Functional Validation Testing

Component: Functional Validation Test

Component Description:

The Functional Validation Test is facilitated by test personnel from an independent test group within the design activity, and executed by people designated and empowered by the application Functional Project Officer (FPO). The primary purpose of the functional validation test is to verify that the software supports the critical requirements identified by the users. It is not intended to be a test to 'certify' the application, but is designed to ensure that the customer is satisfied with the software prior to integration into the DCII. It is considered to be a test event by the FPO representatives. The functional validation test is performed after the successful completion of the software integration test.

Inputs:

1) Application Test and Evaluation Master Plan (TEMP)
2) FVT schedule from the Software Development Plan
3) DCII Specification (Appendix D)
4) User specified functional requirements
5) Application Business Process Model (RD.010)
6) Application Business Function Model (RD.030)
7) Application Business Data Model (RD.040)
8) Module Functional documentation (MD.040)
9) Test artifacts from the Software Integration Test (SIT)
10) Release package
    a) Production database DDL (DB.050)
    b) Application Code (MD.090)
    c) Code Conversion Modules (CV.050)
    d) User Reference Document (DO.060)
    e) User Guide Document (DO.070)
    f) High-Level Technical Reference (DO.080)
    g) System Operations Guide (DO.090)
    h) Online Help Text (DO.140)
11) Approved Test Readiness Review (TRR) from application Software Integration Test

Outputs:

1) Functional Validation Software Test Plan (STP)
2) Test schedule
3) Documented test cases
4) Documented test scenarios / scripts
5) Test data
6) Interface output files
7) Test environment / databases
8) FVT Test discrepancy reports (TDRs)
9) FVT Test Analysis Report(s) for each test cycle
10) FVT Software Test Report (STR)
11) Completed Appendix D of DCII Specification relating to each application
12) Test metrics
13) Release Package (detail)

Task Ordering:

1) Strategy
2) Analysis
3) Design
4) Construct
5) Execute

Functional Validation Testing

Project Management Tasks:

1) Verify functional validation testing time identified in SDP
2) Ensure functional validation testing time is sufficient

Component Tasks:

Configuration Management Tasks:

1) Ensure functional validation testing environment is established
2) Verify Configuration Items are correct for functional validation testing

Component Metrics:

1) Data collected to determine the effort required to test the release using LRS:
   a) Estimated / actual hours to perform test strategy
   b) Estimated / actual hours to perform test analysis
   c) Estimated / actual hours to perform test design
   d) Estimated / actual hours to perform test construct
   e) Estimated / actual hours to perform test execute

2) Data collected to determine quality of the release:
   a) Using CMIS, the number of valid TDRs identified in the next level of testing for release (Enterprise Integration Test)
   b) Using Remedy, the number of valid trouble calls that result in an identified problem report that could have been found during testing (non-operator errors)

Controls:

1) DCII Standards (web site)
   a) Defense Corporate Information Infrastructure (DCII) specification
2) DII COE Standards (web site)
   a) DII Strategic Enterprise Architectures
   b) DII COE Interface and Run-Time Specification (I&RTS)
   c) DII User Interface Specification (UIS)
3) DFAS Information Technology Architecture
   a) Technical Architecture Framework for Information Management (TAFIM)
   b) JTA Standards
4) DoD Data Administration standards
   a) Defense Data Dictionary System (DDDS)
   b) Defense Finance and Accounting Data Model (DFADM)
5) Global Combat Support System (GCSS) (Interoperability)
6) IEEE/EIA J-STD-016
7) IEEE/EIA 12207

Task Name: Develop FVT Strategy **Task Number: T-FVT-001**
Component: Functional Validation Test **Category:** Software Engineering

1. **Task Name:** Develop Functional Validation Test Strategy

2. **Purpose:**

   Define the plan of action, scope, resources, personnel, testing methods, testing techniques, approach, and tools used to perform the functional validation of the application.

3. **Roles:**

   Developing the test strategy is the responsibility of the application test manager for the functional validation test.

4. **Entrance Criteria:**

   a. Application Test and Evaluation Master Plan (TEMP)
   b. FVT schedule from the Software Development Plan
   c. DCII Specification (Appendix D)
   d. User specified functional requirements
   e. Application Business Process Model (RD.010)
   f. Application Business Function Model (RD.030)
   g. Application Business Data Model (RD.040)
   h. Module Functional documentation (MD.040)
   i. Test artifacts from the Software Integration Test (SIT)
   j. Release package
   k. Approved Test Readiness Review (TRR) from application Software Integration Test

5. **Procedures:**

   Information gathered during the strategy task is documented in Sections 1.3, 1.4, 2.0, 3.0 and 4.1 of the Software Test Plan as noted in parentheses. To eliminate the number of software test plans for Functional Validation testing, only one plan will be developed for applications comprising of the DCD. Each application of the DCD will record specific application related information in the appropriate section within a subsection. A separate Functional Validation Test software test plan is required for each application interfacing with DCD, and the DCW.

   a. Review each application Test and Evaluation Master Plan (TEMP) for initial testing plans, schedules, and resources (Section 1.4)
      1) Test and evaluation resource summary
      2) Critical technical parameters
      3) Regulatory and statutory requirements
      4) Management roles and responsibilities

   b. Review the Software Development Plan for project schedules

   c. Review the test products (detail) from the Software Integration Test
      1) Review test cases for reusability
      2) Review test scripts for reusability
      3) Review test data for reusability

   d. Define the depth of test coverage
      1) Review the user specified functional requirements and Business Process Model to determine the critical business processes for validation (Section 1.3, 4.1.1, 4.1.2, 4.1.3)
      2) Choose the test techniques to be used
      3) Criteria for failing the test

4) Criteria for passing the test
5) Number of test cycles (iterations) if required

e. Identify requirements to be validated
1) DCII Specification (Appendix D) requirements deferred to the FVT
2) FFMR requirements deferred to the FVT
3) Critical business processes / requirements
4) Performance assessment criteria

f. Confirm list of test participants and facilitators (Section 3.x.7)
1) Test resources from design activity (facilitators)
2) Subject matter experts
3) FPO representatives (for test execution)
4) Observers
5) FPO / TPO / PM authorities
6) Performance monitor participants

g. Identify the roles and responsibilities of each organization, group, and test resource in support of the test. For each of the following, identify the actual person or persons that will perform the task. (Section 3.x.6)
1) Test setup and execution:
A. Defines test cases / scenarios for execution
B. Participates in the test execution
- Facilitates the test
- Supports the test event

C. Documents the results of the test execution
D. Validates fixes
E. Determines the test as complete
F. Monitors performance
2) Problem reporting / tracking:
A. Tracks Test Discrepancy Reports (TDRs) (web site)
B. Approves problems to be fixed for a release
C. Responsible for making the software fixes
D. Provides feedback to the TDRs
E. Tracks which fixes within a release
F. Performance issues
3) Test data
A. Provides test data
B. Controls test data
C. Validates test data
4) Test environment
A. Pre-loads data within the test environment
B. Controls test interfaces
C. Responsible for configuring the test environment
D. Provides DBA support
- Application level DBA support
- Server DBA support

E. Provides support for the test client machines (O/S, software, and hardware)
F. Provides support for the server (UNIX, Oracle, crons/schedulers, hardware)
G. Performs the system administration on the server
5) Configuration management
A. Provides control over the software configuration items
B. Provides the software releases to testing
C. Provides control over test configuration items
D. Documents the impacts (CIs) to fix a problem
6) Management responsibilities
A. Approves the test plans
B. Approves the test schedule

C. Resolves ambiguities within the functional requirements
D. Resolves discrepancies between the design and functional requirements
E. Responsible for requesting the test environment
F. Approves the software for enterprise level testing

h. Identify the test environment for Functional Validation Test (Section 3.0)
   1) Identify if locked test environment from application SIT will be reused or new schema is required for FVT
   2) Identify test environment requirements:
      A. Schema name / location to be used
      B. Notify DPET of timeline for environment
      C. Test data load procedures / instructions
      D. Backup and recovery procedures
      E. Test location / test sites
      F. Access requirements
         - User access and privileges
         - Role types
         - System administrator accesses
   2) Identify client requirements for test
      A. Building / room location
      B. Peripherals (printers, scanners, projectors, etc.)
      C. Number of test participants / configurations
      D. Client software requirements
      E. Telecommunications / LAN connectivity
   3) Validate release package has been turned over from SIT
      A. Release package complete
      B. Release procedures valid / approved (DPET / Release Management)

i. Identify the test products storage needs
   1) Identify the location to store the test products (test scripts, test plan, etc.)
   2) Identify the space requirements for the storage area
   3) Identify the standards for accessing and using the storage area

j. Identify the test discrepancy reporting procedures (Section 4.1.5)
   1) Who is authorized to create a Test Discrepancy Report (TDR)
   2) What accesses are required for the Configuration Management Information System (CMIS)
   3) Routing of a discrepancy/problem report
   4) Who has action to authorize the correction of a problem, and when
   5) Who provides the impacts of a problem upon the application and project schedule
   6) Who is responsible for documenting the action of correcting a problem, and the configuration items that would be impacted in the TDR within CMIS
   7) Identify the discrepancy/problem report standards
   8) Performance issues

k. Identify the reporting criteria
   1) Identify the report standards to be used
   2) Who is responsible for generating the report
   3) What is the frequency required for generating the report
   4) Routing of the report
   5) Identify who has responsibility for any action items within the report

l. Define test results procedures (Section 4.1.5)
   1) How test results are documented
   2) Who is notified when the test is complete
   3) Who insures all functional validation tests have been validated

m. Identify any externally mandated test standards
   1) Test plan format standard
   2) Test script format standard
   3) Test reporting format standard

n. Build structure to capture effort metrics for test resources
   1) Identify test tasks in LRS for each application under test
      A. Strategy
      B. Analysis
      C. Design
      D. Construct
      E. Execute

6. Verification:

a. Test management review
b. Project management review

7. Exit Criteria:

a. Initial Software Test Plan (format) (Sections **1.3, 1.4, 2.0, 3.0, 4.1.1, 4.1.2, 4.1.3, 4.1.5**)

8. Measures:

a. Data collected to determine the effort required to test the release using LRS:
   1) Estimated / actual hours to perform test strategy

Task Name: Analyze FVT Requirements **Task Number: T-FVT-002**
Component: Functional Validation Test **Category:** Software Engineering

1. **Task Name:** Analyze Functional Validation Test Requirements

2. Purpose:

The purpose of this task is to specify test cases for each function of the application to be tested. The specified test cases provide the information required to build a test script and test data that ensures the design meets the business processes. A result of the analysis task is an estimate of the time required to perform the test, a reasonable measurement of the test cases required for the test, and the means to which the test will be performed. The results of this task are documented in sections 4.2, 4.2.x, 5.0, and 6.0 of the Software Test Plan.

3. Roles:

Performing the test analysis is the responsibility of the application test manager for the functional validation test.

4. Entrance Criteria:

a. Software Test Plan (STP) – initial
b. Application Business Process Model (RD.010)
c. Application Business Function Model (RD.030)
d. Application Business Data Model (RD.040)
e. Module Functional documentation (MD.040)

5. Procedures:

Information gathered during the analysis task is to be documented in sections 4.2, 4.2.x, 5.0, and 6.0 of the Software Test Plan (STP).

a. Define test cases for functional validation test
   1) The following procedures are performed for each requirement type
      A. Analyze the available functional specifications and operational concepts
      B. Identify the critical business processes and operational events
      C. Define the technical operations in support of the business processes
      D. Review the requirements associated with the processes
      E. Identify compliance criteria to be validated (use traceability matrix)
      F. Identify the interactions between the business operations
      G. Identify the application user roles and access levels needed in support of the business operations within the application
      H. Identify the events that would require monitoring for performance measurements
   2) Define test cases that will not be validated, or cannot be tested
   3) Identify the documentation to be validated, i.e. online help, user manuals, operations manual/instructions
   4) Identify the operating environments that the application will support for the client
      A. Windows, UNIX, Web, etc.

b. Document the test cases within the repository (Sections 4.2 and 4.2.x)
   1) Define the execution sequence for the test cases
   2) Identify any dependencies that may need to be resolved in order to execute test cases
   3) Document dependencies

c. Identify test data required for validation
   1) Coordinate with FPO test participants on additional data required for functional validation
   2) Reuse test data from SIT

d. Notify test participants of test event and coordinate arrangements
   1) Agenda for test
   2) Provide travel arrangement information
   3) Directions to test site
   4) Point of contact information during test
   5) Test roles and responsibilities
   6) Security passes / vehicle permits

e. Evaluate how software will be released to testing. Coordinate with Release Management to determine the method that the software will be released.

f. Determine the work breakdown structure / tasking by resource. Identify the test cases to be assigned to each test resource and verified by test participants
   1) Identify tasks of test participants, test facilitators, observers of the test, and management
   2) Identify any additional responsibilities that will be assigned to each test resource
   3) Document tasks within a task form (format) for test resources
   4) Confirm tasking with each test resource
   5) Coordinate tasks with test participants, observers, and management

g. Estimate test duration for the component using the following factors (section 5.0):
   1) Number of new test cases to be developed for functional validation
   2) Time required for test participants to review test cases
   3) Time required for test participants to provide test data for validation
   4) Intial application training for test participants
   5) Test procedure briefs at the beginning of FVT
   6) Time required to travel, attend meetings, and coordinate activities
   7) Time delays waiting for problems to be fixed
   8) Problems with setup / configuration of the test environment
   9) Re-testing after fixes have been made
   10) Documentation of test results
   11) Effort to report and explain the testing status
   12) Follow-up on test cases which failed, including problem reporting, and defect tracing
   13) Time required for test support activities - version control, managing the test environment and databases, and troubleshooting automated test tools

h. Complete Software Test Plan (STP). Format, validate information, and forward to Test Director and Project Management for review and approval. Adjust plan as necessary.

i. Build test schedule in MS Project and provide to Test Director and Project Management

6. Verification:

a. Test management review

7. Exit Criteria:

a. Approved application Software Test Plan (STP) – final
b. Documented test cases

8. Measures:

a. Data collected to determine the effort required to test the release using LRS:
   1) Estimated / actual hours to perform test analysis

Task Name: Design Functional Validation Test **Task Number: T-FVT-003**
Component: Functional Validation Test **Category:** Software Engineering

1. Task Name: Design Functional Validation Test

2. Purpose:

Create a test design, and define the test cases that will be incorporated into the test scenarios to ensure test coverage of the functional requirements. Components of the test design consists of test cases, test scenarios, test scripts, and test data.

3. Roles:

Establishing a test design is the responsibility of each test resource assigned to the application under test. Overall coordination responsibility rests with the application test manager.

4. Entrance Criteria:

a. Software Test Plan (STP) – final
b. Documented test cases

5. Procedures:

a. Ensure software integration testing has been completed for application

b. Identify the different types of test transactions required for each test case
   1) Determine the events that must occur for processing
   2) Processes that are triggered by an event
   3) Steps that occur in each process
   4) Relationship between processes
   5) Supporting documentation for test transactions

c. Identify the sequence of execution
   1) User roles types
   2) Test transaction relating to roles
   3) Specific business functions relating to a role type
   4) Order of system processing

d. Review / update test data as needed
   1) Input data files
   2) Table data

e. Request test participants perform peer review
   1) Test products
      A. Test transactions identified for test
      B. Test cases identified for validation
      C. Sequence of execution
   2) Adjust as necessary to accommodate requirements by test participants

f. Design a method that user documentation will be validated such as online Help text, user manuals, and software training material during test. Document within test cases.

6. Verification:

a. Peer review of work products

7. Exit Criteria:

a. Test design (execution order, test cases / scenarios)
b. Input test data files (if applicable)
c. Table data from interfacing systems

8. Measures:

a. Data collected to determine the effort required to test the release using LRS:
   1) Estimated / actual hours to perform test design

Task Name: Construct Functional Validation Test **Task Number: T-FVT-004**
Component: Functional Validation Test **Category:** Software Engineering

1. **Task Name:** Construct Functional Validation Test

2. **Purpose:**

   Construct the test scenarios necessary to validate the software, and document the execution steps within the test scripts.

3. **Roles:**

   Constructing the test scenarios and documenting in test scripts is the responsibility of each test resource assigned to the application under test.

4. **Entrance Criteria:**

   a. Test cases and scenarios designed from previous task

5. **Procedures:**

   a. Build new manual test scripts (as required). (format) Each test script includes:
      1) Description of test case to be validated
      2) Data to be used in the validation
      3) Pre-setting or staging conditions that must happen prior to validation
      4) Steps required to run the test
      5) Method used to review results and verify what is expected
      6) Expected results
      7) Reference to requirement being validated

   b. Build new manual test scenarios (master test script)
      1) Document sequence of test scripts
      2) Document test scripts that can be executed concurrently

   c. Request test participants to conduct peer review of test scenarios / scripts
      1) Reused scripts / new scripts
      2) Update / modify based upon review feedback

   d. Automate test scripts, where possible

   e. Automate test scenarios ensuring that they simulate the same test scenarios as the manual tests

   f. Update requirements traceability matrix for new test scripts (detail). Document the following crosswalks:
      1) Functional/technical requirements to test scripts
      2) Test data to test scripts (interface files that data is contained within, entity instance charts, etc.)
      3) Functional/technical requirements to compliance requirements
      4) Test scripts to compliance requirements

   g. Build test environment, if required
      1) Setup hardware (client, server, web)
      2) Load client software (O/S, test tools, supporting software)
      3) Load server software (O/S, Oracle, etc.)
      4) Build database instance (DBA function)
      5) Populate database tables with initial test data
      6) Establish security accesses

7) Install application under test
8) Load input test data interfaces and validate load
9) Load actual input interfaces to be used for test
10) Install / setup database query tools
11) Backup / export baseline test data

h. Arrange / order supplies for test
   1) Paper products
   2) Test cases / scenarios / scripts (in work binders)
   3) Working products (i.e. pencils, paper, etc.)
   4) Supporting documentation (i.e. transaction vouchers, contract documents, etc.)
   5) Software documentation (user manuals, training manuals, operations manuals)
   6) Test procedures (i.e. TDR reporting procedures, Cognos query tool procedures, etc.)
   7) Requirements traceability matrix

i. Perform final test peer review. Verify environment, test scripts, and test data are ready for commencement of test. Review order of execution of test cases and simultaneous testing. Adjust for test as necessary within the project timeline.

6. Verification:

a. Peer review of work products

7. Exit Criteria:

a. Manual test scripts
b. Automated test scripts
c. Input interface test files
d. Input interface production type test files
e. Server environment
f. Client environment
g. Validation software (database query tool)
h. Requirements traceability matrix
i. Populated baseline test database

8. Measures:

a. Data collected to determine the effort required to test the release using LRS:
   1) Estimated / actual hours to perform test construct

Task Name: Execute Functional Validation Test **Task Number: T-FVT-005**
Component: Functional Validation Test **Category:** Software Engineering

1. Task Name: Execute Functional Validation Test

2. Purpose:

Execute functional validation test, compare the expected results to the actual results, formally report any noted discrepancies, validate that the software meets the requirement specifications, and identify any potential risk within the software.

3. Roles:

Each test resource assigned to the project under test is responsible for facilitating the test execution, and regression test execution. Test participants representing the Functional Project Office are responsible for test execution and verification of the software.

4. Entrance Criteria:

a. Test scripts
b. Input interface files
c. Access to test environment
d. Access to Configuration Management Information System (CMIS) (web site)

5 Procedures:

a. Obtain test sequence on master script (test scenario). Run test scripts in the order specified.

b. Record start date on master test script

c. Turn on operational monitoring tools for initial performance measurements. Coordinate this effort with the DBA.

d. Execute test scripts
   1) Record the start date / time for the test script
   2) Execute the steps within the test script
   3) Compare the actual results with the expected results and note any discrepancies
   4) During execution, capture proof of validation for compliance criteria such as GUI screen captures, database table updates, audit records
   5) Utilize validation software (i.e. database query report tool), to validate the data was updated and/or stored as expected, and note any discrepancies
   6) Record completion date / time for the test script
   7) Document the results of the test script
   8) Modify / correct test script, if necessary
   9) Execute all steps of the master test script

e. Generate Test Discrepancy Reports (web site) (TDRs) for any noted discrepancies from the test results
   1) Record TDRs in CMIS according to TDR procedures
   2) Document the sequence of events within the TDR
   3) Document the difference between the expected results and the actual test results
   4) Document the suspected problem
   5) Document any test data that was used within the test
   6) Document the test script identification number that the discrepancy was found
   7) Assign a priority level to the TDR
   8) Save and route TDR according to discrepancy reporting procedures

f. Record Test Discrepancy Reports (TDRs) using TDR log (format)

g. Record completion date on master test script once all test scripts have been executed

h. Build a "Quick Look" test summary report (detail)
   1) Document work-arounds, discrepancies found, fixes, and risks
   2) Route to the Test Director and Project Management
   3) Track any open items for Project Management for decision of fixes for next release

i. Re-test as necessary
   1) Reset test environment to same starting point as initial test cycle
   2) Restore backup and/or reload input interfaces as necessary
   3) Coordinate with Configuration Management to receive new release
   4) Install new release (client / server / Web)
   5) Validate version control
   6) Repeat test cycle as specified in the test plan
   7) Validate fixes to reported discrepancies
   8) Track TDRs to resolution

j. Document Software Test Report (STR) (format)
   1) Document work-arounds, discrepancies found, fixes, and risks
   2) Include evaluation of software application quality within report
   3) Route to the Test Director and Project Management
   4) Track that any remaining TDRs are converted to SCRs or cancelled

k. Clean up test products (detail)
   1) Correct / update test scripts as needed
   2) Correct / update test data as needed
   3) Automate regression test scripts / scenarios
   4) Review estimated verses actual level of effort

l. Place test products under Configuration Management (CM) control

m. Provide process improvements to process owner.

n. Validate release package is completed and ready for turnover to Release Management

6 Verification:

a. Project management review of TDRs
b. Acceptance of software application

7 Exit Criteria:

a. Documented risks
b. Resolved / completed test discrepancy reports
c. Software Test Report (STR)
d. Performance assessment
e. Test library (scripts and data) under CM
f. Release package

8 Measures:

a. Data collected to determine the effort required to test the release using LRS:
   1) Estimated / actual hours to perform test construct
b. Data collected to determine quality of the release:
   1) Using CMIS, the number of valid TDRs identified in the next level of testing for release (Enterprise Integration Test)