which can also significantly save lifecycle time and reduce cost.

4.1.4 Outputs

The output of the pre-test involvement process will be written or verbal feedback into the requirements process. The format of this output will be as requested by the responsible PM. If no format is specified, the output will be a marked-up copy of the requirements documents or the default format that is in use for reporting on reviews.

4.2 Integration Test Process

Integration testing is an interim level of testing that occurs between unit testing and system testing and where subassemblies or functional groups of components are tested. This process ensures that a low level of system operations (i.e., application interface calls [APIs], data base queries) operate correctly before moving on to higher level functional tests. If this testing is delayed until higher level functional testing is finsihed, then the problems become more difficult to find, and it may unnecessarily increase development costs.

4.2.1 Purpose

Integration testing ensures that components link and work together, and focuses is on the effectiveness of functional interactions and compatibility at the interfaces. This is a highly technical test and requires a high level of system architecture and system administration expertise.

4.2.2 Required Inputs

The following is a partial listing of inputs required by the PQE Group to begin this process:

- SDF from CM
- Load Plans from CM (developed by System Engineering/ Requirements)

4.2.3 Activities

The integration test process verifies that segments and interfaces interoperate correctly. This process does not test user functionality or requirements; it strictly tests interoperability between pieces of segments and systems. This is a technical test and not a system or user test.

Integration testing will consist of defect validation and execution of test procedures appropriate to exercise the product or products under test.

4.2.4 Outputs

Testing will produce the following documents. As requested by the responsible PM, documents may be excluded from delivery.

- Test Progress Reports (on request only, frequency determined by PM)
- Test Plan and Quality Gates
- Test Report
- Test Procedures

- Defect Reports

Each of the above documents shall be forwarded to the CM Group for tracking after the testing is completed. The report format will be in Microsoft Word 97 or equivalent in the current format. The final reports will be maintained under CM control for the PQE Group. The PQE Manager and the responsible PM will agree on deviations to this procedure.

Test procedures will not normally be distributed outside the PQE Group. Past experience shows that giving test procedures to groups outside the PQE Group results in many support calls with questions on how to execute the test procedures, greatly impacting testing schedules. The PQE Manager (and the PM if reauired) will evaluate all requests for test procedures before delivery. Other documents and reports can be made public.

4.2.5 Pass Criteria

The pass criteria are used to evaluate the potential of the software or system under test to move on to subsequent testing phases. Pass criteria shall be based on the following items:

- Number of Priority 1 and 2 defects
- Ability for system to install based on installation procedures
- Ability for system components to interoperate

The basis of the pass criteria needs evaluation by the Configuration Control Board (CCB) or equivalent to ensure that the software is of sufficient quality to be ready for subsequent testing phases. If the software is unable to meet these basic criteria, then the software should not move to the next phase. Moving the software to subsequent phases before it is mature is a significant drain on resources (both financial and personnel) and should be avoided. Problems that prevent moving to the next phase need to be corrected and validated by integration testing before moving on to he next phase. If there are any failures within the scope of the pass/fail criteria and the software is moved to the next phase based on urgent requirements, those failures must be clearly documented in the test report to ensure successful use of follow-on resources after the integration effort.

4.3 Compliance Test Process

The compliance test process shall be used to establish the degree to which:

- A segment or system achieves conformance with the rules, standards, and specifications identified by the current I&RTS for the DII COE or other similar document
- The segment or system is suitable for integration with the DII COE reference implementation
- The segment or system makes use of COE services

It should be understood that because the product meets the compliance requirements does not mean that it is functionally correct or meets end-user requirements. Completion of compliance testing only indicates that it meets the requirements set forth in the I&RTS for DII COE. A product may pass compliance testing but not pass integration or functional testing.

4.3.1 Purpose

Compliance testing ensures that the software meets the requirements set forth in the I&RTS.

4.3.2 Required Inputs

The following items are required inputs to start the compliance test process:

- Compliance Level Certification from the Developer
- Software from CM
- Installation Guide and SVD for the application and any dependent applications

4.3.3 Activities

This testing shall verify adherence to the current version of the I&RTS, Appendix B. Testing can be conducted for Level 5, 6, or 7, depending on the responsible PM's requirements. Discussion of the requirements contained in the I&RTS are outside the scope of this document.

The first step in this process is to validate any outstanding defects (if they exist) against the new delivery.

4.2.2.1 Use of ChkCompliance Tool

The ChkCompliance Tool (CCT) shall be used whenever possible to test software. This tool permits automated testing of the I&RTS, Appendix B, against the delivered software to Level 7. If developers were required to submit compliance reports with their SDF and that report contained a CCT report generated by the CCT, then only random testing would be conducted. For software that is not scanned by CCT, a desktop review of the results will be conducted.

4.3.3.2 Use of GUI Compliance Check Tool

The Graphical User Interface (GUI) Style Compliance Test Protocol (SCTP) shall be used whenever possible to test software. This tool permits semi-automated testing of the Style Guide compliance, required by the I&RTS to reach Level 6 compliance. If developers were required to submit compliance reports with their SDF and that report contained a GUI SCTP Report, then only random testing will be conducted. For software that is not tested, a desktop review of the results will be conducted.

4.3.3.3 Manual Testing

Whenever the CCT or the SCTP is not appropriate for use, or when directed by the responsible PM, manual testing for I&RTS will be conducted. A checklist developed by the PQE Group will be used to test up to Level 6 compliance. Manual testing of Level 7 compliance is not practical since it requires source code validation.

4.3.4 Outputs

The following documents shall be output as a result of the testing. As requested by the responsible PM, documents may be excluded from delivery.

- Test Progress Reports (on request only, frequency determined by PM)
- Test Plan and Quality Gates
- Test Report
- Test Procedures
- Defect Reports

Each of the above documents shall be forwarded to the CM Group for tracking after the testing is completed. Format of the reports will be in Microsoft Word 97 or equivalent in the current format. The final reports will be maintained under CM control for the PQE Group. The PQE Manager and the responsible PM will agree on deviations to this.

Test procedures will not normally be distributed outside the PQE Group. Past experience shows that giving test procedures to groups outside the PQE Group results in numerous support calls with questions on how to execute the test procedures, greatly impacting testing schedules. All requests for test procedures will be evaluated by the PQE Manager (and the PM if required) before delivery. Other documents and reports can be made public.

4.3.5 Pass Criteria

The pass criteria are used to evaluate the potential of the software or system under test to move on to subsequent testing phases.[7] Pass criteria shall be based on the following items:

- Number of Priority 1 and 2 defects detected
- Risk analysis of defects found
- QA approval of process outputs

The basis of the pass criteria needs to be evaluated by the CCB or equivalent to ensure that the software is of sufficient quality to be ready for subsequent testing phases. If the software is unable to meet these basic criteria, then the software should not move to the next phase. Moving the software to subsequent phases before it is mature is a significant drain on resources (both financial and personnel) and should be avoided. Problems that prevent this need to be corrected and validated by integration testing before moving to the next phase. If there are any failures within the scope of the pass/fail criteria and the software is moved to the next phase based on urgent requirements, those failures

7. Because of the nature of Compliance Testing, it is believed that the pass criteria should be met unless defects were missed in the Integration Test Phase.

must be clearly documented in the test report to ensure successful use of follow-on resources after the integration effort.

4.4 Functional Test Process

These system and acceptance tests focus on the functionality of the system, as seen externally. This testing is commonly called Black Box Testing or functional testing. End-user subject matter expertise is the paramount skill for successful testing, not technical skills.

4.4.1 Purpose

Functional testing ensures that the software meets its functional requirements as laid out in the design specifications (i.e., FRS, SRS, etc.).

4.4.2 Required Inputs

The following items are required to start the functional test process:

- SDF from CM (refer to section 3.9.1 for SDF contents)
- Integration Test Report from CM
- Compliance Test Report from CM

4.4.3 Activities

This testing shall be focused on negative testing. It is designed to test the system from the user's perspective. Therefore, it becomes an ideal point for insertion of operational users, operational test site participants, and trainers into the testing process. Whenever resources and schedule permit, PQE shall use actual system users to augment the testing team.

The first part of this test validates all existing defects, if any, against the new delivery. After finishingdefect validation, test procedures will be executed in the remaining time allocated to the testing process.

4.4.4 Outputs

The following documents shall be output as a result of the testing. As requested by the responsible PM, documents may be excluded from delivery.

- Test Progress Reports (on request only, frequency determined by PM)
    - Test Plan and Quality Gates
    - Test Report
    - Test Procedures
    - Defect Reports

Each of the above documents shall be forwarded to the CM Group for tracking after the testing is completed. Report format will be in Microsoft Word 97 or equivalent in the current format. The final reports will be maintained under CM control for the PQE

Group. The PQE Manager and the responsible PM will agree on deviations to this procedure.

Test procedures will not normally be distributed outside the PQE Group. Past experience shows that giving test procedures to groups outside the PQE Group results in many support calls with questions on how to execute the test procedures, greatly impacting testing schedules.The PQE Manager (and the PM, if required) will evaluate all requests for test procedures before delivery. Other documents and reports can be made public.

4.4.5 Pass Criteria

The pass criteria evaluate the potential of the software or system under test to move to subsequent testing phases. Pass criteria shall be based on the following items:

- Number of Priority 1 and 2 defects detected
- Reliability of software/system
- Risks associated with moving to the next phase
- Quality Assurance approval of outputs
- Security issues detected during testing
- Safety issues detected during testing
- Other criteria as deemed appropriate by the PM

The basis of the pass criteria needs evaluation by the CCB or equivalent to ensure that the software is of sufficient quality to be ready for subsequent testing phases. If the software is unable to meet these basic criteria, then the software should not move to the next phase. Moving the software to subsequent phases before it is mature is a significant drain on resources (both financial and personnel) and should be avoided. Problems that prevent this need to be corrected and validated by integration testing before moving to the next phase. If there are any failures within the scope of the pass/fail criteria and the software is moved to the next phase based on urgent requirements, those failures must be clearly documented in the test report to ensure successful use of follow-on resources after the integration effort.

4.5 Regression Test Process

A regression test is a comprehensive retest of the entire system and/or subsystem after validation that the defect or enhancements were successfully implemented. A regression test should be performed to ensure that the entire system still works as designed. This testing focuses on the remainder of the system functionality, where second order consequences can occur and where errors may have been inadvertently introduced by a modification. Final regression testing should be conducted before system delivery to the end users.

Functional test cases form the core of the regression tests. The functional test cases should be available throughout the life of the system. Regression test coverage should be heaviest for critical functions, complex functions, and functions that have a history of problems. If only portions or subsystems will be affected by a modification, then only a partial regression test of the affected portion will be necessary. Full regression testing should be performed when the overall system architecture has been significantly affected by a modification.

Under circumstances allowing sufficient time and personnel, regression testing should use all Level III test procedures (refer to section 5.2.1 for a description of these test procedures). Where circumstances do not allow adequate time or personnel, testing will be based on depth and breadth (as many areas and as deep into each area as permitted), based on the risk of functionality.

4.5.1 Pass Criteria

The pass criteria are used to evaluate the potential of the software or system under test to move to subsequent testing phases. Pass criteria shall be based on the following items:

- Number of Priority 1 and 2 defects detected
- Reliability of software/system
- Risks associated with moving to the next phase
- Quality Assurance approval of outputs
- Security issues detected during testing
- Safety issues detected during testing
- Excessive number (determined before start of this cycle) of previously corrected defects found
- Other criteria as deemed appropriate by the PM

The basis of the pass criteria needs evaluation by the CCB or equivalent to ensure that the software is of sufficient quality to be ready for subsequent testing phases. If the software is unable to meet these basic criteria, then the software should not move to the next phase. Moving the software on to subsequent phases before it is mature is a significant drain on resources (both financial and personnel) and should be avoided. Problems that prevent this need to be corrected and validated by integration testing before moving to the next phase. If there are any failures within the scope of the pass/fail criteria and the software is moved to the next phase based on urgent requirements, those failures must be clearly documented in the test report to ensure successful use of follow-on resources after the integration effort.

4.6 DT Test Process

The DT test process involves running a simulated laboratory test and operational site tests of developmental software. This ensures that it works and performs on the customer's systems, within the customer's environment, and meets the operational requirements. This phase of testing normally follows successful completion of the integration, compliance, and functional tests. Whereas the previous tests focus on performance and requirements of specific portions of the product, this phase of testing focuses on testing of the system and verifies that it meets higher level requirements.

4.6.1 Purpose

The DT test process ensures that the system meets the system-level specifications. These specifications are normally found in the ORD.

4.6.2 Required Inputs

The following items are required to start the DT test process:

- Integration, Compliance, and Functional Test Reports, Plans, and Procedures from CM
- SDF from CM
- Load Plans from CM
- Developmental Test Plan
- Designation of Developmental Test Site

4.6.3 Activities

This test shall be performed using the entire system with a configuration that is destined for operational use. Testing shall include:

- Load or supervised loading of DT system(s)
- A DT Director provided by the PQE Group
- Control system and software configuration during test
- Coordinate external testing (Joint Interoperability Test Center [JITC]/Navy Center for Tactical Systems Interoperability [NCTSI]/Office of Naval Intelligence [ONI]) as necessary

4.6.4 Outputs

The following items are from the DT test process:

- Test Plan
- Test Report
- Test Quality Gates (system integration test check-off/procedures)

4.6.5 DT Phase I

DT Phase I shall be performed in a controlled laboratory environment. The test team shall be made up of PQE personnel and, if available, actual users of the system. Emphasis shall be on operational validation of requirements from documents such as an ORD or similar high-level requirements specification. Successful execution/completion of this testing is critical to ensure it is stable enough to mitigate any critical risks of taking the software to an operational site.

This phase of testing can also be used as a "Train-the-Trainer" session. It is an opportunity for trainers to view the system in operation and become familiar with it.

4.6.5.1 Stress Testing

Stress testing is putting a system under heavy or varying (high to low to high) communications and/or user loading to evaluate how the system will perform in a demanding environment. This testing can normally be conducted in conjunction with either Functional Test or DT Test Processes as time permits or by request of the responsible PM. This test may also be conducted as a Special Test at any point in the system lifecycle.

4.6.6 DT Phase II

DT Phase II shall be conducted in a controlled environment at an operational site designated by the responsible PM. Pre-test coordination planning for this test shall be closely monitored by the PQE Group to ensure it is successful and that necessary resources for the testing are made available.

It is critical that the system be available to the PQE test engineers as much as possible during this evolution. This is required since this test is similar to DT Phase I, but it is a revalidation of the test at the operational site using the end-user's equipment. Use of the software by operational sites before completion of this phase could introduce unnecessary risk to the operations of the command. If this is not possible, every attempt will be made to ensure that risks are minimized and that appropriate personnel are available to assist the PQE team and the operational site.

4.6.7 DT III or TECHEVAL

DT Phase III, also called the Technical Evaluation (TECHEVAL), will be conducted at the operational site designated by the responsible PM. The site must be involved in an exercise or activities that will put a high load on the system. This test will be conducted by emulating the testing process of the operational certification agency, which will ultimately certify the system with operational users. In this way the test becomes a "dry-run" for the operational tests.

4.6.8 External Agency Testing

Many programs have requirements that their product be tested by independent test agencies such as JITC. In such cases, the PQE Group, under the direction of the PM, shall assist with scheduling and executing these tests. If a test plan and procedures are available, the PQE can execute a "dry-run" before this test to gain insight into the test outcome.

4.6.9 Pass Criteria

The pass criteria are used to evaluate the potential of the software or system under test to move on to subsequent testing phases. Pass criteria shall be based on the following items:

- Number of Priority 1 and 2 defects detected
- Reliability of software/system
- Risks associated with moving to the next phase

- Quality Assurance approval of outputs
- Security issues detected during testing
- Safety issues detected during testing
- Excessive number (determined before start of this cycle) of previously corrected defects found
- Other criteria as deemed appropriate by the PM

The basis of the pass criteria needs evaluation by the CCB or equivalent to ensure that the software is of sufficient quality to be ready for subsequent testing phases. If the software is unable to meet these basic criteria, then the software should not move to the next phase. Moving the software to subsequent phases before it is mature is a significant drain on resources (both financial and personnel) and should be avoided. Problems that prevent this need to be corrected and validated by integration testing before moving to the next phase. If there are any failures within the scope of the pass/fail criteria and the software is moved to the next phase based on urgent requirements, those failures must be clearly documented in the test report to ensure successful use of follow-on resources after the integration effort.

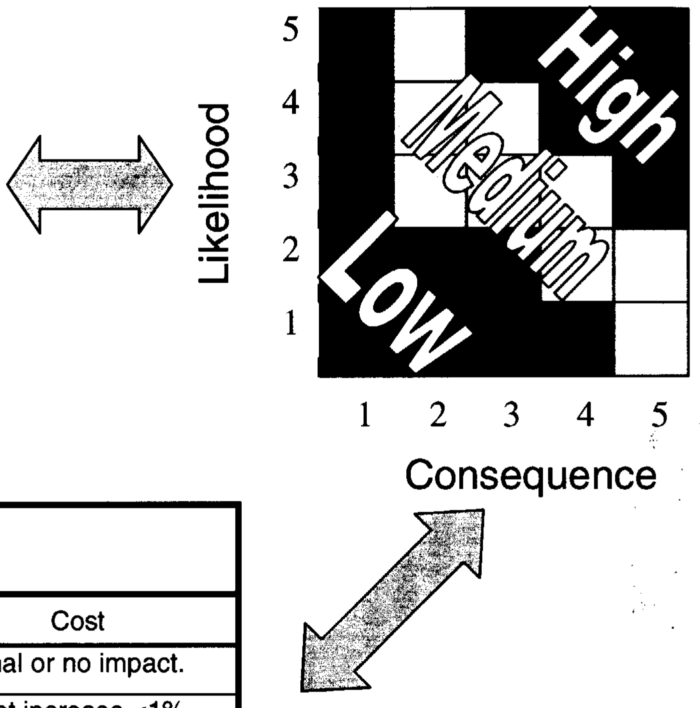
4.7 Risk Management Process

This process uses an extension of the traditional stoplight chart to quantify and present the risk(s) associated with a given project. The PQE Group still uses the traditional red, yellow, and green colors to indicate risk, but there is a standardized method for evaluating a risk. For each risk, the likelihood and consequence are each evaluated on a scale of 1 to 5. Different evaluation criteria are provided for cost, schedule, and technical risks. The process defines the five risk levels, so all testers will be working from the same established foundation. The likelihood and consequence are mapped onto a 5 x 5 matrix that shows whether the risk is high (red), medium (yellow), or low (green). The risk management matrix and an explanation of the evaluation criteria are provided on the following page.

4.7.1 Documenting Risks

For each software product release or update, risks, if there are any, will be identified in the test plan. For each risk identified, one or more consequences will be identified. For each consequence, one or more mitigation options will be identified.

What Is the Likelihood the Risk Will Happen?			
	Level		The Approach and Process ....
Likelihood	1	Not Likely	... will effectively avoid or mitigate this risk based on standard practices.
	2	Low Likelihood	... have usually mitigated this type of risk with minimal oversight in similar circumstances.
	3	Likely	... may mitigate this risk, but workarounds will be required.
	4	Highly Likely	... cannot mitigate this risk, but a different approach might.
	5	Near Certainty	... cannot mitigate this risk; no known processes or workarounds are available.



Given the Risk Is Realized, What Would Be the Magnitude of the Impact?				
	Level	Technical	Schedule	Cost
Consequence	1	Minimal or no impact.	Minimal or no impact.	Minimal or no impact.
	2	Minor performance reduction, same approach refined.	Additional activities required; able to meet key dates.	Budget increase <1%.
	3	Moderate performance reduction, but workarounds available.	Minor schedule slip; will miss need date.	Budget increase < 5%.
	4	Unacceptable, but workarounds available.	Program critical patch affected.	Budget increase <10%.
	5	Unacceptable; no alternatives exist.	Cannot achieve key program milestone.	Budget increase > 10%.

Risk Management Matrix

5.0 ADDITIONAL INFORMATIO N

This section contains additional information on PQE Group processes.

5.1 Test Engineer's Role

During testing, the role of the tester is to develop test plans, test procedures, test cases, and test reports, as well as to execute the testing. In organizations that are not following modern testing methods, testers rely on ad hoc testing without the benefits of planning and reporting. There are many papers and books that show this is ineffective and does not provide adequate testing coverage. The following quote from *The Art of Software Testing* illustrates this point:

"The reason for the importance of test-case design stems from the fact that 'complete' testing is impossible and therefore a test of any program must be necessarily incomplete (i.e., the testing cannot guarantee the absence of errors). The obvious strategy, then, is to try to reduce this incompleteness as much as possible.

Given constraints on time, cost, computer time, etc., the key issue of testing becomes

What subset of all possible test cases has the highest probability of detecting the most errors?

The study of test-case-design methodologies supplies one with answers to this question.

Probably the poorest methodology of all is random-input testing – the process of testing a program by selecting, at random, some subset of all possible input values. In terms of the probability of detecting the most errors, a randomly selected collection of test cases has little chance of being an optimal, or close to optimal, subset."[8]

A tester must never make "on-the-fly" changes to any test configuration. The test suite, just like any other product, must be configuration managed. This is essential to ensuring that the system under test is configured correctly and that the test can be repeatable.

A tester's role is to find and report, not to fix or debug. Whenever test engineers find a defect, they should try, within reason, to ensure that the problem is re-creatable and clearly documented. When test engineers spend too much time trying to debug or isolate a fix, they are not finding defects. This means that 1 day's worth of debugging results in 1 day's worth of defects not being detected, and those undetected defects will be part of the delivered product. It is allowable and understandable for test engineers to assist developers in isolating problems between test events, but careful judgment is required to ensure that their time executing tests is used effectively.

8. *The Art of Software Testing*, Glenford J. Myers, John Wiley & Sons, Inc., 1979.

The test engineer must also be familiar with PQE general information. Appendix C contains the training information that a test engineer is required to know and a list of required classes for test engineers. Appendix D contains additional training materials that prepare the test engineer for their position within the PQE Group.

5.2 Test Procedures

Test procedures shall be generated to ensure that adequate coverage of the product delivered for testing will be scheduled. With the complexity and size of modern computer products, especially software applications, it is impossible to ensure complete coverage of any product. All reasonable efforts will be made to test as much as possible given the schedule and complexity of the testing required.

5.2.1 Depth and Breadth Method

Wherever possible, a Depth and Breadth Method shall be used to select test procedures for execution during a test cycle. Test procedures will be created to ensure that all functions get tested (Breadth) and, depending on Risk and Schedule, more detailed testing (Depth) will be performed. The three levels of test procedures are defined below:

Level I: A high level test that focuses on the basic functionality of the application (What is it supposed to do?).

- Basic functionality checks (Does the application perform its basic function in the system, such as opening a window?)
- Basic database, encoder and decoder checks (Do messages process? Does data get stored?)
- Testing of GUIs is performed only to accomplish basic functionality checks and basic database, encoder, and decoder checks.
- System loading is not specifically evaluated.

Level II: Breadth and detailed depth

- Level I Testing
- System loading that emulates normal operations
- Normal, Consistent DB Activity
- Evaluations of System Communications interfaces
- Standard Negative Testing (Equivalence Partitioning)
- Evaluation of GUIs

Level III: Complete Breadth and depth

- Level II testing
- System loading that emulates heavy operations
- Evaluation of all GUI features
- Evaluation of all system interfaces

• Heavy database activity

5.2.2 Equivalence Partitioning

The basis for Level II testing is equivalence partitioning (EP). EP is a systematic process that identifies, based on the information available, a set of interesting classes of input conditions to be tested. Each class is representative of (or covers) a large set of other possible tests. If partitioning is applied to the product under test, the product is going to behave in much the same way for all members of the class. The aim of EP is to minimize the number of test cases required to cover these input conditions yet maximize the coverage. Studies have demonstrated that EP testing covers a most potential failures with minimal test cases. The two steps to EP are described below.

5.2.2.1 Identifying Equivalence Classes

For each external input, do the following:

a) If the input specifies a range of valid values, define one valid equivalence class (EC) (within the range) and two invalid ECs (one outside each end of the range).

b) If the input specifies the number (N) of valid values, define one valid EC and two invalid ECs (none, and more than N).

c) If the input specifies a set of valid values, define one valid EC (within the set) and one invalid EC (outside the set).

d) If there is reason to believe that the program handles each valid input differently, then define one valid EC per valid input.

e) If the input specifies a "must be" situation, define one valid EC and one invalid EC.

f) If there is reason to believe that elements in an EC are not handled in an identical manner by the program, subdivide the EC into smaller ECs.

5.2.2.2 Identifying Test Cases

Use the following as guidelines to identify each test case.

a) Assign a unique number to each EC.

b) Until all valid ECs have been covered by test cases, write a new test case covering as many of the uncovered ECs as possible.

c) Until all invalid ECs have been covered by test cases, write a test case that covers one, and only one, of the uncovered invalid ECs.

d) If multiple invalid ECs are tested in the same test case, some of those test may never be executed because the first test may mask other tests or terminate execution of the test case.

Reminder: A necessary part of any test case is a description of the expected results, even for tests that use invalid inputs.

EP significantly reduces the number of input conditions to be tested by identifying classes of conditions that are equivalent to many other conditions. It does not test combinations of input conditions.

5.2.3 Test Procedure Metrics

Test procedure metrics will be maintained by the PQE Group. These metrics will be used to better estimate the time required to test a given product. Details of these metrics will not normally be available outside the PQE Group except as represented in schedules. This is to ensure that PQE personnel do not feel controlled by these metrics. All requests for metrics will be reviewed by the PQE Manager on a case-by-case basis.

5.2.4 Test Procedure Execution Priority

For each product, the PQE Group must implement the most cost-effective testing that will ensure that it is reliable enough, safe enough, and meets the user/customer's requirements.[9] No test evolution will permit time to either completely test or execute all available test procedures against a product or set of products. To effectively manage the testing processes and to attempt to get the best coverage of the product in the test time permitted, a logical approach will be used to determine which test procedures to execute and in which order to execute them.

5.2.4.1 Risk Priority

The primary method to establish which test procedures get executed and in which order will be determined by risk analysis. Portions of the product that are identified as high risk or portions that have changed significantly since the last build can be targeted for detailed (Level II or III) testing. In addition, basic testing (Level I) on all other portions of the product should be conducted and carefully tracked. Any new defects found during the Level I testing should be reviewed to evaluate if other areas should be targeted for more detailed testing (Level II). In this way, testing can be targeted to emphasize the high-risk areas of the product and managed to reduce risk.

5.2.4.2 Frequency of Use Priority

Frequency of use is another method for prioritizing testing. Portions of the product that are used heavily by the users, even though they are low risk, may be targeted for testing. A low-priority problem in one of these portions may have a high impact on usability of the system by the end-user.

5.2.5 Test Procedure Generation Priority Matrix

Given that there is not always enough time to perform complete testing of a product, the PQE Group will use a Test Procedure Generation Priority Matrix to guide the creation

9. Edward Kit. *Software Testing in the Real World – Improving the Process.* Addison-Wesley. 1997.

and update of test procedures. This matrix will enable the PQE Group to target high-risk portions of the product for test procedure completion first.

The matrix is designed to use weighted values to ensure that test procedure completeness is emphasized on the quality of the test procedure as well as the risk of operational software failure. Test procedures for portions of the product with the highest risk will be completed and/or updated first, and only then will lower risk procedures be completed or updated.

5.3 Product Changes During Testing

It is critical to executing proper testing that a stable and fixed environment be used whenever possible. However, it is understood that software defect detection is an ongoing process. During testing, it is likely that one or more defects of a high priority (normally Priority 1 or 2 defects) will be encountered. In some cases, these defects may halt the testing evolution (unable to complete testing), impose safety risks, or create security problems that must be corrected before continuation of the testing effort. The PM, developer(s), and the test team need to review each critical defect to evaluate potential risks and impacts to the test evolution. No changes shall ever be made to the DT configuration without the consent of the Test Director. Whenever necessary, especially with safety or security issues, the testers shall have the authority to halt the testing in the absence of the test director or PM until they can be contacted.

During DT, control of the product configuration is even more essential. Whenever possible, a separate controlled test outside of DT needs to be conducted to validate any product changes made during testing before inserting them into the DT.

5.4 Year 2000 Testing

Y2K testing is testing the system to determine how it will respond when the calendar transitions from December 31, 1999 to January 1, 2000. Many systems were designed without checking for this event and may have problems. Many programs have extended the scope of their Y2K testing to include other dates, such as leap year for 2000, fiscal year for 2000, and the millennium rollover (2001). These tests can be conducted during either the Functional or DT Test process; however, they should be conducted during the Functional Test process where detection is more likely. The dates used by PQE were derived from various sources, including Department of Defense (DoD), commercial, and other sources.

5.4.1 Test Procedures

For all major tests, three levels of testing will be performed to test the software thoroughly:

1. Check-off Test Procedures
2. Baseline Functional Test Procedures
3. Commercial Y2K Test Tool

5.4.1.1 Check-off Test Procedures

Level 1 tests include a collection of check-off procedures that are used to exercise every window within the system. All windows, date fields, and database entry windows will be tested to ensure Y2K compliance.

5.4.1.2 Baseline Functional Test Procedures

Level 2 procedures are a more detailed set of baseline test procedures specifically developed to verify software operation within the boundaries of each test case identified as major. These procedures move testers through component, interface, and integration level testing. While the check-off procedures verify minimal system-wide Y2K operation for each window, the baseline functional test procedures ensure full system functionality.

5.4.1.3 Commercial Y2K Test Software

To provide an additional layer of Y2K testing, TRACER 2000s or another similar tool will be used as Level 3 testing in conjunction with the formal test procedures. Use of the automated tool provides the ability to cross-check the functional testing with another method. The tools currently available review software source code line by line and identify or "flag" potentially problematic date fields. In addition, the tester can adjust the tool settings to scan code for date-sensitive operations. After the review of the source code is completed, software developers will revisit the lines of code that were flagged to examine them in greater detail and to modify code, if necessary. In lieu of test engineers performing this check, developers can run the tool and provide the output upon delivery to ensure Y2K compliance. Table 5-1 lists major category transition dates.

Table 5-1. Major Category Transition Dates.

Date	Type of Test
Jan 1, 2000	Year 2000 Transition Processing
January 0, 2000	Year 2000 Transition Process – Negative Test
February 28-29, 2000	Year 2000 Leap year Processing
February 29-30, 2000	Year 2000 Leap Year Processing – Negative Test
March 1, 2000	Year 2000 Leap Year Date Calculations
December 31, 2000	366th Day of the Year
January 1, 2001	Millennium Transition Processing

5.4.2 Year 2000 Transition Processing (1 January 2000)

This test case involves the most critical date within the scope of Y2K testing. The actual Year 2000 transition provides the highest probability for system failure and therefore is the case that will be most examined during Y2K testing.

All three levels of testing will be performed for this test case.

5.4.3 Year 2000 Transition Processing—Negative Test (0 January 2000)

This case is a negative test for the date 0 Jan 2000. To ensure that this date is recognized as invalid, test drivers and data will be generated and fed into the system.

Only the first two levels of testing will be performed; the use of the automated test tool for this case is not appropriate.

5.4.4 Year 2000 Leap Year Processing (28–29 February 2000)

The dates of 28-29 Feb 2000 provide an additional, potentially troublesome transition because 2000 is a leap year (century dates *not* divisible by 400 are *not* leap years, even though they are divisible by four, but century dates divisible by 400 *are* leap years). This special case warrants attention and therefore is a Major test case.

All three levels of testing will be performed for this test case.

5.4.5 Year 2000 Leap Year Processing - Negative Test (29–30 February 2000)

This case is a negative test for the dates 29-30 February 2000. Once again, the first two levels of procedural testing will be performed to ensure that no problems exist and that the system properly recognizes 30 February 2000 as an invalid date.

5.4.6 Year 2000 Leap Year Date Calculations (1 March 2000)

Once testing has been performed to ensure that 28-29 February 2000 is correctly recognized as a leap year, additional testing is warranted to ensure that all system time calculations around the date of 1 March 2000 will be processed correctly. (It is possible that, even though 29 February 2000 would be properly recognized as a leap date, individual software components might not perform proper time, date, speed, etc., calculations around the date of 1 March 2000. This case is provided to ensure that all calculations are performed correctly.)

All three levels of testing will be performed for this test case.

5.4.7 366th Day of the Year 2000 (31 Dec 2000)

Once 29 February 2000 has been correctly identified as a valid date, additional testing will be performed to ensure that proper time, date, speed, etc. calculations are performed around the date 31 December 2000. (The potential for problems arises in this situation because this is the 366th day of Year 2000. It is possible that software would either misidentify this date as invalid or improperly perform date calculations around this date.)

All three levels of testing will be performed for this test case.

5.4.8 Millennium Transition Processing (1 January 2001)

Once all critical dates within Year 2000 are examined and tested, potentially harmful dates beyond 2000 will be tested. The next date that poses the greatest risk for system degradation is 1 January 2001. (Not only is this a year transition date, but it is also the first day of the new millennium.) Table 5-2 lists minor category transistion dates.

All three levels of testing will be performed for this test case.

Table 5-2. Minor Category Transition Dates.

Date	Type of Test
September 9, 1999	Programmer Default (9/9/99 or 9999)
February 29, 2001	Negative Test – Non-leap Years
February 29, 2002	Negative Test – Non-leap Years
February 29, 2003	Negative Test – Non-leap Years
January 1, 2002	Verification of backward calculations (1980s/90s)
January 10, 2000	First nine-character date
October 10, 2000	First ten-character date
February 29, 2004	Leap Year
January 1, 2010	ANSI C overflow problem (specialized testing for EMPSKD and CASREP)

Note: Because of the lower degree of vulnerability posed by these minor transition dates, only Level 1 (Check-off procedures) and Level 3 (Automated Testing Tool) testing will be performed for these cases.

5.4.9 Y2K Test Reporting

As problems are discovered during Y2K testing, defect reports will be generated to document the deficiencies.

5.5 Cost Tracking

It is the desire of the PQE Group to track the costs associated with testing a product. To this end, use of metrics will enable the PQE Group to track the cost of testing a product as well as estimate the costs of testing new segments.

Currently, financial management tools do not permit detailed tracking of independent tasking. The financial tracking tools only apply to government employees; however, many personnel of the PQE Group are contractors. To the best of its ability, the PQE Group shall track the following:

- Process improvement time (time spent on improving the PQE organization)
- Test preparation time (i.e., generation of test procedures, files, scripts)
- Test execution time (execution of the testing cycle)
- Test reporting time (time to generate reports for the testing cycle)

5.6 Test Tools

The following paragraphs describe test tools used by the PQE team.

5.6.1 PQE Test Procedures—File Server

The PQE test procedure file server will host all test procedures for the PQE Group. The PQE file server is a Pentium PC desktop computer running the Windows NT® 4.0 operating system and is connected to the SSC San Diego Unclassified building LAN.

The PQE file organization will be determined by the PQE Lead. PQE files will be stored in a logical sequence based on the applications being tested. Figure 5-1 (located on the following page) is an example of a basic file organization for GCCS-M releases 3.0.2.5 and 4.0.

As test procedures are completed and the corresponding tests are finished, the related directory (i.e., Software Qualification Test [SQT]-1 directory for SQT-1 testing) will be set to "Read Only." This will prevent completed test procedures from being altered or deleted. The file server will be backed-up twice a month using a backup procedure determined by the PQE Lead.

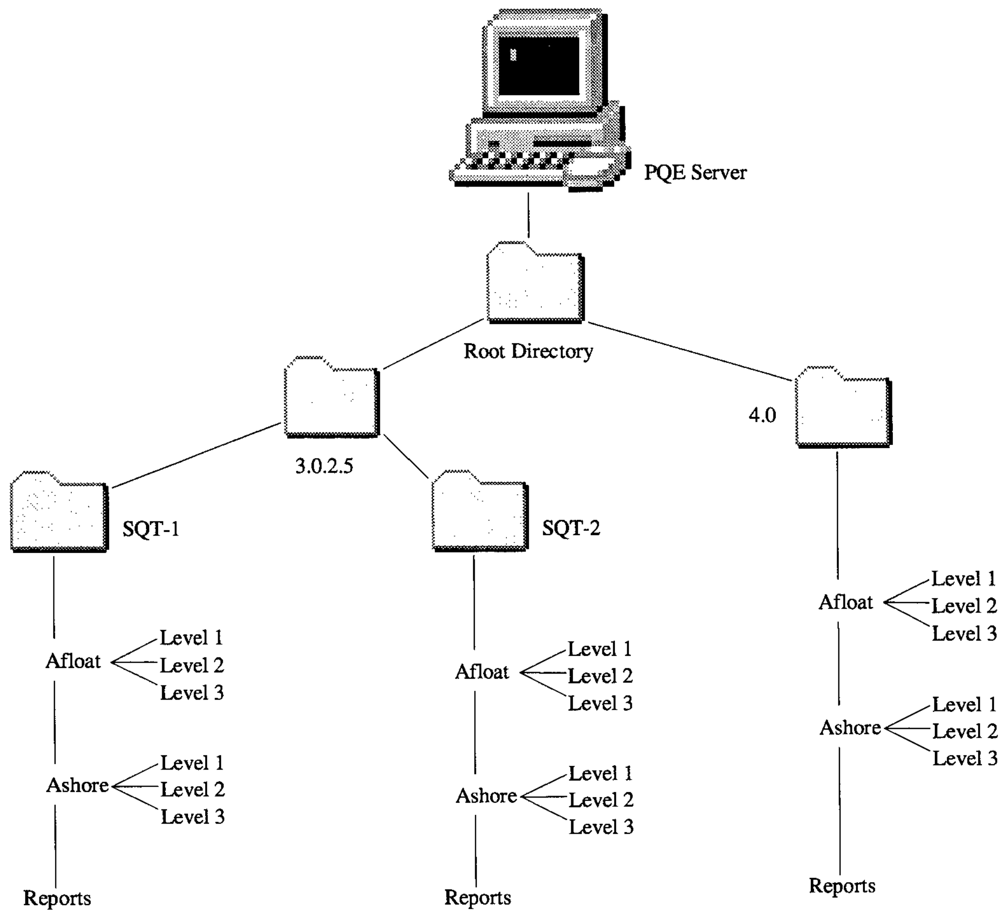


Figure 5-1. Example of Basic PQE Server File Organization.

5.6.2 Test Procedure Numbering System

Test procedures will have a logical numbering system, as assigned by the PQE Lead, so that they may be easily identified. The numbering system consists of fields a, b, c, and d, that identify the sponsor of the testing, test level, application, and component. Table 5-3 shows the test procedure number system.

Table 5-3. Test Procedure Numbering System.

Field	Explanation
a	Sponsor of testing (i.e., DISA, Ashore, Afloat, etc.) 1 = Sponsor 1/Application/System 2 = Sponsor 2/Application/System 3 = Sponsor 3/Application/System 4 = Sponsor 4/ Application/System 5 = Sponsor 5 Application/System
b	Test Level: 1, 2, or 3 for Lever I, II or III
c	Application being tested (i.e., Joint Message Handling System [JMHS], Imagery, Joint Message Tool Kit [JMTK])
d	Component being tested (i.e., UDIE, Image Tracking System [ITS], Profile Server [Prof Svr], Overlays) If there is no component being tested, this field is left blank

For Example: ***4.1.IMG.UDIE.doc*** would be a ***Level I*** test procedure written for ***Sponsor 4***, testing the ***UDIE*** component of the ***Imagery*** application.

Test procedures will be saved to the PQE Test Procedure File Server using this naming convention. For consistency, all test procedures shall be written using Microsoft Word.

5.6.3 ARM Tool

The Automated Requirement Measurement (ARM) tool was developed by the Software Assurance Technology Center (SATC) at the NASA Goddard Space Flight Center as an early lifecycle tool for assessing requirements that are specified in natural language. The ARM tool provides measures that can be used by project managers to assess the quality of a requirements specification document.

The ARM tool searches each line of the requirements document for specific words and phrases that the SATC has identified as "quality indicators." Using these indicators, the ARM tool creates a file that includes three reports: (1) a Summary Report, (2) a detailed Imperative Report, and (3) a detailed Weak Phrase Report. The ARM Summary Report includes the total number of times each quality indicator occurs in the requirements document. The location within the source file of each specification statement identified by the tool and a copy of the specification statement are listed by the Imperative Report. The Weak Phrases Report lists the location and specifications that contain indicators that are considered to be phrases that weaken the specification.

5.6.3.1 PQE Usage

PQE uses the ARM tool to provide metrics on the quality of requirements specifications. The tool can be used for SRS reviews to provide inputs or can be used to evaluate specifications that are delivered as part of a testing environment.

5.6.3.2 More Information

For additional requirements and ARM-related presentations, papers, and tutorials, please view the SATC Project Support and Outreach page, located at: http://satc.gsfc.nasa.gov/support/index.html.

SATC has also released Version 2.0 of ARM 95 with enhanced viewing capabilities, a redesigned GUI, step-by-step walk-through guides, and other enhancements over the previous version. The file can be downloaded from the SATC site after completing the download form located at: http://satc.gsfc.nasa.gov/tools/arm/arm_tool_download_form.html.

5.6.4 REPEAT

The Repeatable Performance Evaluation and Analysis Tool (REPEAT) System software is designed to be used for developmental and operational testing of DoD Command and Control Systems. This system is a compilation of software designed to test asynchronous communications as well as OTCIXS synchronous communications devices. REPEAT is currently used to monitor the receipt and transmission of data to and from Tactical Data Processors (JMCIS/ATWCS/GCCS) and related communications peripherals (ON-143 [V] 6, Generic Front-end Communications Processor [GFCP], and Tactical Receive Equipment [TRE]).

REPEAT system capabilities provide:

- Recording of up to four lines of asynchronous/synchronous data
- Data time stamped to a tenth of a second
- Replay of previously recorded data
- Full screen Y2K compliant text editor to modify pre-recorded data

Analysis features include:

- Conversion of formatted (OTH-GOLD, JUNIT, TACELINT, TACREP, LOCATOR, SENSOREP and TRE TABULAR) messages to a REPEAT database format
- Y2K-compliant volume and timeliness statistics on data received and transmitted at each node
- System throughput and timeliness
- Net Loading
- Common Operational/Tactical Picture System Interoperability
- Accuracy of reported positions versus ground truth
- Analysis of LINK-11 data
- Creation of GOLD, JUINT, SENSOREP, and TACELINT messages for testing.

5.6.5 REPEAT Test File Server

The REPEAT Test File Server, located in Lab 160E at SSC SD, will be used to store REPEAT test files for use in the testing of GCCS-M encoder/decoder capabilities. The REPEAT LAN, shown in figure 5-2, will consist of seven machines. The server will consist of a Pentium II 333MHz computer running Windows NT® 4.0 Workstation. Incremental backups will be conducted daily with full backups occurring every two weeks. Storage media will be 4mm digital audio tape (DAT). Problems or technical support questions regarding the REPEAT server can be e-mailed to pqe@spawar.navy.mil.

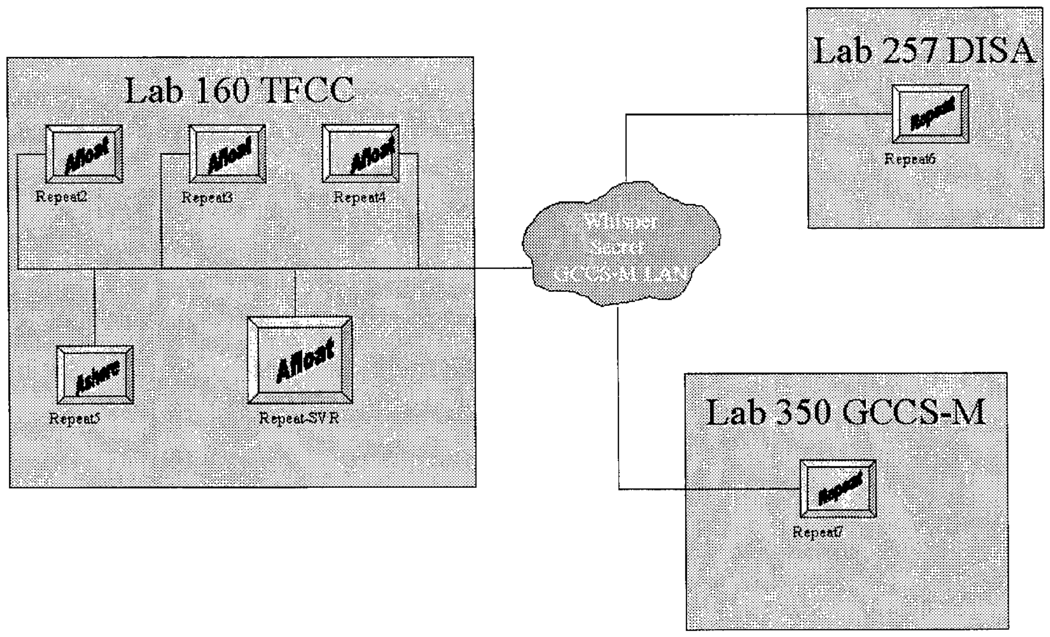


Figure 5-2. REPEAT LAN Configuration.

5.7 Test Methods

Test methods are the techniques used during a test to verify the system requirements. There are five different methods. Each method is defined and described in the following paragraphs.

5.7.1 Observation

Observation is typically done during system activities that do not require operator interaction. The test normally includes a script of planned events to assure the full range of system operation is exercised. Interface message exchange activities where the system under test automatically responds to data/message requests from outside sources is an example. The test method would be to observe the normal, error-free operation of the system under test and the external interfacing system to verify the requirement.

5.7.2 Inspection

Inspection is normally used to verify equipment requirements. An inspection method might be to visually inspect the equipment of the system under test for compliance with equipment-related requirements.

5.7.3 Demonstration

With this method, the operator exercises the system under test through the full range of system activities. This is the most common technique for testing software performance. System operation and expected outputs are evaluated in the normal course of events. This method of verification requires the establishment of procedures that, when performed and successfully completed, demonstrate that the system operation is in accordance with the requirements verified. For tests where the results of the procedure are not readily visible through a menu/screen, this will include inspection of data directory structures and/or files, as appropriate.

5.7.4 Analysis

For this method, system outputs are recorded and then analyzed after completing the test event. For example, a data extraction capability records the system outputs on a medium that supports post-test analysis. The use of tools to aid in analysis of the extracted data may be required.

5.7.5 Calculation

This method is similar to analysis, where the operation of the system under test includes various complex calculations to produce the required outputs. Examination and evaluation of calculation results is normally accomplished during the post-test phase.

5.8 Testware

The following paragraphs are an abridged and paraphrased version of the "Testware" discussion from the book *Software Testing in the Real World* by Ed Kit, published by Addison-Wesley, 1997.

Testware is the product of software test engineers. It includes verification checklists, test data, test plans, test specifications, test procedures, test cases, and test reports.

Testware, like hardware and software, has a life beyond its initial use; therefore, it should be placed under the control of a configuration management system, saved, and maintained. Testware has significant value because it can be reused, sometimes without modification, without incurring the cost of redevelopment with each use.

It is important to maintain testware. Part of the tester's job is to create testware that is going to have a specified lifetime and is a valuable asset to the company. As it is created, testware needs to be put under some sort of control so that it is not lost, either through the loss of the test engineer or some other occurrence. This way the testware can be maintained and the process continued.

Giving testware a name helps to give validity to a test organization that usually does not think of itself as providing any sort of specific deliverable. It gives ownership to what testers do. Testware can also make it easier for testers to communicate with each other and other organizations. Professional testers have a product, and it is testware.

Examples of testware for the PQE Group are contained in Appendix B, Sample Test Forms, Appendix E, PQE Process Forms, and Appendix F, Other Forms.

5.9 Defect QA Process

All defect reports are subject to an internal QA process as described in the following paragraphs. This process assures that the defect reports are clear and can be used both by testers and developers to ensure that the defect can be duplicated. Just as PQE expects clear, quality documents from developers, PQE must provide developers with clear reports of defects found to ensure timely analysis and correction of defects.

5.9.1 Defect Documentation

A Software Change Request (SCR) form will be filled out and submitted documenting the problem. The format used will vary depending on the sponsor's needs and reporting requirements. Often, these forms will be electronic and can be submitted over the Internet, intranet, or similar network.

In addition, the following additional information will also be documented.

1. Provide steps to duplicate the problem. If the problem cannot be duplicated or is intermittent, this will be so stated.

2. Provide any significant environmental conditions/limitations that may have contributed to the problem.

    ➢ system date/time error

    ➢ low disk space/memory

   - high network traffic

3. State the reason for the problem and the likelihood of encountering the problem in an operational environment.
4. State a solution, if applicable.
5. State the impact this problem will have on the use of the software.
6. All Priority 1 and 2 defect reports must contain a justification for their priority status.
7. All Priority 3 defect reports must have a clearly stated work-around.

A sample SCR form and an example defect documentation process are provided in Appendix F.

APPENDIX A: GLOSSARY

Defect	A problem with software, hardware, or the associated documentation
API	Application Programming Interface
ARM	Automated Requirement Measurement
CCB	Configuration Control Board
CCT	ChkCompliance Tool
CM	Configuration Management
CMM	Capability Maturity Model
COE	Common Operating Environment
COTS	Commercial Off-the-shelf
CPU	Central Processing Unit
DAT	Digital Audio Tape
DBDD	Data Base Design Document
DFG	Document Format Guide
DII	Defense Information Infrastructure
DISA	Defense Information Systems Agency
DoD	Department of Defense
DT	Developmental Test
EC	Equivalence Class
ECP	Engineering Change Proposal
EP	Equivalence Partitioning
FDS	Functional Design Specification/Document
FRS	Functional Requirements Specification
GFCP	Generic Front-end Communications Processor
GSPR	Global Software Problem Report
GUI	Graphical User Interface
I&RTS	Information and Runtime Specification
IDS	Interface Design Specification
IPR	In-process Review
IRD	Interface Requirements Document
ITS	Image Tracking System
JITC	Joint Interoperability Test Center

APPENDIX A: GLOSSARY	
JMHS	Joint Message Handling System
JMTK	Joint Message Tool Kit
MIL-STD	Military Standard
NCTSI	Navy Center for Tactical Systems Interoperability
OM	Operator's Manual
ONI	Office of Naval Intelligence
ORD	Operational Requirements Document
OS-OTG	Operational Specification for Over-the-Horizon Targeting Gold
PM	Program Manager
PQE	Product Quality Engineering
Prof Svr	Profile Server
QA	Quality Assurance
REPEAT	Repeatable Performance Evaluation and Analysis Tool
SAM	System Administrator's Manual
SATC	Software Assurance Technology Center
SCM	Software Configuration Management
SCMP	SCM Plan
SCP	Software Change Proposal
SCTP	Style Compliance Test Protocol
SDF	Software Development Folder
SDP	Software Development Plan
SPAWARSYSCEN	Space and Naval Warfare Systems Center
SQA	Software QA
SQT	Software Qualification Test
SRS	Software Requirements Specification
SSC San Diego	SPAWAR Systems Center, San Diego
STD	Software Test Design
STE	Software Test Environment
STP	Software Test Plan
STR	Software Trouble Report
SUM	Software User's Manual
SVD	Software Version Description document
SYSBLD	System Build

APPENDIX A: GLOSSARY

TAO	Tactical Action Officer
TECHEVAL	Technical Evaluation
TR	Trouble Report
TRE	Tactical Receive Equipment
TRR	Test Readiness Review
UDIE	Universal Data Import/Export
USMTF	United States Message Traffic Format
VDD	Version Description Document
Y2K	Year 2000

APPENDIX B: SAMPLE TEST FORMS

The following pages contain samples of the test forms and logs that will be used for conducting tests and recording data. Other forms and logs, more specific than those included here, may also be used for conducting tests and recording data.

QUESTIONNAIRE FOR EFFECTIVENESS

The purpose of this questionnaire is to evaluate effectiveness of the system. Please reply to the following questions or statements on the lines provided by circling the appropriate responses. Comments may be written under any of the items or on a separate piece of paper. This is not an evaluation of the operator but an evaluation of the system. Operator information is requested so that if questions arise on the information provided, the individual can be contacted for clarification. The more information that you provide, the better we can make the system for you.

Name	
Rank/Rate	
Position/Job Description	

1. What is the most important feature of the system? Why?

____
____
____
____

2. List the system features that you think should be changed, starting with the change you think is most important.

____
____
____
____

3. Are there any additional types of information or capabilities that you would find useful? YES/NO If yes, please explain.

____
____
____
____

4. Did you encounter information-handling problems between your system and other systems or subsystems? YES/NO If yes, please explain.

____
____
____
____

5. Do you or did you ever have occasion to suspect the validity of any data? YES/NO
If yes, explain.

6. The system provides me with a better capability to perform my job than alternative/ previous methods.

STRONGLY DISAGREE	DISAGREE	NEUTRAL	AGREE	STRONGLY AGREE

Comments:

7. I am satisfied with the way the system functions.

STRONGLY DISAGREE	DISAGREE	NEUTRAL	AGREE	STRONGLY AGREE

Comments:

8. There are some functions of the system that could be improved.

STRONGLY DISAGREE	DISAGREE	NEUTRAL	AGREE	STRONGLY AGREE

Comments:

9. The system displays information in a timely fashion.

STRONGLY DISAGREE	DISAGREE	NEUTRAL	AGREE	STRONGLY AGREE

Comments:

10. Delays in system operation, which have an impact on operations, seldom occur.

STRONGLY DISAGREE	DISAGREE	NEUTRAL	AGREE	STRONGLY AGREE

If you disagreed with this, please list specific types of delays and explain.

11. The system processes data with sufficient speed to meet my needs.

STRONGLY DISAGREE	DISAGREE	NEUTRAL	AGREE	STRONGLY AGREE

Comments:

12. The system processes data with sufficient accuracy to meet my needs.

STRONGLY DISAGREE	DISAGREE	NEUTRAL	AGREE	STRONGLY AGREE

Comments:

13. The system response to keyboard function key commands is adequate for timely execution of tasks.

STRONGLY DISAGREE	DISAGREE	NEUTRAL	AGREE	STRONGLY AGREE

Comments:

14. Track data displays are rapidly interpreted.

STRONGLY DISAGREE	DISAGREE	NEUTRAL	AGREE	STRONGLY AGREE

Comments:

15. The databases are large enough to support operations.

STRONGLY DISAGREE	DISAGREE	NEUTRAL	AGREE	STRONGLY AGREE

Comments:

16. What single function of the system do you use the most?

17. What single function of the system do you use the least?

18. What single function that is not present in the system would you like to have the most?

WORKSTATION TEST LOG

DTG-ZULU	INITIALS	MACHINE	COMMENTS

APPENDIX C: TRAINING INFORMATION

C-1 SCOPE

This appendix contains a sample listing of information that should be known to all testers. Each small section contains a table that identifies the information it contains. Testers may be required to know the following information that is not included in this section: Immediate Contacts, Organization Division Supervisors, Classes to Attend, Hardware, Software, and Required Reading.

C-2 TESTING

The following table lists test documents that the tester must be familiar with. Next to each item is the expected timetable to become familiar with that document, followed by a person assigned to verify that the tester is familiar with that document.

Name	Timetable	Assigned	Check-off	Date
TEST PLANS	3 months		☐	
TEST PROCEDURES	1 month		☐	
TEST REPORTS			☐	
- Condensed	2 months		☐	
- Full Formal	3 months		☐	
-Executive Summary	4 months		☐	
STR SUBMISSIONS	1 month		☐	
NCR DATABASE	1 month		☐	
RUNNING TEST PROCEDURES	1 month		☐	

APPENDIX D: PQE PROCESS FORMS

This appendix contains the process review forms used by the PQE Group.

ST-001

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Software Test Plan (STP) Development

PROCESS NUMBER: ST-001	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

RELATED PROCESSES: None.

PURPOSE:
To ensure early and continual user-centered software verification, software test planning begins during the project start-up phase and continues in earnest once a preliminary set of system/software requirements has been defined. The major objective of test planning is to develop the framework for attaining system certification. Test design and implementation closely follow the development activities of software designing and implementation: design and documentation of the specific plans, methods, descriptions, procedures, and techniques to be employed to ensure that the developed software product meets the allocated requirements. Software testing may be conducted at multiple points in the development cycle and against a variety of software baselines. The tests range from informal such as unit level (using simulators and drivers), integration and regression tests, to formal, sponsor/customer witnessed system tests in the target environment using real-time data and end users. The purpose of test planning is to ensure that the amount, type, and formality of testing is consistent with the nature of the software product (i.e., complexity, criticality, maturity, size, etc.), establish the system certification pass/fail criteria, identify required testing resources and address contractual test obligations. Specifications for the Software Test Environment (STE) are also defined during the test planning process.

SPECIAL CONSIDERATIONS: None anticipated.

RESPONSIBILITY:
The project Test Manager is responsible for developing the STP.

INPUTS:
a. Tasking agreement or Contract.
b. Statement of Work.
c. Software Development Plan (SDP).
d. T&E Master Plan (TEMP).
e. Other project plans, as applicable.
f. System/Software requirements specifications.
g. PQE Process Control Document

ENTRY CRITERIA:
Availability of a preliminary version of system/software requirements.

OUTPUTS:
a. STP completed and ready for review.
b. Other related informal project plans, such as test data (as appropriate).
c. STE specification

EXIT CRITERIA:
STP baselined under configuration control.

VERIFIABLE OBJECTIVE EVIDENCE:
a. Completed STP development process document (ST-001).
b. Completed PQE metrics collection spreadsheet.

SUGGESTED METRICS:
a. Number of tests defined by test technique.
b. STP development effort in labor hours by skill level (low, mid, and high).

TOOLS/FORMS/CHECKLISTS:
PQE metrics collection spreadsheet

DEVIATION/TAILORING:
Deviation from this process is authorized at the direction of the PQE Manager.

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Software Test Plan (STP) Development

PROCESS NUMBER: ST-001	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

REFERENCE DOCUMENTS:

a. Integration and Run Time Specification (I&RTS), Version 3.0, Joint Interoperability and Engineering Organization, July 1997.

b. User Interface Specifications for the Defense Information Infrastructure (DII), Version 2.0, 1 April 1996.

c. PQE Process Control Document, Version 1.5, 7 June 1999.

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	ECP	Engineering Change Proposal
7.	GUI	Graphical User Interface
8.	I&RTS	Integration and Runtime Specification
9.	PCD	Process Control Document
10.	PQE	Product Quality Engineering
11.	RTM	Requirement Traceability Matrix
12.	SCTP	GUI Style Compliance Test Protocol
13.	SDP	Software Development Plan
14.	SEPO	Software Engineering Process Office
15.	SPCR	Software Problem Change Request
16.	SQA	Software Quality Assurance
17.	SSC SD	Space and Naval Warfare Systems Center, San Diego
18.	STD	Software Test Design
19.	STE	Software Test Environment
20.	STP	Software Test Plan
21.	STPR	Software Test Procedure
22.	STR	Software Test Report
23.	SVD	Software Version Description
24.	TEMP	Test & Evaluation Master Plan

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Plan (STP) Development			
PROCESS NUMBER: ST-001	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	**ACTIVITIES**	**DATE**
1	**Determine the overall system and program test strategy, and the pass/fail criteria for both formal and informal tests.** Note: To ensure system usability, early and continual user-centered testing must be a major goal for all test planning and testing activities. Ensure that the following topics are considered: a. The role of the software being developed in the overall system/program in terms of the users and tasks. The level of testing must be commensurate with the level of criticality and importance of the software function. Note: All projects must conduct regression, system and acceptance testing, unless specified otherwise in the tasking agreement or contract. b. The level(s) of testing to demonstrate that the software meets requirements and the organization(s) responsible for each level of testing. Include organizations beyond the immediate customer that must buy-off on the system being developed and any special operational, field, and/or certification tests required. Identify schedules and locations for any such testing and the role of the software developer. Note: Use the information in Table 1 to help determine the categories of testing that are required and the focus of such testing. Ensure that planned tests verify satisfaction of established system/human behavioral goals. c. Any specific requirements that must be met as specified in contractually binding documents (e.g., computer/human interface standards, a TEMP that imposes additional test requirements, etc.) This includes establishing the pass/fail (acceptance) criteria that will be used and formally agreed to by the customer. d. External factors that may influence the developer's ability to adequately demonstrate software performance such as the availability, maturity, and stability of interfacing configuration items, systems or subsystems and/or customer supplied items. Include alternative workarounds for potential problems related to external factors. e. Standards or guidelines to be used for the development of test documentation and test conduct including the ground rules for tailoring and/or obtaining waivers. Also, identify all automated test tools (test generator, checkers, etc.) that will be used during the test cycle.	

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes		
PROCESS: Software Test Plan (STP) Development		
PROCESS NUMBER: ST-001	**REV./CHG: 0**	**SUPERSEDES:** **EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
	f. Availability of any existing test documentation, tools, and scenarios that may be candidates for re-use of this test effort. For example, a program may have multiple systems or subsystems with common or similar requirements and it may be feasible to use existing test materials (often with minor modifications) even though there is no reuse of software. g. Level of testing required for test drivers, simulators and other support software that are used to test or validate the system during various test phases. h. Requirements to use live versus simulated data.	
2	**Review the feasibility of the SDP, schedules, staffing, and contractual agreements in relation to each required test phase to ensure that test obligations/activities are accurately reflected. Initiate any necessary changes.**	
3	**Identify project specific factors that will influence the test strategy using the following list as a guide:** a. Review all system and allocated software requirements and the preliminary design to identify the following: 1) Relative size, complexity, and critically of each component 2) Complex algorithms and data structures 3) Special security considerations 4) Required level of confidence in the software (e.g. tactical vs. simulation) 5) Interfaces (internal and external) 6) Integration requirements 7) Functional requirements 8) Defense Information Infrastructure Common Operating Environment (DII COE) compliance requirements. 9) Computer Human Interface (CHI) requirements. b. If any requirements are deemed untestable, initiate an Engineering Change Proposal (ECP) against the requirement baseline. Notify the project Test Manager. c. Review the SDP to determine the software build approach and content. This will have a direct influence on the approach to the component integration level tests and the need to develop test drivers and simulators. d. Review design and coding schedules. e. Review plans for staffing the test team and the availability of qualified test personnel.	

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Software Test Plan (STP) Development			
PROCESS NUMBER: ST-001	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
4	**Develop the overall test strategy based on all factors identified in steps 1 and 3, including:** a. The levels and types of tests to be conducted. b. The organization responsible for each level/type of testing. c. Formal versus informal software tests. d. Regression testing.	
	Note: Since there is rarely enough time and resources to perform ideal testing, developing the test strategy should be an iterative process involving risk and trade-off analyses. The test strategy should reflect the most thorough and comprehensive test coverage feasible given the project's requirements, cost and schedule considerations. Remember that cost and schedule should never be traded for test coverage where human safety requirements exist. Obtain project Test Manager, PQE Manager, Account Manager and Software Quality Assurance (SQA) concurrence with the proposed test strategy before proceeding to the next step.	
5	**Conduct detailed test planning as follows for each software item under development:** a. Define the STE in which the tests will be conducted. Ensure that the STE definition addresses requirements for both laboratory and field environments. Identify the following as a minimum: 1) All hardware/firmware components and configurations required to support each test. • Include all system peripherals, simulators and interfacing equipment. • Address any procedures required to obtain hardware resources not directly assigned to support testing. 2) All software required to support each test. • Include all configurations of the software under test, compilers, test tools, drivers, and simulators as well as any database and scenario data requirements. • Identify the source of all software items, including those items to be developed specifically for this project. • Identify the support software licenses required to conduct testing.	

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Plan (STP) Development			
PROCESS NUMBER: ST-001	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
	3) The strategy and timing for integrating and testing of software and hardware components. 4) Whether the engineering load build or the configuration load build is to be used for regression tests Note: The target hardware and software should be used whenever possible to minimize risk. Test drivers and simulators should be used to 1) augment the target environment where the full system is not available or 2) support stress and endurance tests. Test tools, which help to streamline, automate and control the test process, should be used to the greatest extent possible. 5) Plans for installing and testing each support item in the STE and plans for controlling and maintaining the STE during each test phase.	
	b. Define the specific test cases necessary to implement the test strategy and the associated pass fail criteria for each test case. c. Develop a traceability matrix that maps the allocated system and software requirements to each test case. Note: This matrix can be made an appendix of the STP and will become increasingly more detailed as the software development effort progresses and, therefore should be maintained in a format that facilitates easy update. Each test case may be decomposed into one or more test procedures. d. Define the data reduction and data analysis plans and techniques to be used to evaluate the test results and determine whether or not each test case passed or failed.	
	e. Define the minimum test report contents: • Test description, purpose, and overview. • Test results • Test evaluation and recommendations. • Test logs. f. Develop a schedule and staffing plan to implement the test strategy. Include: • The schedule for test design, procurement, test development, and informal and formal test conduct. • The development and verification of required tools and drivers • The production of all test documentation. • The numbers, qualifications, and sources of all personnel required to support planned tests.	
6	**Review, negotiate and agree on all plans with groups that are involved in or affected by the testing activities (including the customer).**	

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Software Test Plan (STP) Development			
PROCESS NUMBER: ST-001	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
7	**Implement a mechanism to keep the test team informed of all project-related items that may affect testing. Include the project Test Manager in decisions regarding any changes in schedule or significant changes in product requirements and/or design.**	
8	**Document the results of the test planning activities, step 1 through 7, above, in the STP in accordance with the documentation standard specified in the tasking agreement or contract.** Note: The SSCSD Software Engineering Process Office (SEPO) web page (http://sepo.spawar.navy.mil/) contains an STP template.	
9	**Update the Requirements Traceability Matrix.**	
10	**Conduct a review of the STP in accordance with PQE Software Test Plan Review Process ST-002.**	
11	**Make all required modifications and corrections to the STP based upon review comments.**	
12	**Submit the STP for re-review, if the previous review identified:** a. One or more severe defects and/or b. A large volume of other defects.	
13	**Repeat Steps 10 and 11 until the STP successfully completes the review process.**	
14	**Submit STP to the customer for review and approval. Upon approval, the STP is placed under configuration control.**	

Test Manager Signoff:		**Date:**
SQA Signoff:		**Date:**
Project Id:		

Table 1. Categories of Testing.

	TEST LEVEL	TESTING TECHNIQUE	PURPOSE	CHARACTERISTICS
1.	Regression	Combination of Integration, Functional and System/ Stress	Verify correctness of changes and search for side-effects as a result of changes.	Decompose configuration items into their constituent components, as necessary. Repeat test scenarios on updated software to verify defect correction. Assess impact of changes. Integrate and retest configuration items, as appropriate.
2.	Integration/ Functional	Boundary Condition	Interface testing finds errors in input and output parameter tolerances and verifies the program limits are correctly stated and implemented.	Test tolerances such as parameter minimums, maximums, and "just beyond" minimums and maximums. Choose input parameters that test both input and output tolerances.
3.	Integration/ Functional	Path	Execution of every logic branch and line of code finds logic errors at loop boundaries and errors in loop initializations.	Perform McCabe's cyclematic complexity analysis to help determine number and focus of Software Test Descriptions. Perform path sensitization to determine test case parameters. Choose parameters that complement other techniques, such as boundary conditions and invalid syntax inputs.
4.	Integration/ Functional	Transaction Flow	Uncovers functional and performance errors in the execution of integrated units and components.	Similar to path testing but on a functional/performance level. Perform transaction flow path selection using functional specifications, from perspective of the system users. Test for valid and invalid paths.
5.	Integration/ Functional	Input Validation And Syntax	Verify the product error handling capabilities operate as required and that these capabilities are sufficient for the errors that occur. Valid and invalid inputs uncover errors in the user/system interface under test.	Force every error message and verify the accuracy and clarity of each. Choose valid and invalid input parameters. Invalid parameters include wrong type, scope, length, and special keyboard characters, ESC, CTRL, etc.

	TEST LEVEL	TESTING TECHNIQUE	PURPOSE	CHARACTERISTICS
6.	Integration/ Functional	Equivalence Partitioning	Reduce necessary number of Software Test Descriptions for adequate coverage. Uncover functional errors in the execution of integrated units and components.	Improve probability of uncovering errors versus random cases by partitioning Software Test Descriptions by class. Choose a representative set of cases that covers all classes.
7.	Integration/ Functional	Database (as applicable)	Testing uncovers problems with Commercial Off The Shelf (COTS) interfaces, data corruption, and unauthorized access.	Known starting point and expecting end points are mandatory. Tests should be automated to facilitate retest.
8.	Integration/ Functional	State Transition (as applicable)	Uncover incorrect, dead, unreachable, and impossible states.	Traverse successive states and verify correct next state and outputs (if any).
9.	Functional	Equivalence Partitioning	Reduce necessary number of Software Test Descriptions for adequate coverage. Uncover functional errors in the execution of integrated units and components. Verify proper functional capability to system/software requirements.	Improve probability of uncovering errors versus random cases by partitioning Software Test Descriptions by class. Choose a representative set of test cases that cover all classes.
10.	DII COE Complianc e	Demonstrati on	Determine product compliance with DII COE rules, specifications and standards.	Run ChkCompliance Tool (CCT) on the application software, as applicable, to automatically test for compliance to the Integration and Run Time Specification (I&RTS). Run the Graphical User Interface (GUI) Style Compliance Test Protocol (SCTP) tool on the application software, as applicable, to test compliance with the GUI style guide required by the I&RTS. Manually test the application software for I&RTS compliance up to Level 6.
11.	System/ Stress	Volume	Determine level of continuous heavy load at which system fails.	Run near peak load conditions for a sustained period of time.
12.	System/ Stress	Performance	Determine actual performance for throughput, timing, etc.	Measure transaction rates and response times.

	TEST LEVEL	TESTING TECHNIQUE	PURPOSE	CHARACTERISTICS
13.	System/ Stress	Configuratio n	Test various hardware and software configurations that must be supported.	Exercise hardware dependent code. Exercise code on various hardware configurations to verify that there are no hidden hardware dependencies.
14.	System/ Stress	Compatibility	Verify that the program is consistent with any other program(s) with which it claims compatibility.	Exercise hardware and software interfaces, data and language classes.
15.	System/ Stress	Load/Stress	Identify peak load conditions at which the system fails.	Subject system to peak rates for key parameters.
16.	System/ Stress	Security	Identify unauthorized system entry points.	Attack the system. Use domain experts to construct strategy.
17.	System/ Stress	Reliability and Availability	Determine Reliability and Availability at typical load over a long period.	Estimate reliability based on metrics.
18.	System/ Stress	Degradation and Recovery	Verify both graceful and unexpected shutdown and recovery behavior.	Simulate environment, system failure.
19.	System/ Stress	Insatiability	Identify incorrect installation procedures.	Dry-run installation procedures.
20.	Operational	Usability Testing	To ensure usable/functional product(s) by proving fundamental concepts early.	Test CHI of production prototype(s) using target user population performing representative tasks.
21.	Acceptance	Combination of Functional and System/ Stress	Demonstrate the system performs to software requirements. Achieve system sell-off.	Perform testing of functional and system level requirements from the system user perspective.

ST-002

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Software Test Plan (STP) Review

PROCESS NUMBER: ST-002	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

RELATED PROCESSES: ST-001

PURPOSE:
Conduct a comprehensive technical evaluation of the STP to ensure the document is complete, correct and addresses all required software test issues.

SPECIAL CONSIDERATIONS: None anticipated.

RESPONSIBILITY/AUTHORITY:
The project Test Manager is responsible for assigning experienced technical personnel to conduct a comprehensive review of the STP and designating a review team leader or review moderator.

INPUTS:
h. Tasking agreement or Contract.
i. Statement of Work.
j. Software Development Plan (SDP).
k. Test and Evaluation Master Plan (TEMP).
l. Other project plans, as applicable.
m. System/Software requirements specifications.
n. Approved Software Test Environment (STE) specifications (as appropriate).
o. STP under configuration control.
p. PQE Process Control Document

ENTRY CRITERIA:
STP completed and ready for review.

OUTPUTS:
STP review process checklist and review comments form (See table1).

EXIT CRITERIA
STP approved and placed under configuration control.

VERIFIABLE OBJECTIVE EVIDENCE:
a. Completed PQE Software Test Plan Review process, ST-002.
b. Completed STP review comments forms.
c. Approved STP under configuration control.
d. Completed PQE metrics collection spreadsheet.

SUGGESTED METRICS:
a. Level of effort in labor hours by skill levels (low, mid, high)
b. Number of STP defects, by type and severity

TOOLS/FORMS/CHECKLISTS:
a. PQE metrics collection spreadsheet
b. PQE Software Test Plan Review Process, ST-002.
c. STP review comment form (See table 1).

DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.

REFERENCE DOCUMENTS:
d. Integration and Run Time Specification (I&RTS), Version 3.0, Joint Interoperability and Engineering Organization, July 1997.
e. User Interface Specifications for the Defense Information Infrastructure (DII), Version 2.0, 1 April 1996.
f. PQE Process Control Document, Version 1.5, 7 June 1999.

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Software Test Plan (STP) Review			
PROCESS NUMBER: ST-002	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	ECP	Engineering Change Proposal
7.	GUI	Graphical User Interface
8.	I&RTS	Integration and Runtime Specification
9.	PCD	Process Control Document
10.	PQE	Product Quality Engineering
11.	RTM	Requirement Traceability Matrix
12.	SCTP	GUI Style Compliance Test Protocol
13.	SDP	Software Development Plan
14.	SEPO	Software Engineering Process Office
15.	SPCR	Software Problem Change Request
16.	SQA	Software Quality Assurance
17.	SSC SD	Space and Naval Warfare Systems Center, San Diego
18.	STD	Software Test Design
19.	STE	Software Test Environment
20.	STP	Software Test Plan
21.	STPR	Software Test Procedure
22.	STR	Software Test Report
23.	SVD	Software Version Description
24.	TEMP	Test & Evaluation Master Plan

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Software Test Plan (STP) Review			
PROCESS NUMBER: ST-002	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

STEP	ACTIVITIES	DATE
1	**Evaluate the entry criteria to determine the STP readiness for review** • STP is complete, baselined and under configuration control. Note: If the entry criteria have not been met the STP review cannot be conducted at this time. Correct the deficiencies and schedule the review for another time.	
2	**Verify availability of required references to support the STP review:** a. Tasking agreement or Contract. b. Statement of Work. c. SDP. d. TEMP. e. Other project plans, as applicable. f. System/Software Requirements Specification(s). g. STE specifications.	
3	**Assign review team participants:** • Two or more senior test engineers; • At least one senior System Engineer or S/W Engineer; • Test specialists, as required. Note: Invite customer/client representatives to participate in the review, as appropriate.	
4	**Obtain copies of the STP Review Comments Form and appropriate PQE metrics collection spreadsheet.**	
5	**Initiate STP review.** If review is to be conducted formally: a. Assign an experienced/trained moderator to organize and conduct the review. b. Assign a scribe to document comments. c. Provide the moderator with the required STP review materials, Peer Review Formal Inspection Log and STP review checklist. d. Instruct the moderator to prepare for and conduct the review.	

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Plan (STP) Review			
PROCESS NUMBER: ST-002	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

STEP	**ACTIVITIES**	**DATE**
	If review is to be conducted informally: a. Assign a STP review team leader. b. Distribute the review materials to the participants with instructions to complete the Peer Review Formal Inspection Log and the STP review checklist and return the materials to the team leader. For formal and informal reviews: a. Instruct the STP author to incorporate straightforward comments, resolve any conflicting or seemingly inappropriate comments with the moderator or team leader, and initial and date all resolved comments. b. Moderator/team leader reviews the updated STP, Peer Review Formal Inspection Log and review checklist. Where comments are related to problems and issues requiring resolution, assign appropriate action items. c. Moderator/team leader fills out the PQE metrics collection spreadsheet and forwards to Configuration Management with the STP review checklist and review comment form.	
6	**Verify a completed copy of the STD Review Checklist ST-002 and the completed STP review comment forms have been placed in the project files.**	

Moderator/Team leader Signoff:		**Date:**
SQA Signoff:		**Date:**
Project Id:		

Table 1
STP Review Comment Form

Date:	Reviewer:		
Project Id:	Document ID:		
		Yes	No
1.	Does the Test Plan identify the test organization and appropriate staffing (skills and skill-levels)?		
2.	Does the Test Plan define a workable overall formal test approach?		
3.	Does the test plan identify a standard naming convention for test conditions/variations and test procedures?		
4.	Does the test approach/schedule promote evaluation of software soon after its availability?		
5.	Does the test plan address all relevant types of tests; e.g., integration, compliance, functional, DTI/II, regression, etc.?		
6.	Does the Test Plan identify all relevant environmental factors and variants; e.g., hardware constraints, etc.?		
7.	Does the plan correctly interpret requirements?		
8.	Do all test conditions/variations trace back to the software requirements?		
9.	Does the Test Plan identify a reasonable set of test conditions/variations for each requirement?		
10.	Do the test conditions/variations include tests of messages generated by the software?		
11.	Do the test conditions/variations identified provide adequate test coverage; error, as well as legitimate conditions?		
12.	Does the Test Plan identify all needed support software (i.e., drivers, simulators, emulators, etc.), test equipment and responsibility for acquisition/development?		
13.	Does the Test Plan describe hardware constraints, requirements?		
14.	Does the Test Plan identify test data requirements and needed time frames?		
15.	Does the Test Plan identify predicted test problem areas and approaches to mitigation?		
16.	Does the Test Plan identify how tests will be verified?		
17.	Does the allocation of conditions/variations to test procedures minimize redundancy?		
18.	Is the allocation of test conditions/variations to test procedures reasonable and logical?		
19.	Are all test conditions/variations allocated to at least one test procedure?		
20.	Does the Test Plan address configuration control of all test materials?		

Additional Comments:

ST-003

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Design (STD) Development			
PROCESS NUMBER: ST-003	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 8/23/99**
RELATED PROCESSES: ST-001			
PURPOSE: Translate each Software Test Plan (STP) defined test into corresponding test case(s) to verify the allocated system/software requirements. Define test case inputs, outputs, expected results and requirement pass/fail evaluation criteria. Identify the unique test environment for each test case, as appropriate. Update the STP Requirement Traceability Matrix (RTM) for each test case.			
SPECIAL CONSIDERATIONS: None anticipated.			
RESPONSIBILITY: The project Test Manager is responsible for developing the STD.			
INPUTS: a. Tasking agreement or Contract. b. Statement of Work. c. Software Development Plan (SDP). d. Test and Evaluation Master Plan (TEMP). e. Other project plans, as applicable. f. System/Software requirements specifications. g. Approved STP under configuration control. h. Approved Software Test Environment (STE) specifications (as appropriate). i. PQE Process Control Document.			
ENTRY CRITERIA: STP is approved and baselined under configuration control.			
OUTPUTS: a. STD completed and ready for review. b. Updated STP, as appropriate. c. Updated STE Specification, as appropriate.			
EXIT CRITERIA: STD baselined and placed under configuration control.			
VERIFIABLE OBJECTIVE EVIDENCE: Completed PQE Software Test Design development process, ST-003. Completed PQE metrics collection spreadsheet			
SUGGESTED METRICS: a. Number of test cases developed. b. Level of effort in labor hours by skill levels (low, mid, high). c. Average number of requirements per test case.			
TOOLS/FORMS/CHECKLISTS: PQE metrics collection spreadsheet.			
DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.			

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Software Test Design (STD) Development			
PROCESS NUMBER: ST-003	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 8/23/99**

REFERENCE DOCUMENTS:

a. Integration and Run Time Specification (I&RTS), Version 3.0, Joint Interoperability and Engineering Organization, July 1997.

b. User Interface Specifications for the Defense Information Infrastructure (DII), Version 2.0, 1 April 1996.

c. PQE Process Control Document, Version 1.5, 7 June 1999.

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	ECP	Engineering Change Proposal
7.	GUI	Graphical User Interface
8.	I&RTS	Integration and Runtime Specification
9.	PCD	Process Control Document
10.	PQE	Product Quality Engineering
11.	RTM	Requirement Traceability Matrix
12.	SCTP	GUI Style Compliance Test Protocol
13.	SDP	Software Development Plan
14.	SEPO	Software Engineering Process Office
15.	SPCR	Software Problem Change Request
16.	SQA	Software Quality Assurance
17.	SSC SD	Space and Naval Warfare Systems Center, San Diego
18.	STD	Software Test Design
19.	STE	Software Test Environment
20.	STP	Software Test Plan
21.	STPR	Software Test Procedure
22.	STR	Software Test Report
23.	SVD	Software Version Description
24.	TEMP	Test & Evaluation Master Plan

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Software Test Design (STD) Development			
PROCESS NUMBER: ST-003	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 8/23/99**

STEP	ACTIVITIES	DATE
1	**Review the requirements allocated to each test defined in the STP.** a.Identify how the requirements are interrelated. b.Sanity check the allocation of requirements to tests.	
2	**Describe each software test case to include:** a. Test case name b. Test type(s) and objectives/purpose c. Allocated system/software requirements and test conditions/variations Note: Requirements allocated to tests must be traceable to system/software requirements and all system/software requirements must be represented in test procedures. d. Required hardware and software environment, including software under test, COTS software versions, analysis tools, other test equipment, etc. May reference specific STE configurations and any modifications required for the specific test. e. Test set up (i.e. pre-test) conditions for hardware and software. f. Description of test inputs (source, values, accuracy, sequencing, etc. as applicable) g. Description of expected test results. h. Criteria for evaluating the test results. i. Assumptions and constraints.	
3	**Identify logical groups of allocated requirements that can easily be tested together:** a. Functionally-related conditions/variations b. Conditions/variations that are testable without restoration of the test environment. Note: Ensure each test condition/variation is traceable to at least one of these groupings.	
4	**For each logical group of requirements, identify and document a sequence of test events that:** a. Tests one or more conditions b. Requires relatively few steps c. Can be executed in reasonable period of time d. Is easily repeatable e. Maximizes the integrity and independence of individual tests f. Minimizes duplication of steps/tests g. Minimizes dependency on data/conditions generated in previous test steps h. Culminates in an obvious pass/fail result i. Traces to the allocated test conditions/variations.	

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Software Test Design (STD) Development			
PROCESS NUMBER: ST-003	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 8/23/99**

STEP	ACTIVITIES	DATE
5	**Document the expected results of each sequence of test events (including accuracy, bounds, limits, duration/timing, pass/fail, etc.) and the verification method.**	
6	**Are additional support software requirements defined in the STD that are not identified in the STE specifications?** Determine: a. What support software is needed ? b. When will each support software be required ? c. How will each support software be obtained; e.g., make/buy ? d. Who is responsible for obtaining/developing each support software ?	
7	**Identify test-specific data requirements.** Determine: a. What test data are needed to support each test design? b. When will test data be required ? c. How will test data be obtained (e.g., develop, use selected live data, etc.) ? d. Who is responsible for obtaining/developing test data ?	
8	**Repeat Steps 2 through 7 until development of all required test cases are completed.**	
9	**Update the STP Requirement Traceability Matrix with STD data.**	
10	**Complete the STD peer review, as described in PQE process number ST-004.**	
11	**Forward the STD to Configuration Management for placement under configuration control.**	

Test Manager Signoff:		**Date:**
SQA Signoff:		**Date:**
Project Id:		

ST-004

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Design (STD) Review			
PROCESS NUMBER: ST-004	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

RELATED PROCESSES: ST-003

PURPOSE:
Conduct a comprehensive technical evaluation to ensure the STD is complete, correct, efficient, and addresses the allocated functional requirements.

SPECIAL CONSIDERATIONS: None anticipated.

RESPONSIBILITY/AUTHORITY:
The project Test Manager is responsible for assigning experienced technical personnel to conduct a comprehensive STD review and designating a review team leader or review moderator.

INPUTS:
a. Tasking agreement or Contract.
b. Statement of Work.
c. Software Development Plan (SDP).
d. Test and Evaluation Master Plan (TEMP).
e. Other project plans, as applicable.
f. System/Software Requirements Specification(s).
g. Approved Software Test Plan (STP) under configuration control.
h. Approved Software Test Environment (STE) specifications (as appropriate).
i. STD under configuration control.
j. PQE Process Control Document

ENTRY CRITERIA:
STD completed and ready for review.

OUTPUTS:
STD review process checklist and review comments form (See table 1).

EXIT CRITERIA:
STD approved and placed under configuration control.

VERIFIABLE OBJECTIVE EVIDENCE:
a. Completed copy of PQE Software Test Design Review process, ST-004.
b. Completed STD review comments forms.
c. Approved STD under configuration control.
d. Completed PQE metrics collection spreadsheet.

SUGGESTED METRICS:
a. Level of effort in labor hours by skill levels (low, mid, high).
b. Number of STD defects, by type and severity.

TOOLS/FORMS/CHECKLISTS:
a. PQE metrics collection spreadsheet.
b. PQE Software Test Description Review Process, ST-004.
c. STD review comment form (See table 1).

DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.

REFERENCE DOCUMENTS:
a. Integration and Run Time Specification (I&RTS), Version 3.0, Joint Interoperability and Engineering Organization, July 1997.
b. User Interface Specifications for the Defense Information Infrastructure (DII), Version 2.0, 1 April 1996.
c. PQE Process Control Document, Version 1.5, 7 June 1999.

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Software Test Design (STD) Review

PROCESS NUMBER: ST-004	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	ECP	Engineering Change Proposal
7.	GUI	Graphical User Interface
8.	I&RTS	Integration and Runtime Specification
9.	PCD	Process Control Document
10.	PQE	Product Quality Engineering
11.	RTM	Requirement Traceability Matrix
12.	SCTP	GUI Style Compliance Test Protocol
13.	SDP	Software Development Plan
14.	SEPO	Software Engineering Process Office
15.	SPCR	Software Problem Change Request
16.	SQA	Software Quality Assurance
17.	SSC SD	Space and Naval Warfare Systems Center, San Diego
18.	STD	Software Test Design
19.	STE	Software Test Environment
20.	STP	Software Test Plan
21.	STPR	Software Test Procedure
22.	STR	Software Test Report
23.	SVD	Software Version Description
24.	TEMP	Test & Evaluation Master Plan

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Design (STD) Review			
PROCESS NUMBER: ST-004	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
1	**Evaluate the entry criteria to determine the STD readiness for review:** •STD is complete, baselined and under configuration control. Note: If the entry criteria have not been met, the STD review cannot be conducted at this time. Correct the deficiencies and schedule the review for another time.	
2	**Verify availability of required references to support the STD review:** a. Tasking agreement or Contract. b. Statement of Work. c. SDP. d. TEMP. e. Other project plans, as applicable. f. System/Software Requirements Specification(s). g. STP. h. STE specifications.	
3	**Assign STD review participants:** a. Two or more senior Test Engineers b. At least one senior System Engineer or S/W Engineer c. Test specialists, as required Note: Invite customer/client representatives to participate in the review, as appropriate.	
4	**Obtain copies of the STD Review Comments Form and appropriate PQE metrics collection spreadsheet.**	
5	**Initiate STD review.** If review is to be conducted formally: a. Assign an experienced/trained moderator to organize and conduct the review. b. Assign a scribe to document comments. c. Provide the moderator with the required STD review materials and STD review data sheets. d. Instruct the moderator to prepare for, and conduct, the review.	
	If review is to be conducted informally: c. Assign an STD review team leader. d. Distribute the review materials to the participants with instructions to complete the STD review comments and return the materials to the team leader.	

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Software Test Design (STD) Review			
PROCESS NUMBER: ST-004	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	**ACTIVITIES**	**DATE**
	For formal and informal reviews: d. Instruct the STD author to incorporate straightforward comments, resolve any conflicting or seemingly inappropriate comments with the moderator or team leader, and initial and date all resolved comments. e. Moderator/team leader reviews the updated STD and review comment forms. Where comments are related to problems and issues requiring resolution, assign appropriate action items. f. Moderator/team leader fills out the PQE metrics collection spreadsheet and forwards to Configuration Management with the completed STD review checklist and review comment form.	
6	**Verify a completed copy of STD Review Checklist ST-004 and the completed STD review comment forms have been placed in the project files.**	

Moderator/team leader Signoff:		**Date:**
Test Manager Signoff:		**Date:**
Project Id:		

Table 1
STD Review Comment Form

Date:	Reviewer:		
Project Id:	Document ID:		
		Yes	No
1.	Does each test design trace to the test conditions/variations allocated in the Test Plan?		
2.	Does the test design satisfy all allocated test conditions/variations?		
3.	Is each test design implementable?		
4.	Is each test design logically executable?		
5.	Does the test design minimize duplication of tests?		
6.	Is the test design efficient?		
7.	Does the test design minimize dependence on data/conditions generated in previous tests?		
8.	Does each test design minimize execution time?		
9.	Does the test design minimize the effort required to restore the test environment if restart is necessary?		
10.	Do the test design execution methods validate the associated requirements and test conditions/variations?		
11.	Is there sufficient detail to fully implement the test design?		
12.	Has test procedure maintenance been considered?		
13.	Are verification methods well defined?		

Additional Comments:

ST-005

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Procedure (STPR) Development			
PROCESS NUMBER: ST-005	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

RELATED PROCESSES: ST-001, ST-003

PURPOSE:
Translate each software test case into an executable software test procedure based on the designated test types and test objectives described in the Software Test Design (STD). The step-by-step test procedure is designed to methodically exercise all requirements allocated to the test case through the full range of allowable data inputs.
Detailed test procedures are developed to assist in ensuring that adequate evaluation of the product delivered for testing will be scheduled. All reasonable efforts will be made by the PQE Group to evaluate the product under test as much as possible given the schedule and complexity of the required testing.
Refer to the PQE Process Control Document for a description of the Test Procedure Generation Priority Matrix and the depth/breadth methodology of testing that will be used by the PQE Group.

SPECIAL CONSIDERATIONS:
Development of executable software test procedures may require the creation of machine-readable test data files and data scripts, as well as updates to the STD and/or the Software Test Plan (STP).

RESPONSIBILITY:
The project Test Manager is responsible for ensuring complete and accurate development of the STPR.

INPUTS:
a. Tasking agreement or Contract.
b. Statement of Work.
c. Software Development Plan (SDP).
d. Test & Evaluation Master Plan (TEMP).
e. Other project plans, as applicable.
f. System/Software requirements specifications.
g. Approved STP under configuration control.
h. Approved STD under configuration control
i. Approved Software Test Environment (STE) specifications (as appropriate).
j. PQE Process Control Document

ENTRY CRITERIA:
STD is approved and baselined under configuration control.

OUTPUTS:
a. Completed STPR ready for review.
b. Updated STP and/or STD, as appropriate
c. Updated STE specification, as appropriate

EXIT CRITERIA:
Approved STPR under configuration control.

VERIFIABLE OBJECTIVE EVIDENCE:
a. Completed STPR development process document (ST-005).
b. Updated STP, as appropriate.
c. Updated STE specifications, as appropriate.
d. Updated STD, as appropriate.
e. Completed PQE metrics collection spreadsheet.

SUGGESTED METRICS:
a. Level of effort in labor hours by skill levels (low, mid, high).
b. Number of test procedures developed.

TOOLS/FORMS/CHECKLISTS:
PQE metrics collection spreadsheet (see PQE Process Control Document).

DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Software Test Procedure (STPR) Development			
PROCESS NUMBER: ST-005	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

REFERENCE DOCUMENTS:

a. Integration and Runtime Specification (I&RTS), Joint Interoperability and Engineering Organization.
b. User Interface Specifications for the Defense Information Infrastructure (DII).
c. PQE Process Control Document.

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	ECP	Engineering Change Proposal
7.	GUI	Graphical User Interface
8.	I&RTS	Integration and Runtime Specification
9.	PCD	Process Control Document
10.	PQE	Product Quality Engineering
11.	RTM	Requirement Traceability Matrix
12.	SCTP	GUI Style Compliance Test Protocol
13.	SDP	Software Development Plan
14.	SEPO	Software Engineering Process Office
15.	SPCR	Software Problem Change Request
16.	SQA	Software Quality Assurance
17.	SSC SD	Space and Naval Warfare Systems Center, San Diego
18.	STD	Software Test Design
19.	STE	Software Test Environment
20.	STP	Software Test Plan
21.	STPR	Software Test Procedure
22.	STR	Software Test Report
23.	SVD	Software Version Description
24.	TEMP	Test & Evaluation Master Plan

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Procedure (STPR) Development			
PROCESS NUMBER: ST-005	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
Step 1	**Review the requirements and conditions/restrictions/evaluation criteria allocated to the test cases in the STD.**	
Step 2	**Build a descriptive header for each test procedure based upon the corresponding test case in the approved STD. Refer to the PQE Process Control Document for the approved test procedure numbering system and document format guide.**	
Step 3	**Translate each test case from the approved STD into a series of executable test steps.** a. Ensure that test steps are clear and concise. b. Include sufficient detail to ensure procedure repeatability.	
Step 4	**Document the execution of each series of test steps.** a. Describe all test setup and test environment verification steps b. Identify the requirements and conditions being tested c. Identify the verification method to be used d. Identify the specific results expected, including values, accuracy, bounds, limits, duration/timing, pass/fail, etc. e. Identify the evaluation criteria for determining test step pass/fail results f. Document instructions for restarting the test, if necessary, i.e., must the entire test be re-executed or is there a way to safely restart (without jeopardizing overall test procedure verifiability) somewhere in the middle?	
Step 5	**Review procedure-specific support software requirements; i.e., drivers, stubs (temporary software modifications to emulate incomplete software , emulators/simulators, etc.** a. Have support requirements changed since review of the STP? STD? b. Is development/acquisition of support software on track?	
Step 6	**Initiate required support software corrective action, as appropriate.** a. Notify the responsible developer/tester of support software requirements changes. b. Submit appropriate change requests against the STP and/or STD to Configuration Management for processing.	
Step 7	**Repeat Steps 2 through 6 until test procedure development is completed.**	
Step 8	**Update the STP Requirement Traceability Matrix with STPR data.**	
Step 9	**Submit the STPR for peer review.**	
Step 10	**Make all required modifications and corrections to the STPR based upon review comments.**	
Step 11	**Submit the STPR for re-review, if the previous review identified:** a. One or more severe defects and/or b. A large number of other defects.	

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Procedure (STPR) Development			
PROCESS NUMBER: ST-005	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	**ACTIVITIES**	**DATE**
Step 12	**Repeat steps 10 and 11 until the STPR successfully completes the peer review process.**	
Step 13	**Forward the STPR to Configuration Management for placement under configuration control.**	

Test Manager Signoff:		**Date:**
SQA Signoff:		**Date:**
Project Id:		

ST-006

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Procedure (STPR) Review			
PROCESS NUMBER: ST-006	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**
RELATED PROCESSES: ST-005			
PURPOSE: Conduct a comprehensive technical evaluation to ensure the STPR is complete, correct, efficient, implementable, and traces to the allocated requirements.			
SPECIAL CONSIDERATIONS: None anticipated.			
RESPONSIBILITY/AUTHORITY: The project Test Manager is responsible for assigning experienced technical personnel to conduct a comprehensive STPR review and designating a review team leader or review moderator.			
INPUTS: a. Tasking agreement or Contract. b. Statement of Work. c. Software Development Plan (SDP). d. Test and Evaluation Master Plan (TEMP). e. Other project plans, as applicable. f. System/Software requirements specifications. g. Approved Software Test Plan (STP) under configuration control. h. Approved Software Test Description (STD) under configuration control i. Approved Software Test Environment (STE) specifications (as appropriate). j. STPR under configuration control. k. PQE Process Control Document			
ENTRY CRITERIA: STPR completed and ready for review.			
OUTPUTS: STPR review process checklist and review comments form (See table 1).			
EXIT CRITERIA: STPR approved and placed under configuration control.			
VERIFIABLE OBJECTIVE EVIDENCE: a. Completed PQE Software Test Procedure Review process, ST-006. b. Completed STPR review comments forms. c. Approved STPR under configuration control. d. Completed PQE metrics collection spreadsheet.			
SUGGESTED METRICS: a. Level of effort in labor hours by skill levels (low, mid, high). b. Number of STPR defects, by type and severity.			
TOOLS/FORMS/CHECKLISTS: a. PQE metrics collection spreadsheet. b. PQE Software Test Procedure Review process, ST-006. c. STPR review comment form (See table 1).			
DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.			
REFERENCE DOCUMENTS: a. Integration and Run Time Specification (I&RTS), Version 3.0, Joint Interoperability and Engineering Organization, July 1997. b. User Interface Specifications for the Defense Information Infrastructure (DII), Version 2.0, 1 April 1996. c. PQE Process Control Document, Version 1.5, 7 June 1999.			

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Procedure (STPR) Review			
PROCESS NUMBER: ST-006	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	ECP	Engineering Change Proposal
7.	GUI	Graphical User Interface
8.	I&RTS	Integration and Runtime Specification
9.	PCD	Process Control Document
10.	PQE	Product Quality Engineering
11.	RTM	Requirement Traceability Matrix
12.	SCTP	GUI Style Compliance Test Protocol
13.	SDP	Software Development Plan
14.	SEPO	Software Engineering Process Office
15.	SPCR	Software Problem Change Request
16.	SQA	Software Quality Assurance
17.	SSC SD	Space and Naval Warfare Systems Center, San Diego
18.	STD	Software Test Design
19.	STE	Software Test Environment
20.	STP	Software Test Plan
21.	STPR	Software Test Procedure
22.	STR	Software Test Report
23.	SVD	Software Version Description
24.	TEMP	Test & Evaluation Master Plan

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Software Test Procedure (STPR) Review			
PROCESS NUMBER: ST-006	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

STEP	ACTIVITIES	DATE
1	**Evaluate the entry criteria to determine the STPR readiness for review:** •STPR is complete, baselined and under configuration control. Note: If the entry criteria have not been met, the STPR review cannot be conducted at this time. Correct the deficiencies and schedule the review for another time.	
2	**Verify availability of required references to support the STPR review:** a. Tasking agreement or Contract. b. Statement of Work. c. SDP. d. TEMP. e. Other project plans, as applicable. f. System/Software Requirements Specification(s). l. STP. g. STE specifications. h. STD.	
3	**Assign STPR review participants:** a. Two or more senior Test Engineers b. At least one senior System Engineer or S/W Engineer c. Test specialists, as required Note: Invite customer/client representatives to participate in the review, as appropriate.	
4	**Obtain copies of the STPR Review Comments Form and appropriate PQE metrics collection spreadsheet.**	
5	**Initiate STPR review.** If review is to be conducted formally: a. Assign an experienced/trained moderator to organize and conduct the review. b. Assign a scribe to document comments. c. Provide the moderator with the required STPR review materials and STPR review data sheets. d. Instruct the moderator to prepare for, and conduct, the review.	
	If review is to be conducted informally: a. Assign an STPR review team leader. b. Distribute the review materials to the participants with instructions to complete the STPR review comments and return the materials to the team leader.	

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Procedure (STPR) Review			
PROCESS NUMBER: ST-006	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

STEP	ACTIVITIES	DATE
	For formal and informal reviews: a. Instruct the STPR author to incorporate straightforward comments, resolve any conflicting or seemingly inappropriate comments with the moderator or team leader, and initial and date all resolved comments. b. Moderator/team leader reviews the updated STPR and review comment forms. Where comments are related to problems and issues requiring resolution, assign appropriate action items. c. Moderator/team leader fills out the PQE metrics collection spreadsheet and forwards to Configuration Management with the completed STPR review checklist and review comment form.	
6	**Verify a completed copy of the STPR Review Checklist ST-006 and the completed STPR review comment forms have been placed in the project files.**	

Moderator/team leader Signoff:		**Date:**
Test Manager Signoff:		**Date:**
Project Id:		

Table 1
STPR Review Comment Form

Date:	Reviewer:		
Project Id:	Document ID:		
		Yes	No
1.	Does each test procedure trace to the test conditions/variations allocated in the STD?		
2.	Does the test procedure correctly implement all allocated test conditions/variations defined in the STD?		
3.	Is each test procedure fully executable?		
4.	Are test steps fully documented?		
5.	Are instructions for execution clear and concise?		
6.	Does the test procedure minimize duplication of steps?		
7.	Does the test procedure minimize dependence on data/conditions generated in previous test steps?		
8.	Does the test procedure minimize execution time?		
9.	Does the implementation of test procedure minimize the effort required to restore the test environment if restart is necessary?		
10.	Does each logical series of steps in the test procedure culminate in an obvious pass/fail result?		
11.	Are expected results clearly documented?		

Additional Comments:

ST-007

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Software Test Report (STR) Preparation

PROCESS NUMBER: ST-007	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

RELATED PROCESSES: ST-001, ST-005

PURPOSE:
Prepare a STR that provides a detailed record of the testing performed on a software product. This process addresses the preparation of full and condensed test reports.

SPECIAL CONSIDERATIONS: None anticipated.

RESPONSIBILITY/AUTHORITY: The project Test Manager is responsible for developing the STR.

INPUTS:
a. Tasking agreement or Contract.
b. Statement of Work.
c. Software Development Plan (SDP).
d. T&E Master Plan (TEMP).
e. Other project plans, as applicable.
f. System/Software requirements specifications.
f. PQE Process Control Document
g. Updated defect tracking (SPCR) database.
h. Updated STD as required.
i. Updated STPR as required.
j. Updated STE specifications as required.
k. Record of testing conducted, certified by SQA.
l. PQE Process Control Document

ENTRY CRITERIA:
Completion of scheduled test activity certified by SQA.

OUTPUTS:
STR completed and ready for review.

EXIT CRITERIA:
STR approved and under configuration control.

VERIFIABLE OBJECTIVE EVIDENCE:
a. Completed Software Test Report (STR) Preparation process: (ST-007).
b. Completed PQE metrics collection spreadsheet.

SUGGESTED METRICS:
STR preparation effort in labor hours by skill level (low, mid, and high).

TOOLS/FORMS/CHECKLISTS:
PQE metrics collection spreadsheet

DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.

REFERENCE DOCUMENTS:
a. Integration and Run Time Specification (I&RTS), Version 3.0, Joint Interoperability and Engineering Organization, July 1997.
b. User Interface Specifications for the Defense Information Infrastructure (DII), Version 2.0, 1 April 1996.
c. PQE Process Control Document, Version 1.5, 7 June 1999.

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Software Test Report (STR) Preparation

PROCESS NUMBER: ST-007	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	ECP	Engineering Change Proposal
7.	GUI	Graphical User Interface
8.	I&RTS	Integration and Runtime Specification
9.	PCD	Process Control Document
10.	PQE	Product Quality Engineering
11.	RTM	Requirement Traceability Matrix
12.	SCTP	GUI Style Compliance Test Protocol
13.	SDP	Software Development Plan
14.	SEPO	Software Engineering Process Office
15.	SPCR	Software Problem Change Request
16.	SQA	Software Quality Assurance
17.	SSC SD	Space and Naval Warfare Systems Center, San Diego
18.	STD	Software Test Design
19.	STE	Software Test Environment
20.	STP	Software Test Plan
21.	STPR	Software Test Procedure
22.	STR	Software Test Report
23.	SVD	Software Version Description
24.	TEMP	Test & Evaluation Master Plan

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Software Test Report (STR) Preparation			
PROCESS NUMBER: ST-007	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
Step 1	**Determine the type of STR required for the completed test event (full or condensed). Refer to the PQE Process Control Document guidelines for selecting the type of test report.**	
Step 2	**Complete all required analysis of test results. The STPR provides directions for analyzing the test results and preparing the data for inclusion in the test report.**	
Step 3	**Review the record of testing conducted and the defect tracking (SPCR) database. A failed test step shall be documented with a corresponding SPCR.**	
Step 4	**Develop the STR in accordance with the documentation standard specified in the tasking agreement or contract.** Note: The SSCSD Software Engineering Process Office (SEPO) web page (http://sepo.spawar.navy.mil/) contains an STR template.	
Step 5	**Submit the STR for peer review.**	
Step 6	**Make all required modifications and corrections to the STR based upon peer review comments.**	
Step 7	**Submit the STR for re-review, if the previous review identified:** a. One or more severe defects and/or b. A large number of other defects.	
Step 8	**Repeat steps 6 and 7 until the STR successfully completes the peer review process.**	
Step 9	**Forward the STR to Configuration Management for placement under configuration control.**	

Test Manager Signoff:		**Date:**
SQA Signoff:		**Date:**
Project Id:		

ST-008

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**
PROCESS: Software Test Report (STR) Review
PROCESS NUMBER: ST-008 | **REV./CHG: 0** | **SUPERSEDES:** | **EFFECTIVE DATE: 9/1/99**
RELATED PROCESSES: ST-005
PURPOSE: Conduct a comprehensive technical evaluation to ensure that the full or condensed STR is complete, correct and accurately describes the results of the tests conducted.
SPECIAL CONSIDERATIONS: None anticipated.
RESPONSIBILITY/AUTHORITY: The project Test Manager is responsible for assigning experienced technical personnel to conduct a comprehensive STR review and designating a review team leader or review moderator.
INPUTS: a. Tasking agreement or Contract. b. Statement of Work. c. Software Development Plan (SDP). d. Test and Evaluation Master Plan (TEMP). e. Other project plans, as applicable. f. System/Software requirements specifications. g. Approved Software Test Plan (STP) under configuration control. h. Approved Software Test Description (STD) under configuration control i. Approved Software Test Environment (STE) specifications (as appropriate). j. Approved Software Test Procedures (STPR) under configuration control. k. STR under configuration control l. SQA record of testing conducted. m. Analysis of test results (as appropriate). n. PQE Process Control Document
ENTRY CRITERIA: STR completed and ready for review.
OUTPUTS: STR review process checklist and review comment form (See table 1).
EXIT CRITERIA: STR approved and placed under configuration control.
VERIFIABLE OBJECTIVE EVIDENCE: a. Completed PQE Software Test Report Review process, ST-008. b. Completed STR review comment forms. c. Approved STR under configuration control. d. Completed PQE metrics collection spreadsheet.
SUGGESTED METRICS: a. Level of effort in labor hours by skill levels (low, mid, high). b. Number of STR defects, by type and severity.
TOOLS/FORMS/CHECKLISTS: a. PQE metrics collection spreadsheet. b. PQE Software Test Report Review process, ST-008. c. STR review comment form (See table 1).
DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.
REFERENCE DOCUMENTS: a. Integration and Run Time Specification (I&RTS), Version 3.0, Joint Interoperability and Engineering Organization, July 1997. b. User Interface Specifications for the Defense Information Infrastructure (DII), Version 2.0, 1 April 1996. c. PQE Process Control Document, Version 1.5, 7 June 1999.

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Software Test Report (STR) Review			
PROCESS NUMBER: ST-008	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	ECP	Engineering Change Proposal
7.	GUI	Graphical User Interface
8.	I&RTS	Integration and Runtime Specification
9.	PCD	Process Control Document
10.	PQE	Product Quality Engineering
11.	RTM	Requirement Traceability Matrix
12.	SCTP	GUI Style Compliance Test Protocol
13.	SDP	Software Development Plan
14.	SEPO	Software Engineering Process Office
15.	SPCR	Software Problem Change Request
16.	SQA	Software Quality Assurance
17.	SSC SD	Space and Naval Warfare Systems Center, San Diego
18.	STD	Software Test Design
19.	STE	Software Test Environment
20.	STP	Software Test Plan
21.	STPR	Software Test Procedure
22.	STR	Software Test Report
23.	SVD	Software Version Description
24.	TEMP	Test & Evaluation Master Plan

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**				
PROCESS: Software Test Report (STR) Review				
PROCESS NUMBER: ST-008	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**	

STEP	ACTIVITIES	DATE
1	**Evaluate the entry criteria to determine the STR readiness for review:** •STR is complete, baselined and under configuration control. Note: If the entry criteria have not been met, the STR review cannot be conducted at this time. Correct the deficiencies and schedule the review for another time.	
2	**Verify availability of required references to support the STR review:** a. Tasking agreement or Contract. b. Statement of Work. c. SDP. d. TEMP. e. Other project plans, as applicable. f. System/Software Requirements Specification(s). o. STP. g. STE specifications. h. STD. i. STPR. j. SQA certified record of tests conducted.	
3	**Assign STR review participants:** a. Two or more senior Test Engineers b. At least one senior System Engineer or S/W Engineer c. Test specialists, as required	
4	**Obtain copies of the STR Review Comments Form and appropriate PQE metrics collection spreadsheet.**	
5	**Initiate STR review.** If review is to be conducted formally: a. Assign an experienced/trained moderator to organize and conduct the review. b. Assign a scribe to document comments. c. Provide the moderator with the required STR review materials and STR review data sheets. d. Instruct the moderator to prepare for, and conduct, the review.	
	If review is to be conducted informally: a. Assign an STR review team leader. b. Distribute the review materials to the participants with instructions to complete the STR review comments and return the materials to the team leader.	

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Software Test Report (STR) Review			
PROCESS NUMBER: ST-008	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

STEP	**ACTIVITIES**	**DATE**
	For formal and informal reviews: a. Instruct the STR author to incorporate straightforward comments, resolve any conflicting or seemingly inappropriate comments with the moderator or team leader, and initial and date all resolved comments. b. Moderator/team leader reviews the updated STR and review comment forms. Where comments are related to problems and issues requiring resolution, assign appropriate action items. c. Moderator/team leader fills out the PQE metrics collection spreadsheet and forwards to Configuration Management with the completed STR review checklist and review comment form.	
6	**Verify a completed copy of the STR Review Checklist ST-008 and the completed STR review comment forms have been placed in the project files.**	

Moderator/team leader Signoff:		**Date:**
Test Manager Signoff:		**Date:**
Project Id:		

Table 1
STR Review Comment Form

Date:	Reviewer:		
Project ID:	Document ID:		
		Yes	No
1.	Is the SQA certified record of testing conducted included in the STR?		
2.	Is the required analysis of test results complete and accurately described in the STR?		
3.	Is each test failure documented in an SPCR?		
4.	Are deviations from the approved and baselined STPR identified during the test adequately described in the STR?		
5.	Are all software test environment problems identified during the test adequately described in the STR?		
6.	Is the overall assessment of the product tested adequately described in the STR?		
7.	Are recommended product improvements adequately described in the STR?		

Additional Comments:

ST-009

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Integration Testing			
PROCESS NUMBER: ST-009	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**
RELATED PROCESSES: ST-001, ST-003 and ST-005			
PURPOSE: Integration testing demonstrates the system components link and work together correctly and identifies interface defects between both internal subsystems and external systems. Integration testing is conducted on each software build to verify the integration of subsystems is functionally correct. Integration testing is conducted iteratively using a collection of functional tests with emphasis on application software interoperability. Integration testing will consist of appropriately selected subsets of existing tests to ensure that new integrated software subsystems have not disrupted previously integrated functions, and new functional test procedures to validate new interfaces and interoperative functions (system functions that require the interaction of more than one system element).			
SPECIAL CONSIDERATIONS: None anticipated.			
RESPONSIBILITY/AUTHORITY: The project Test Manager is responsible for scheduling and conducting Integration Testing.			
INPUTS: a. Approved STP under configuration control. b. Approved STD under configuration control. c. Approved Integration STPR under configuration control. d. Current Software Problem Change Request (SPCR) database. e. Approved Software Test Environment (STE) specifications (as appropriate). f. Software build plan and schedule. g. Baseline application software under configuration control. h. Baseline application Software Version Description (SVD) under configuration control. i. Test data to support integration testing of the baseline application software build. j. PQE Process Control Document.			
ENTRY CRITERIA: a. Successful completion of developer testing of the integrated software products. b. Notification the integrated baseline application software under configuration control is ready for integration testing. c. STE is scheduled and configured to support the integration tests. d. The required documentation, tools, drivers, test data, and scenarios to support integration tests are available.			
OUTPUTS: a. Updated SPCR database including problem reports identified during integration testing. b. Updated STD as required. c. Updated STPR as required. d. Updated STE specifications as required. e. Integration test record certified by SQA. f. Qualitative analysis report on the tested application software (see PQE Process Control Document).			
EXIT CRITERIA: a. Subsystem interfaces are tested satisfactory. b. No new high priority (1 or 2) defects are discovered during the test. c. All scheduled tests are executed and results documented in an SQA certified test record.			
VERIFIABLE OBJECTIVE EVIDENCE: a. Completed copy of Integration Testing Process: ST-009. b. Record of tests conducted certified by SQA. c. Completed STPR. d. Completed PQE metrics collection spreadsheet.			

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Integration Testing

PROCESS NUMBER: ST-009	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

SUGGESTED METRICS:
a. Level of test effort in labor hours by skill levels (low, mid, high).
b. Number of tests conducted.
c. Number of system/software interface requirements tested.
d. Number of deviations from test procedure per test conducted
e. Number of defects by priority per test conducted.

TOOLS/FORMS/CHECKLISTS:
a. PQE metrics collection spreadsheet (see PQE Process Control Document).
b. Record of tests conducted worksheet (see PQE Process Control Document).
c. Qualitative analysis report template (see PQE Process Control Document).

DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.

REFERENCE DOCUMENTS:
a. Integration and Runtime Specification (I&RTS), Joint Interoperability and Engineering Organization.
b. User Interface Specifications for the Defense Information Infrastructure (DII).
f. PQE Process Control Document.

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	DT	Development Testing
7.	ECP	Engineering Change Proposal
8.	GUI	Graphical User Interface
9.	I&RTS	Integration and Runtime Specification
10.	PCD	Process Control Document
11.	PQE	Product Quality Engineering
12.	RTM	Requirement Traceability Matrix
13.	SCTP	GUI Style Compliance Test Protocol
14.	SDP	Software Development Plan
15.	SEPO	Software Engineering Process Office
16.	SPCR	Software Problem Change Request
17.	SQA	Software Quality Assurance
18.	SSC SD	Space and Naval Warfare Systems Center, San Diego
19.	STD	Software Test Design
20.	STE	Software Test Environment
21.	STP	Software Test Plan
22.	STPR	Software Test Procedure
23.	STR	Software Test Report
24.	SVD	Software Version Description
25.	TEMP	Test & Evaluation Master Plan

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Integration Testing			
PROCESS NUMBER: ST-009	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

STEP	ACTIVITIES	DATE
Step 1	**Review the approved STP to identify the interface requirements that will be scheduled for testing.**	
Step 2	**Identify the Integration STPR appropriate for this test.** Note: There may be only one increment. The plan may be to test the integrated system and its interface requirements all at once.	
Step 3	**Verify that each Integration STPR scheduled to be conducted has successfully completed peer review and dry run (to the extent possible) to ensure that no run time problems exist in the procedures, test setup and/or data.** •Defer the Integration test procedures that have not completed peer review. •If run-time problems can be easily corrected without invalidating some/all of test: -Mark up the STPR with the appropriate corrections. -Make all corrections to the STE needed and note all changes in the test record. •If run-time problems are more serious, defer the execution of those Integration tests until such time as appropriate corrections can be made and the STPR updated, peer reviewed and dry-run. •Ensure that the project Test Manager approves all changes. •Notify the Account Manager and/or PQE Manager of required changes.	
Step 4	**Verify that adequate test time is available to complete the Integration test sequence or that continuation (restart) of a test into another test period is feasible.** •If insufficient time is available to conduct the scheduled tests, rescheduled some or all of the tests for another time period. •Ensure scheduling approval of the project Test Manager. •Notify the Account Manager and/or PQE Manager of test scheduling changes.	
Step 5	**Verify that the required software build(s), test data, external systems, communications links, etc. is available.**	
Step 6	**Initiate and maintain a test record of Integration test activities, results, problems, errors, and anomalies that occur during the tests. Include complete identification of test personnel, STPR, and the test environment (both hardware and software).**	

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Integration Testing

PROCESS NUMBER: ST-009	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

Step 7	**Conduct a pretest procedure to verify that the STE is appropriately configured to support the Integration tests.** Note: It is advisable to develop and maintain one test procedure for each test configuration that will be used to ensure that the required test setup and environment is correct before conducting any scheduled tests. • Document all discrepancies. • Notify the project Test Manager of any pre-test discrepancies uncovered or potential setup/configuration conflicts with other scheduled tests. • Attempt to correct the discrepancies. • Document the rationale for STE changes. • Defer the execution of any tests which cannot be executed due to unresolved discrepancies. • Ensure that the project Test Manager approves changes in test plans. • Notify the Account Manager and/or PQE Manager of changes in test plans. • If one or more integrated system components cannot be tested due to STE discrepancies, obtain project Test Engineering Manger approval to proceed. • If none of the scheduled tests can be executed, get the approval of the project Test Manager, notify the Account manager and/or PQE Manager then exit this process.	
Step 8	**Conduct Integration testing in accordance with the STPR:** a. If SQA is required to witness functional tests: • Ensure that SQA representative is informed of the test schedule before beginning testing. • Ensure that SQA representative has the latest version of each STPR to be conducted. • Ensure that SQA representative concurs with any changes made to the STPR and/or STE before proceeding. Note: SQA representative needs to be aware of any changes made to the STPR or the STE to correct for pre-test deficiencies uncovered and must agree that the validity of the particular tests has not been compromised. b. Enter the specified inputs into the system and observe the results. c. Document any STPR deviations. • Include the rationale for deviating and the impact of the deviation on the validity of the test. • All deviations should be approved by the project Test Manager. d. Record the test results from each series of test steps. Note: Document all symptoms related to obvious test discrepancies for subsequent analysis.	

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Integration Testing			
PROCESS NUMBER: ST-009	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 9/1/99**

Step 9	**Perform required test analysis or data reduction to determine test results, e.g., pass/fail as specified in the test procedure.** a. If the actual test results differ from the expected results: • Attempt to determine if the discrepancy is due to the STPR, the STE, or the application software under test. • Attempt to isolate the discrepancy to a particular software element. Note: Assistance from software engineering to isolate the cause of test procedure failure may be required. b. Document all errors as SPCRs. Ensure that the SPCR contains sufficient information for a software engineer to duplicate and analyze the problem.	
Step 10	**Submit new SPCRs to Configuration Management for processing.**	

Test Director Signoff:		**Date:**
SQA Signoff:		**Date:**
Project Id:		

ST-010

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Defense Information Infrastructure (DII) Common Operating Environment (COE) Compliance Testing

PROCESS NUMBER: ST-010	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

RELATED PROCESSES: ST-001, ST-003 and ST-005

PURPOSE: The purpose of DII COE compliance testing is to validate the application software:

a. Conforms to rules, standards and specifications identified in the current Integration and Runtime Specification (I&RTS) for the DII COE.
b. Is ready for DII COE integration.

DII COE compliance testing is conducted following successful completion of software integration testing. DII COE compliance testing does **not** verify compliance with system/software performance requirements.

SPECIAL CONSIDERATIONS: None anticipated.

RESPONSIBILITY/AUTHORITY: The project Test Manager is responsible for scheduling and conducting DII COE Compliance Testing.

INPUTS:

a. Approved Software Test Plan (STP) under configuration control.
b. Approved Software Test Design (STD) under configuration control.
c. Approved DII COE Compliance Software Test Procedure (STPR) under configuration control.
d. Current Software Problem Change Request (SPCR) database.
e. Approved Software Test Environment (STE) specifications (as appropriate).
f. Software build plan and schedule.
g. Baseline application software under configuration control.
h. Baseline application Software Version Description (SVD) under configuration control.
i. Test data to support DII COE compliance testing of the baseline application software build.
j. Compliance Level Certification from the software development organization.
k. Product Quality Engineering (PQE) Process Control Document.

ENTRY CRITERIA:

a. Successful completion of integration testing of the application software.
b. Notification that the integrated baseline application software under configuration control is ready for DII COE Compliance Testing.
c. STE is scheduled and configured to support DII COE Compliance Testing.
d. The documentation, tools, drivers and test data required to support DII COE Compliance Testing are available.
e. Developer has successfully completed preliminary I&RTS validation testing (at level 5, 6 or 7) as specified by the Account Manager.

OUTPUTS:

a. Updated SPCR database including problem reports identified during DII COE compliance testing.
b. Updated STD as required.
c. Updated DII COE Compliance STPR as required.
d. Updated STE specifications as required.
e. DII COE Compliance test record certified by SQA.
f. Qualitative analysis report on the tested application software (see PQE Process Control Document).

EXIT CRITERIA/NEXT PROCEDURE:

a. I&RTS rules, standards and specifications are tested satisfactory.
b. No new high priority (1 or 2) defects are discovered during the test.
c. All scheduled DII COE compliance tests are completed and results documented in an SQA certified test record.

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Defense Information Infrastructure (DII) Common Operating Environment (COE) Compliance Testing

PROCESS NUMBER: ST-010	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

VERIFIABLE OBJECTIVE EVIDENCE:
a. Completed copy of DII COE Compliance Testing Process: ST-010.
b. Record of DII COE Compliance Testing conducted certified by SQA.
c. Completed DII COE Compliance Testing procedures.
d. Completed PQE metrics collection spreadsheet.

SUGGESTED METRICS:
a. Level of test effort in labor hours by skill levels (low, mid, high).
b. Number of tests conducted.
c. Number of I&RTS rules, standards and specifications tested.
d. Number of deviations from test procedure per test conducted
e. Number of defects by priority per test conducted.

TOOLS/FORMS/CHECKLISTS:
a. ChkCompliance Tool (CCT).
b. GUI Style Compliance Test Protocol (SCTP).
c. PQE metrics collection spreadsheet (see PQE Process Control Document).
d. Record of tests conducted worksheet (see PQE Process Control Document).
e. Qualitative analysis report template (see PQE Process Control Document).

DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.

REFERENCE DOCUMENTS:
a. Integration and Runtime Specification (I&RTS), Joint Interoperability and Engineering Organization.
b. User Interface Specifications for the Defense Information Infrastructure (DII).
c. PQE Process Control Document.

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	DT	Development Testing
7.	ECP	Engineering Change Proposal
8.	GUI	Graphical User Interface
9.	I&RTS	Integration and Runtime Specification
10.	PCD	Process Control Document
11.	PQE	Product Quality Engineering
12.	RTM	Requirement Traceability Matrix
13.	SCTP	GUI Style Compliance Test Protocol
14.	SDP	Software Development Plan
15.	SEPO	Software Engineering Process Office
16.	SPCR	Software Problem Change Request
17.	SQA	Software Quality Assurance
18.	SSC SD	Space and Naval Warfare Systems Center, San Diego
19.	STD	Software Test Design
20.	STE	Software Test Environment
21.	STP	Software Test Plan
22.	STPR	Software Test Procedure
23.	STR	Software Test Report
24.	SVD	Software Version Description
25.	TEMP	Test & Evaluation Master Plan

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**		
PROCESS: Defense Information Infrastructure (DII) Common Operating Environment (COE) Compliance Testing		
PROCESS NUMBER: ST-010 | **REV./CHG: 0** | **SUPERSEDES:** | **EFFECTIVE DATE: 6/3/99**		
STEP	**ACTIVITIES**	**DATE**
Step 1	**Review the current I&RTS rules, standards and specifications applicable to the application software under test.** Note: The purpose of this review is to ensure that I&RTS rules, standards and specifications not applicable to the software under test are not included in the compliance test and that all required I&RTS criteria are included in the scheduled test. - Ensure that DII COE Compliance Testing STPR is reviewed, approved and dry run prior to conducting the formal tests. - Ensure project Test Manager and Account Manager approval of the STPR prior to test execution. - Ensure that the DII COE Compliance Testing STPR clearly demonstrate agreed upon I&RTS requirements.	
Step 2	Review the Compliance Level Certification provided by the software development organization for conformance and completeness with the I&RTS rules, standards and specifications defined in Step 1.	
Step 3	**Develop a detailed schedule for DII COE Compliance Testing. Identify:** • Specific dates, time frames and locations for testing • Specific tests to be executed • A specific sequence for test execution • Specific staff DII COE Compliance Testing responsibilities.	
Step 4	**Obtain Account Manager concurrence with the proposed DII COE Compliance Testing schedule.**	
Step 5	**Designate members of the project team to perform specific support roles during DII COE Compliance testing, e.g., test operators, hardware support specialists, communications specialists, systems administrators, senior software engineers, etc.**	
Step 6	**Prior to beginning DII COE Compliance Testing conduct a pretest procedure to verify that the STE is properly configured to support the planned tests.** • Document all STE discrepancies. • Notify the project Test Manager of STE discrepancies. • Correct or find a circumvention for STE discrepancies. • Document the rationale for any changes from the planned STE. • Notify the project Test Manager, PQE Manager and/or Account Manager of proposed changes in the STE. Obtain approval for the proposed STE changes.	
Step 7	**At the start of testing, verify the participation of all required test and support personnel, including the SQA representative.**	

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes		
PROCESS: Defense Information Infrastructure (DII) Common Operating Environment (COE) Compliance Testing		
PROCESS NUMBER: ST-010	**REV./CHG: 0**	**SUPERSEDES:** **EFFECTIVE DATE: 6/3/99**
STEP	**ACTIVITIES**	**DATE**
Step 8	**Conduct a pretest brief to ensure that all participants understand the planned sequence of DII COE Compliance Test events.** • Review the purpose of the test and the general approach to DII COE Compliance testing. • Review the planned sequence of test events. • Review any STPR changes. • Ensure that each participant has the latest version of the DII COE STPR. Note: If DII COE Compliance tests are to be conducted over several days conduct a prebrief at the beginning of each day to review completed testing and describe the testing to be accomplished that day. Also note that errors uncovered during previous test sessions may make changes in the schedule necessary. In such cases, the pretest brief becomes very important to ensure that all test participants understand the plan of action.	
Step 9	**Initiate and maintain a test log of all test activities, results, problems, errors, and anomalies that occur throughout the DII COE Compliance test. Include complete identification of test personnel, STPR and the STE configuration (both hardware and software).**	
Step 10	**Load the system build image into the test system.** Note: The SQA representative will witness Configuration Management providing the test organization with the configuration controlled system build image and the test team installing the software.	
Step 11	**Conduct the DII COE Compliance test(s) as specified in the STPR.** a. The test manager will describe the current series of test steps to be conducted. b. Designated test operators will execute the test steps. c. The SQA representative will verify the test operators conduct the DII COE Compliance testing as described in the STPR and the results of test activities are accurately recorded, including deviations from the STPR, test discrepancies and the pass/fail results of test activities.	
Step 12	**At the end of each test session, conduct a debrief with the project Test Manager, SQA and CM as a minimum, to summarize and ensure consensus on test results.**	
Step 13	**Perform any required test analysis or data reduction. Evaluate test results against pass/fail criteria specified in the STPR.**	
Step 14	**Document any unresolved test discrepancies as SPCRs. Submit SPCRs to CM for processing.**	

Test Manager Signoff:		**Date:**
SQA Signoff:		**Date:**
Project Id:		

ST-011

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Functional Level Testing

PROCESS NUMBER: ST-011	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

RELATED PROCESSES: ST-005

PURPOSE: Functional testing verifies that the functional, performance and interface requirements allocated to the application software under test have been properly implemented. Functional testing consists of a collection of test scenarios that exercise various software capabilities, such as boundary condition, path, transaction flow, input validation and syntax, equivalence partitioning, database and state conditions.

SPECIAL CONSIDERATIONS: None anticipated.

RESPONSIBILITY/AUTHORITY: The project Test Manager is responsible for both scheduling and conducting Functional Testing.

INPUTS:
a. Approved STP under configuration control.
b. Approved STD under configuration control.
c. Approved Functional STPR under configuration control.
d. Current Software Problem Change Request (SPCR) database.
e. Approved Software Test Environment (STE) specifications (as appropriate).
f. Software build plan and schedule.
g. Baseline application software under configuration control.
h. Baseline application Software Version Description (SVD) under configuration control.
i. Test data to support functional testing of the baseline application software build.
j. PQE Process Control Document.

ENTRY CRITERIA:
a. The application software under test has successfully completed integration testing.
b. Notification that the integrated baseline application software under configuration control is ready for functional testing.
c. STE is scheduled and configured to support the functional tests.
d. The required documentation, tools, drivers, test data and scenarios are available to support functional tests.

OUTPUTS:
a. Updated SPCR database including problem reports identified during functional testing.
b. Updated STD, as required.
c. Updated STPR, as required.
d. Updated STE specifications, as required.
e. Functional test record certified by SQA.
f. Qualitative analysis report on the tested application software (see PQE Process Control Document).

EXIT CRITERIA:
a. Functional, performance and interface requirements are tested satisfactory.
b. No new high priority (1 or 2) defects are discovered during the test. A high priority defect disqualifies the software product from proceeding to the next phase of the development cycle.
c. All scheduled functional tests are completed and the results are documented in the SQA certified test record.

VERIFIABLE OBJECTIVE EVIDENCE:
a. Completed copy of Functional Level Testing Process: ST-011.
b. Record of functional tests conducted certified by SQA.
c. Completed functional STPR.
d. Completed PQE metrics collection spreadsheet.

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Functional Level Testing

PROCESS NUMBER: ST-011	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

SUGGESTED METRICS:
a. Level of test effort in labor hours by skill levels (low, mid, high).
b. Number of functional tests conducted.
c. Number of system/software functions tested.
d. Number of deviations from test procedure, per functional test conducted
e. Number of defects by priority, per functional test conducted.

TOOLS/FORMS/CHECKLISTS:
a. PQE metrics collection spreadsheet (see PQE Process Control Document).
b. Record of tests conducted worksheet (see PQE Process Control Document).
c. Qualitative analysis report template (see PQE Process Control Document).

DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.

REFERENCE DOCUMENTS:
a. Integration and Runtime Specification (I&RTS), Joint Interoperability and Engineering Organization.
b. User Interface Specifications for the Defense Information Infrastructure (DII).
c. PQE Process Control Document.

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	DT	Development Testing
7.	ECP	Engineering Change Proposal
8.	GUI	Graphical User Interface
9.	I&RTS	Integration and Runtime Specification
10.	PCD	Process Control Document
11.	PQE	Product Quality Engineering
12.	RTM	Requirement Traceability Matrix
13.	SCTP	GUI Style Compliance Test Protocol
14.	SDP	Software Development Plan
15.	SEPO	Software Engineering Process Office
16.	SPCR	Software Problem Change Request
17.	SQA	Software Quality Assurance
18.	SSC SD	Space and Naval Warfare Systems Center, San Diego
19.	STD	Software Test Design
20.	STE	Software Test Environment
21.	STP	Software Test Plan
22.	STPR	Software Test Procedure
23.	STR	Software Test Report
24.	SVD	Software Version Description
25.	TEMP	Test & Evaluation Master Plan

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Functional Level Testing			
PROCESS NUMBER: ST-011	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
Step 1	**Verify each of the functional test procedures scheduled to be conducted has successfully completed peer review and dry run execution, as necessary, to ensure that no run-time problems exist in the procedures, test setup and/or test data.** • Defer the execution of functional test procedures that have not completed peer review. • If run-time problems can be easily corrected without invalidating some or all of test: - Mark up the test procedures with the appropriate corrections. - Make all corrections to the STE needed and note all changes in the test log. • If run-time problems are more serious, defer the execution of those test procedures until such time as appropriate corrections can be made and the test procedure re-reviewed. • Ensure that the project Test Manager approves all changes. • Notify the PQE Manager and/or Account Manager of required STPR changes.	
Step 2	**Review the requirements of all scheduled tests. Review the pass/fail evaluation criteria for each requirement.**	
Step 3	**Verify that adequate test time is available to complete the entire test sequence or that continuation (restart) of a test into another test period is feasible.** • If insufficient time is available to conduct the scheduled tests, rescheduled some or all of the tests for another time frame. • Ensure project Test Manager concurrence. • Notify the PQE Manager and/or Account Manager of test schedule changes.	
Step 4	**Verify that the required software load build and test data are available.**	
Step 5	**Initiate and maintain a test log of all test activities, results, problems, errors, and anomalies that occur throughout the test conduct period. Include complete identification of test personnel, STPR and the STE (both hardware and software).**	

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**				
PROCESS: Functional Level Testing				
PROCESS NUMBER: ST-011	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**	

STEP	**ACTIVITIES**	**DATE**
Step 6	**Conduct a pretest to verify the STE is appropriately configured to support the scheduled functional tests.** • Document all discrepancies. • Notify the project Test Manager of any pre-test discrepancies uncovered or potential setup/configuration conflicts with other scheduled tests. • Attempt to correct all discrepancies. • Document the rationale for all changes. • Defer the execution of tests which cannot be executed due to unresolved discrepancies. • Ensure that the project Test Manager agrees with all changes in test plans. • Notify the PQE Manager and/or Account Manager of changes in test plans. • If none of the scheduled tests can be executed, notify the project Test Manager, PQE Manager and/or Account Manager and exit this procedure.	
Step 7	**Conduct functional testing in accordance with the STPR:** a. Ensure the SQA representative: • is informed of the test schedule before beginning testing. • has the latest version of each STPR to be conducted. • concurs with any changes made to the STPR and/or STE before proceeding. d. Enter the specified inputs into the system and observe the results. c. Document any deviation from the approved STPR. Include the rationale for deviating and the impact of the deviation on the validity of the test. d. Record the test results of each test step. Note: Document all symptoms related to obvious test discrepancies for subsequent analysis.	

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Functional Level Testing			
PROCESS NUMBER: ST-011	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
Step 8	**Perform any required test analysis to evaluate the test results, e.g., pass/fail as specified in the STPR.** a. If the actual test results differ from the expected results described in the STPR: • Attempt to determine whether the discrepancy is due to the STPR, the STE, or the application software under test. • Attempt to isolate the discrepancy to a particular software element. Note: Assistance from appropriate software engineers may be needed to isolate the cause of the failure. b. Document all errors as SPCRs. Ensure that the SPCR contains sufficient information for software engineering personnel to duplicate and analyze the problem. Note: Problems later isolated to tester, STPR, or STE problems will be closed out as test errors.	
Step 9	**Submit new SPCRs to Configuration Management for processing.**	

Test Manager Signoff:		**Date:**
SQA Signoff:		**Date:**
Project Id:		

ST-012

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Development Testing (DT)-I/II			
PROCESS NUMBER: ST-012	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

RELATED PROCESSES: ST-005, ST-009, ST-010, ST-011

PURPOSE:
Development testing (DT) is conducted in two phases.

DT-I level testing is conducted in a controlled laboratory environment with the emphasis on validation of system level requirements in support of product acceptance. DT-I level consists of integration, functional and compliance testing of the application software.

DT-II level testing is conducted at an operational site coordinated by the Account Manager of the application software under test. The emphasis of DT-II testing is to re-validate the system level requirements in an operational environment on the end-user's equipment.

See the PQE Process Control Document for a detailed description of DT level tests.

SPECIAL CONSIDERATIONS: Every effort should be made to schedule and conduct DT-II level testing on a not to interfere basis with the operational organization providing the site and equipment for testing. The validity of DT-II testing could be significantly impacted if it is conducted in a shared operational environment.

RESPONSIBILITY/AUTHORITY: The project Test Manager is responsible for conducting DT-II testing and is responsible for scheduling, coordination and conducting DT-I testing. The application software Account Manager is responsible for DT-II test site scheduling and coordination.

INPUTS:
a. Approved Software Test Plan (STP) under configuration control.
b. Approved Software Test Design (STD) under configuration control.
c. Approved Software Test Procedures (STPR) under configuration control.
d. Approved Software Test Environment (STE) specifications, as appropriate for DT-I/II level testing.
e. Current Software Problem Change Request (SPCR) database.
f. Baseline application software under configuration control.
g. Baseline application Software Version Description (SVD) under configuration control.
h. Test data to support DT-I/II testing of the baseline application software build, as appropriate.
i. Metrics and completion records of prior integration, functional and compliance testing conducted on the application software, as appropriate.
j. PQE Process Control Document.

ENTRY CRITERIA:
DT-I
a. Required preliminary testing (integration [ST-009], functional [ST-010] and/or compliance testing [ST-011]) on the application software is completed.
b. The STE is configured to support the scheduled DT-I testing
c. Notification that the application software is ready for DT-I testing.
d. Required test documentation, tools, drivers, test data, and scenarios are available to support DT-I testing.

DT-II
a. All required DT-I level testing on the application software is completed.
b. The application software Account Manager has scheduled the operational test site (STE) on a not to interfere basis and the site is available/configured to conduct the scheduled DT-II testing.
c. Required test documentation, tools, drivers, test data, and scenarios are available to support DT-II testing.

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Development Testing (DT)-I/II

PROCESS NUMBER: ST-012	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

OUTPUTS:
a. Updated SPCR database including problem reports identified during DT-I/II testing.
b. Updated STD as required.
c. Updated STPR as required.
d. Appropriate DT-I/II level test record certified by SQA.
e. Certified application software under configuration control.
f. Qualitative analysis report on the tested application software (see PQE Process Control Document).

EXIT CRITERIA:
a. System requirements are tested satisfactory.
b. No new high priority (1 or 2) SPCRs identified during the test.
c. All scheduled DT tests are completed satisfactory and results are documented in the SQA certified test record.

VERIFIABLE OBJECTIVE EVIDENCE:
a. Completed copy of DT-I/II Testing Process (ST-012).
b. Record of tests conducted certified by SQA.
c. Completed test procedures.
d. Completed PQE metrics collection spreadsheet.

SUGGESTED METRICS:
a. Level of test effort in labor hours by skill levels (low, mid, high).
b. Number of tests executed.
c. Number of deviations from test procedure per test conducted.
d. Number of new SPCRs, by priority per test conducted.

TOOLS/FORMS/CHECKLISTS:
a. PQE metrics collection spreadsheet (see PQE Process Control Document).
b. Record of tests conducted worksheet (see PQE Process Control Document).
c. Qualitative analysis report template (see PQE Process Control Document).

DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.

REFERENCE DOCUMENTS:
a. Integration and Runtime Specification (I&RTS), Joint Interoperability and Engineering Organization.
b. User Interface Specifications for the Defense Information Infrastructure (DII).
c. PQE Process Control Document.

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	DT	Development Testing
7.	ECP	Engineering Change Proposal
8.	GUI	Graphical User Interface
9.	I&RTS	Integration and Runtime Specification
10.	PCD	Process Control Document
11.	PQE	Product Quality Engineering
12.	RTM	Requirement Traceability Matrix
13.	SCTP	GUI Style Compliance Test Protocol
14.	SDP	Software Development Plan
15.	SEPO	Software Engineering Process Office
16.	SPCR	Software Problem Change Request
17.	SQA	Software Quality Assurance

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Development Testing (DT)-I/II			
PROCESS NUMBER: ST-012	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

18.	SSC SD	Space and Naval Warfare Systems Center, San Diego
19.	STD	Software Test Design
20.	STE	Software Test Environment
21.	STP	Software Test Plan
22.	STPR	Software Test Procedure
23.	STR	Software Test Report
24.	SVD	Software Version Description
25.	TEMP	Test & Evaluation Master Plan

SSC San Diego Product Quality Engineering (PQE) Test and Evaluation Processes			
PROCESS: Development Testing (DT)-I/II			
PROCESS NUMBER: ST-012	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
Step 1	**Verify readiness for conducting DT level testing** a. Integration, functional and/or compliance tests are complete in support of DT-I testing b. DT-I level testing is complete for entry into DT-II level testing	
Step 2	**Review the system requirements for all scheduled DT tests. Review the pass/fail evaluation criteria for each system requirement.**	
Step 3	**At DT-I level testing verify each of the DT test procedures scheduled to be conducted has successfully completed peer review and dry run execution, as necessary, to ensure that no run-time problems exist in the procedures, test setup and/or test data.** a. Defer the execution of DT test procedures that have not completed peer review or are dependent on DT test procedures that have not completed peer review. b. If run-time problems can be easily corrected without invalidating some or all of test: c. Mark up the test procedures with the appropriate corrections. d. Make all corrections to the STE needed and note all changes in the test log. e. If run-time problems are more serious, defer the execution of those DT test procedures until such time as appropriate corrections can be made and the test procedure re-reviewed. f. Ensure that the project Test Manager approves all changes. g. Notify the PQE Manager and/or Account Manager of required STPR changes.	
Step 4	**Verify that adequate time is available to complete the entire DT test sequence or that continuation (restart) of a test is feasible.** a. At DT-I if insufficient time is available to conduct the scheduled tests, notify the project test manager of the need to reschedule some or all of the tests. b. At DT-II if insufficient time is available to conduct the scheduled tests, notify the Account Manager of the need to re-schedule additional time at the operational site, if available. c. Ensure project Test Manager approves changes in test planning. d. Notify the PQE Manager and/or Account Manager of all DT test schedule changes.	
Step 5	**Initiate and maintain a test log of all DT test activities, results, problems, errors, and anomalies that occur during the DT test.**	
Step 6	**Install the configuration controlled application software build to conduct scheduled DT testing.**	

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Development Testing (DT)-I/II			
PROCESS NUMBER: ST-012	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
Step 7	**Conduct a pretest to verify the STE is appropriately configured to support the scheduled DT tests.** a. Document all discrepancies. b. Notify the project Test Manager of any pre-test discrepancies uncovered or potential setup/configuration conflicts with other scheduled tests. c. Attempt to correct STE discrepancies. d. Document the rationale for all STE changes. e. Defer the execution of tests that cannot be executed due to unresolved discrepancies. f. Obtain project Test Manager approval for changes in DT test planning. g. Notify the PQE Manager and/or Account Manager of changes in DT test planning. h. If none of the scheduled tests can be executed, notify the project Test Manager, PQE Manager and/or Account Manager and exit this procedure.	
Step 8	**Conduct DT testing in accordance with the STPR:** a. Ensure the SQA representative: • Is informed of the test schedule before beginning testing. • Has a copy of the latest version of each STPR to be conducted. b. Enter the specified test inputs into the system. Observe and compare the actual test results to the expected test results. c. Document any deviation from the approved STPR. Include the rationale for deviating and the impact of the deviation on the validity of the test. d. Record the test results of each test step. Note: Document all symptoms related to obvious test discrepancies for subsequent analysis.	
Step 9	**Perform required DT test analysis to evaluate the test results, e.g., pass/fail as specified in the STPR.** a. If the actual test results differ from the expected results described in the STPR: • Attempt to determine whether the discrepancy is due to the STPR, the STE, the operational test environment or the application software under test. • Attempt to isolate the discrepancy to a particular software element. b. Document all errors as SPCRs. Ensure that each SPCR contains sufficient information for software engineering personnel to duplicate and analyze the problem.	
Step 10	**Submit new SPCRs to Configuration Management for processing.**	

Test Manager Signoff:		**Date:**
Account Manager Signoff		**Date:**
SQA Signoff:		**Date:**
Project Id:		

ST-013

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Regression Testing

PROCESS NUMBER: ST-013	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

RELATED PROCESSES: ST-005, ST-009, ST-010, ST-011

PURPOSE:
The Regression Testing process addresses retest and regression test activities.

Retest is conducted following implementation of an approved system or software Engineering Change Proposal (ECP) or a fix that corrects a software problem.. Retesting ensures that the ECP function(s) have been correctly implemented, or the fix corrects the identified problem. Retesting is typically limited to the subset of the application software that is directly affected by the fix or the ECP function(s).

Regression testing consists of necessary:
1) Integration testing of subsequent application software builds (i.e., builds #2 through n) during an incremental development effort, or
2) Functional testing of a fully developed system to ensure that previously tested software has not been disturbed by the new increment, or any implemented engineering changes and/or fixes.

The level and scope of required regression testing is dependent on the maturity of the application software under test, the functional complexity of the new or modified software, the degree of coupling with other application software components, and the magnitude of the software change(s).

Regression testing involves the execution of various subsets and combinations of previously executed test procedures. Regression testing, based on the required depth/breadth of testing, may be conducted on:
1. a software build created from the developmental software library (informal engineering software release that, unless otherwise directed, does not require formal test records), or
2. a software build designated for formal testing which is created from the Configuration Management Master Software Library.

Regression and retest planning is an integral part of the application software change control process. Adequate time for proper retesting/regression testing must be factored into the final delivery schedule for the application software. Grouping software modifications to eliminate the overhead of additional builds and to consolidate the retesting/regression testing into a small number of actual tests is generally the most cost effective means of verifying software changes, but requires careful selection/development of test descriptions/test procedures to minimize duplication of effort.

SPECIAL CONSIDERATIONS: None anticipated.

RESPONSIBILITY/AUTHORITY: The project Test Manager is responsible for ensuring the scheduling and conduct of retesting and regression testing.

INPUTS:
a. Approved ECP to the system or software requirement specification.
b. Approved Software Test Plan (STP) under configuration control.
c. Approved Software Test design (STD) under configuration control.
d. Approved Software Test Procedure (STPR) under configuration control.
e. Current Software Problem Change Request (SPCR) database.
f. Approved Software Test Environment (STE) specifications (as appropriate).
g. Current software build plan and schedule.
h. Baseline application software under configuration control.
i. Baseline application Software Version Description (SVD) under configuration control.
j. Test data to support retesting and/or regression testing of the baseline application software.
k. Metrics of prior tests conducted on the application software.
l. PQE Process Control Document.

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Regression Testing

PROCESS NUMBER: ST-013	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

ENTRY CRITERIA:

a. Notification that the baseline application software under configuration control is ready for retesting and/or regression testing.
b. STE is scheduled and configured to support the retesting and/or regression testing.
c. The required documentation, tools, drivers, test data and scenarios are available to support retesting and/or regression testing.

OUTPUTS:

a. Updated SPCR database including problem reports identified during regression testing.
b. Updated STD, as required.
c. Updated STPR, as required.
d. Test record certified by SQA.
e. Certified application software.
f. Qualitative analysis report on the tested application software (see PQE Process Control Document).

EXIT CRITERIA:

a. Software fixes are verified to correct the known software defects.
b. ECP requirements are tested satisfactory.
c. No new high priority (1 or 2) defects are discovered during the test.
d. All scheduled tests are executed and results documented in an SQA certified test record.

VERIFIABLE OBJECTIVE EVIDENCE:

a. Completed copy of Regression Testing Process ST-013.
b. Record of tests conducted certified by SQA.
c. Completed test procedures.
d. Completed SPCR records.
e. Completed ECP records.
f. Completed PQE metrics collection spreadsheet.

SUGGESTED METRICS:

a. Level of test effort in labor hours by skill levels (low, mid, high).
b. Number of tests executed.
c. Number of system/software requirements tested.
d. Number of deviations from test procedure per test conducted
e. Number of corrected defects tested, by priority per test conducted.

TOOLS/FORMS/CHECKLISTS:

a. PQE metrics collection spreadsheet (see PQE Process Control Document).
b. Record of tests conducted worksheet (see PQE Process Control Document).
c. Qualitative analysis report template (see PQE Process Control Document).

DEVIATION/TAILORING: Deviation from this process is authorized at the direction of the PQE Manager.

REFERENCE DOCUMENTS:

a. Integration and Runtime Specification (I&RTS), Joint Interoperability and Engineering Organization.
b. User Interface Specifications for the Defense Information Infrastructure (DII).
c. PQE Process Control Document.

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Regression Testing

PROCESS NUMBER: ST-013	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

ACRONYMS:

1.	CCT	ChkCompliance Tool
2.	CHI	Computer Human Interface
3.	COE	Common Operating Environment
4.	COTS	Commercial Off The Shelf
5.	DII	Defense Information Infrastructure
6.	DT	Development Testing
7.	ECP	Engineering Change Proposal
8.	GUI	Graphical User Interface
9.	I&RTS	Integration and Runtime Specification
10.	PCD	Process Control Document
11.	PQE	Product Quality Engineering
12.	RTM	Requirement Traceability Matrix
13.	SCTP	GUI Style Compliance Test Protocol
14.	SDP	Software Development Plan
15.	SEPO	Software Engineering Process Office
16.	SPCR	Software Problem Change Request
17.	SQA	Software Quality Assurance
18.	SSC SD	Space and Naval Warfare Systems Center, San Diego
19.	STD	Software Test Design
20.	STE	Software Test Environment
21.	STP	Software Test Plan
22.	STPR	Software Test Procedure
23.	STR	Software Test Report
24.	SVD	Software Version Description
25.	TEMP	Test & Evaluation Master Plan

SSC San Diego
Product Quality Engineering (PQE)
Test and Evaluation Processes

PROCESS: Regression Testing

PROCESS NUMBER: ST-013	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
Step 1	**Review all approved SPCRs and ECPs for retest/regression test requirements.** • Ensure that the SPCRs/ECPs identify the system(s), configuration item(s), component(s), unit(s), any external interfaces, and all baseline documents affected by each change. • Review the current build plan and schedule.	
Step 2	**Determine the amount and scope of required retesting/regression testing.** • Assess the magnitude of the change(s) in the application software. Identify components with complex data structures or algorithms that have a high level of coupling to other components. Analyze test error metrics looking for trends or other indications that a component has a history of errors. • Identify components that interface directly or indirectly with the application software under test. Be especially aware of data that is updated or accessed from multiple control points or that is not readily visible to the tester. • Assess necessity for conducting stress and/or endurance tests following application software change validation. • Based on the above analysis, assess the scope of each required change (ECP or SPCR) relative to existing test procedures: - Identify any new test procedures that need to be developed. - Include any needed changes to test tools, drivers, and scenarios. • Identify the depth and breadth of testing required and which existing software test procedures need to be executed to adequately test the application software. - Ensure that the identified tests not only address the specific change but will address potential side affects associated with the change. - If existing procedures are not adequate, identify the scope of required changes. Review any available SPCR analysis to ensure no details are overlooked.	
Step 3	**Review the planned retesting/regression testing with the Account, Risk and Project Managers. Get retest/regression test planning approval from the managers.**	
Step 4	**Following approval of retest/regression test planning, make any necessary changes to test documentation and/or support software following the procedures approved for the specific level of test.** • Ensure that any required changes to test tools, drivers, and scenarios have been completed and verified. • Test documentation should follow the project standard review and approval process and the final product placed under configuration control.	
Step 5	**Load the system build containing the fixed and/or new application software.**	

SSC San Diego **Product Quality Engineering (PQE)** **Test and Evaluation Processes**			
PROCESS: Regression Testing			
PROCESS NUMBER: ST-013	**REV./CHG: 0**	**SUPERSEDES:**	**EFFECTIVE DATE: 6/3/99**

STEP	ACTIVITIES	DATE
Step 6	**Execute retest/regression tests using the designated test procedures.** Note: Minor deviations to the test sequence are at the discretion of the Test Director. Significant deviations, especially any that may impact test quality or schedule, should be brought to the immediate attention of the PQE and Risk Managers. • Record the test results on the test log, ensuring that an entry is made for each SPCR/ECP and that any deviations from the planned tests are documented. • Develop new SPCRs and/or ECPs to account for any new problems detected or additional changes recommended.	
Step 7	**Review all test results (logs and reports) to determine the status of regression testing.** • Ensure that all scheduled tests were executed. • Ensure that the status of all executed tests has been documented via a test report and updated ECP/SPCR status.	
Step 8	**Forward a copy of the test report, test log, updated SPCRs/ECPs and any new ECPs/SPCRs to Configuration Management.**	

Test Director Signoff:		**Date:**
SQA Signoff:		**Date:**
Project Id:		

APPENDIX E: OTHER FORMS

This appendix contains forms used by the PQE Group in developing and tracking software. These forms do not necessarily fall under the category of process documentation, but are integral to PQE Group functionality.

E-1 MEETING FORM

The PQE Group meets to discuss progress on software and documentation issues in the evolution of the program. This form is used to track notes, comments, and documentation revisions made or suggested during these meetings.

Meeting Minutes and Action Items

Title: ____________________ Date/Time: ________________ Location: ______________________

Item Type	Person Assigned	Description	Date Expected	Completed Y/N

Item Types:

AI = Action Item

E = Editorial comment

N = Note

E-2 EXAMPLE DEFECT DOCUMENTATION

The following paragraphs provide an example of the additional defect documentation. This example is based on a simulated problem that occurred when importing a picture file into a Word document.

E-2.1 Steps to Duplicate Problem

1. Open Word
2. Open the document, "PQE Members" located on the PQE server in the "members" directory
3. From the pull-down menu, select Insert -> Picture -> From File

4. Locate the file "Members.jpg" and click OK

At this point, the hard drive spins for several minutes. After that time, the "Blue Screen of Death" appears and the computer must be restarted.

E-2.2 Significant Environmental Conditions/Limitations

Although the computer in question has 64 MB of RAM, it was noted that several programs were open at the same time (Outlook, Internet Explorer) and that the unused hard drive space was less than 100 MB.

E-2.3 Problem Reasons and Likelihood

This problem is a direct result of low available memory in handling the JPEG image. Additionally, note that the image is several megabytes in size (9 MB).

It is highly likely that this problem will be encountered in an operational environment. Many documents are including images that require color and clarity, both of which require large amounts of memory.

E-2.4 Solution

There are three identified solutions to this problem. The first is to reduce the image in size. This requires a graphics program on the user's computer. The second solution is to close all open programs and work only in Word while importing the graphic. This second solution, while effective, interferes with the user's ability to receive email or access the Internet while working with this document and graphic. The third solution is to increase the amount of RAM available on the user's computer. This solution is more costly than the others, but it does have a positive impact on other applications run on the subject computer.

E-2.5 Impact

This problem does not directly interfere with the overall use of this software. While it does create problems in using the software with this document (and with similar documents using similar-size graphics), the overall impact is minimal. This is a Priority 3 problem.

E-2.6 Work-Around

Because there is a work-around to this problem, it is considered a Priority 3 problem. The work-around is to close all open programs and work only with Word until the graphic has been inserted and the document saved.

E-2.7 Conclusion

At this point, the documentation would be handed over to a second test engineer to undergo the peer review process as described below.

E-2.8 Defect Report Peer Review

Whenever possible, each defect report will be peer reviewed by a second test engineer. The following steps will be taken by the second engineer in a modified peer review format to validate the defect report.

1. Validate that the IEEE 12207 priority is valid
2. Confirm that the problem exists and can be duplicated
3. Confirm all other defect details, such as component, segment, and developer
4. Add name to description as a peer reviewer

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 2000	3. REPORT TYPE AND DATES COVERED Final
4. TITLE AND SUBTITLE PROCESS CONTROL DOCUMENT FOR THE SSC SAN DIEGO PRODUCT QUALITY ENGINEERING GROUP		5. FUNDING NUMBERS PE: OMN AN: IC000083 WU: CA60
6, AUTHOR(S) R. P. Stone		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) SSC San Diego San Diego, CA 92152–5001		8. PERFORMING ORGANIZATION REPORT NUMBER TD 3107
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Space and Naval Warfare Systems Command PMW 171 4301 Pacific Highway San Diego, CA 92110		10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES		
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This Process Control Document provides guidance within the SSC San Diego Product Quality Engineering (PQE) Group, and for outside groups, agencies, and developers, on the high-level processes that will be used by the PQE Group and all its team members. This creates an initial framework for starting a "repeatable level of statistical control by initiating rigorous project management." This document is not a testing handbook, but a guide to implement the various testing methods used in industry today.

14. SUBJECT TERMS Mission Area: Command, Control, Communications, Computers, and Intelligence software test and evaluation configuration management integrated verification/validation			15. NUMBER OF PAGES 144 16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAME AS REPORT

21a. NAME OF RESPONSIBLE INDIVIDUAL R. P. Stone	21b. TELEPHONE *(include Area Code)* (619) 553–4517 e-mail: stone@spawar.navy.mil	21c. OFFICE SYMBOL D4222

INITIAL DISTRIBUTION

D0012	Patent Counsel	(1)
D0271	Archive/Stock	(6)
D0274	Library	(2)
D027	M. E. Cathcart	(1)
D0271	D. Richter	(1)
D40	R. Smith	(1)
D42	R. Jaffee	(15)
D84	D. Milstead	(1)

Defense Technical Information Center
Fort Belvoir, VA 22060–6218 (4)

SSC San Diego Liaison Office
Arlington, VA 22202–4804

Center for Naval Analyses
Alexandria, VA 22302–0268

Navy Acquisition, Research and
Development Information Center
Arlington, VA 22202–3734

Government–Industry Data Exchange
Program Operations Center
Corona, CA 91718–8000

Space and Naval Warfare Systems Command
San Diego, CA 92110–3127