

Enterprise Performance Testing

Component: Enterprise Performance Testing

Component Description:

Performance testing provides views of the behavior of an application and captures measurements of the end-to-end infrastructure utilization during client load testing ([detail](#)) and server load testing ([detail](#)). Performance testing may include but not be limited to **measuring response time** ([detail](#)), **measuring throughput** ([detail](#)), **measuring message volume** ([detail](#)) and **identifying capacity requirements** ([detail](#)) during load and stress testing. A meaningful performance test requires that accurate workload information be available in order to simulate the actual number of users and the work they will be performing for each business function.

The views and measures produced by enterprise performance testing are best achieved through joint participation of key personnel for both DISA and DFAS. Key personnel for DISA would include the Unix Administrator, Oracle Administrator, physical database DBA and the capacity analysis team. Key personnel for DFAS would include the application DBA and the performance test team.

The premise is that these key personnel would effect the execution of software that would measure and monitor the behavior of the server and communications handlers during the execution of the performance test by the DFAS performance test team. The system reports and information gathered would then be evaluated and analyzed by applicable key personnel to assess how the system handled the workload generated by the performance test and to identify capacity requirements ([detail](#)) for the application.

Measuring Response Time and Throughput

The user is the ultimate critical test for determining the operational success of any Application Information System [AIS]. Applications that do not meet user's requirements and expectations for response time, throughput, ease of use, reliability, availability, and cost-effectiveness will fail the tests for user acceptance. The challenge for performance testing is to strike the delicate balance between the time and cost expended to collect performance data against achieving performance goals and economically operating an application system. Achieving performance goals means that the defined response time and throughput requirements for each business process is achieved and the customer can accomplish 'peak' workload within planned staffing.

The objective of performance testing is to validate that the performance parameters specified for the application have been met. The method used to accomplish the validation is to stress ([detail](#)) and measure the performance of an application by simulating various user loads and system usage. The captured measurements help isolate and identify potential client, network and server bottlenecks. Server performance testing measures application performance and scalability by sending hundreds or even thousands of client messages to evaluate how the server performed under load testing ([detail](#)). Client performance testing measures the application "end-to-end" response time and throughput under various load test scenarios created to emulate the application's processing to determine if it operates within the specified performance parameters. Throughput is defined to include information (report) delivery ([detail](#)) and data transmission ([detail](#)).

Automated performance testing will be conducted using Mercury Interactive's automated testing tools WinRunner and LoadRunner. These tools are used to simulate real users executing an application. They provide the capability to conduct scaleable system load tests for an application that simulate the system activity generated by one, hundreds or even thousands of users. Performance is measured end-to-end (from the client through the application server and to the database) providing transaction response time from the human-user point of view: time elapsed from the moment the OK button is pressed until results appear on the screen. Performance testing results are then evaluated against expected response times and throughput defined during the requirements definition phase of application development.

Enterprise Performance Testing

Measure Message Volume

The objective of measuring message load is to identify the size of the message traffic that is sent and received by the application and to observe the SQL statements transmitted between the client and server. Telecommunications traffic is defined as the number, size, frequency and peak periods for the messages [frames] that are transmitted back and forth between the client and server.

Some application functions will not require measuring transaction message load. Part of the effort to identify the performance test strategy will be to define the criteria for selecting the application functions that will require message load measuring. This criteria will then be used to select the functions that will be measured.

The measures gathered will provide some of the **capacity sizing data** required in the transaction volume section of the Infrastructure Services Request (ISR). Message volume information should also be used to evaluate the effectiveness and efficiency of how the client and server parts of the application were designed and coded as well as the database activity required to support the function. The insights and ideas gained from analysis of the message volume information and SQL activity may suggest or provide solutions and ideas for improving the operational efficiency of the unit in order to achieve the unit's performance requirements.

The performance test plan should identify the means and methods that will be used to capture message and SQL statement information. The alternatives available for choosing will necessarily depend on whether the application is 16 or 32 bit. Client-side sniffers (network analyzers) provide one means of capturing the desired information. Another means is to use SQL inspection software that captures all SQL and message data exchanged between the client and server. The database virtual user generator provided with LoadRunner provides one tool for capturing all message statements and SQL statements sent to Capacity Requirements or received by the client.

Sniffers can be used to conduct telecommunications measuring activities. Sniffers provide the capability to listen in and capture message traffic passing through a node of the client/server infrastructure. The resulting information must then be evaluated by telecommunications personnel who have the expertise required to interpret and classify the information captured by the sniffer. The results of this effort is expected to provide some of the information needed to "size" the telecommunications equipment for each client/server node where sniffer measurements were taken.

Identify Capacity Requirements

Another objective of performance testing is to identify the server Capacity Requirements required to operate an application within the system performance requirements specified by the customer. Server configuration is defined to include server manufacturer and model, number of processors, memory size, I/O controller, and other peripheral devices required to support the application.

The server configuration determination process requires the existence of a test environment in which to test drive an application to conduct automated performance and load testing. Server capacity measurement activities and telecommunications message measuring activities would be conducted when the load test scenarios are executed to stress the system.

The SAR [System Activity Reporter] is a significant source of UNIX system activity information used for server capacity sizing. The SAR provides a periodic system-wide view of system activity and provides information such as CPU utilization, certain I/O activity, queue counts of processes in memory and runnable, and unused memory pages and disk blocks. In some cases, special software tools like IOStats may be executed to capture total system I/O activity from all sources. All together, a variety of software tools are available, depending on the server brand, to capture, evaluate and present system activity and measurements. The collective result of these efforts is expected to provide the information needed to identify the "size" of the server required to economically operate an application system, meet the customers performance requirements and contains sufficient headroom [room to grow] to accommodate future growth in application usage and data storage.

Enterprise Performance Testing

Inputs:

- 1) Operational Requirements Document (ORD)
- 2) Test and Evaluation Master Plan (TEMP)
- 3) Operational Concept Description (OCD)
- 4) Software Requirements Specification
- 5) Interface Requirements Specification
- 6) Application Design Specification
- 7) Technical Architecture
- 8) User software documentation
- 9) System Architectural Design Specification
- 10) Test Plan, scripts, stubs and drivers from Functional and Unit Testing
- 11) Test data from Functional and Unit Testing
- 12) Business Process Model
- 13) Functional Hierarchy Diagram Model
- 14) Dataflow Diagram Model
- 15) CRUD activity for each Business Function and Module
- 16) Performance assessment criteria (requirements / parameters)
- 17) Approved initial performance test plan
- 18) Products from enterprise performance test analysis task
- 19) Metrics / System / SAR analysis software
- 20) Oracle utilities to produce report file "Report.TXT"
- 21) Sniffer on Test Client
- 22) Sniffer Analysis Skills
- 23) Approved Test Plan (S-PM-004)
- 24) Completed / Accepted Test Readiness Review Report (S-SE-001)
- 25) Test Discrepancy Report Standard (S-SE-011)
- 26) Testing Analysis Report Standard (S-SE-014)
- 27) Completed Test CIs (Scripts & Data)
- 28) Established DISA Testing Environment
- 29) Available Mercury software and licenses
- 30) Available Client systems

Outputs:

- 1) Performance Test Plan (PTP)
- 2) Test schedule
- 3) Automated test cases
- 4) Automated test scripts
- 5) Automated scenarios
- 6) Test data
- 7) Interface files
- 8) Test environment
- 9) Test discrepancy reports (TDR)
- 10) Performance Analysis Reports and Graphics
- 11) Performance Test Report (STR)

Task Ordering:

- 1) [Strategy](#)
- 2) [Analysis](#)
- 3) [Design](#)
- 4) [Construct](#)
- 5) [Execute](#)

Enterprise Performance Testing

Project Management Tasks:

- 1) Verify performance testing time identified in SDP
- 2) Ensure performance testing time is sufficient

Component Tasks:

Configuration Management Tasks:

- 1) Ensure performance testing environment is established
- 2) Verify Configuration Items are correct for performance testing

Component Metrics:

- 1) Data collected for each release:
 - a) Total number of TDRs and their resulting status (i.e. number fixed, canceled, deferred, and returned)
 - b) Total number of work around solutions identified
- 2) Data collected for each TDR
 - a) Release number
 - b) Priority
 - c) Status
 - d) Actual origin of problem
 - e) Date initiated
 - f) Date resolved
 - g) Configuration items impacted
- 3) Data collected for each test cycle:
 - a) Actual start date (note difference between planned start date and actual start date)
 - b) Actual completion date (note difference between planned completion date and actual completion date)
 - c) Test effort (manpower/resources)
 - d) Tasks performed (number of test cases executed, number not executed)
- 4) Data collected for each risk at the completion of the test:
 - a) Priority or level of risk
 - b) Date problem was identified
 - c) Status (if risk is high enough to require a fix to the software – SCR)
 - d) Release number
- 5) Factors affecting job dimensions for each test-related Configuration Item (CI):
 - a) Size of change
 - b) Number of new test scripts required
 - c) Number of test scripts to be modified
 - d) Test data impacted / built
 - e) Resource impact
 - f) Schedule impact
- 6) Performance measurements:
 - a) Application specific measurements
 - b) Limited access / interactions on test environment

Controls:

- 1) Project level technical standards
- 2) DCII standards
- 3) DII COE standards
- 4) JTA standards
- 5) IEEE/EIA J-STD-016
- 6) IEEE/EIA 12207

Enterprise Performance Testing

Task Name: Develop Test Strategy
Component: Enterprise Performance Test

Task Number: T-EPT-001
Category: Software Engineering

1. Task Name: Develop Strategy for Enterprise Performance Test

2. Purpose:

Identify the performance testing strategy to be used to validate the performance of an application. This task will identify the project and application information and documentation that will be needed to formulate a strategy for conducting the enterprise performance test. The primary purpose of this task is to identify the depth and extent to which these performance test activities will be performed.

3. Roles:

Development of the enterprise performance test strategy is the responsibility of the team responsible for enterprise performance testing.

4. Entrance Criteria:

- a. Operational Requirements Document (ORD)
- b. Test and Evaluation Master Plan (TEMP)
- c. Test Plan, scripts, stubs and drivers from Functional and Unit Testing
- d. Test data from Functional and Unit Testing
- e. Business Process Model
- f. Functional Hierarchy Diagram Model
- g. Dataflow Diagram Model
- h. CRUD activity for each function and module
- i. System Architectural Design Specification
- j. Performance assessment criteria (requirements / parameters)

5. Procedures:

The ORD and TEMP provides the information for the performance requirements for an application. During the strategy task, the plans, schedules, and resources outlined in the TEMP are reviewed and modified, as necessary. Plans specific to the performance test are documented in a Performance Test Plan (PTP). The performance test plan should be reviewed and certified by authorized individuals such as the Test Director and Program Manager, ensuring they are complete, meet standards, and adequately performance test the software. Information gathered during the strategy task is to be documented in sections 3.x.1, 3.x.2, 3.x.3, 3.x.4, 3.x.5, 3.x.6, 3.x.7, 4.1.1, 4.1.2, 4.1.3, and 4.1.5 of the Performance Test Plan (PTP).

- a. Identify available application information
 - 1) The operational environment
 - 2) The user community
 - 3) Available Project / Application documentation
 - 4) Criticality of Business Processes
 - 5) Business processes and associated criticality
 - 6) Workload information
 - 7) Performance criteria defined for the application
- b. Determine scope of performance test
 - 1) End-To-End Response Time
 - 2) Client
 - 3) Network
 - 4) Server
 - 5) GUI Response Time [End-To-End minus Network & Server]
 - 6) Batch

Enterprise Performance Testing

- 7) FTP
- c. Determine performance measurement tools availability and methods to be used
 - 1) Mercury Interactive's automated testing tools and licenses
 - 2) Sniffers
 - 3) SQL Inspection Tools
- d. Identify roles and responsibilities of all participating personnel (Section 3.x.6)
 - 1) Test setup and execution:
 - A. Who provides support for using unit, software integration and functional test cases and scripts
 - B. Who participates in the test execution
 - DISA DBA
 - DISA Oracle Administrator
 - DISA performance analysis staff
 - Application DBA
 - Performance test team members
 - C. Who executes the test cases
 - D. Who supports the test execution
 - E. Who documents the results of the test cases
- e. Problem reporting / tracking:
 - A. Who is responsible for making the software fixes
 - B. Who provides feedback to the TDRs
 - 2) Test Data:
 - A. Who provides test data for the test cases
 - B. Who pre-loads data for testing
 - C. Who controls test data
 - D. Who validates test data prior to test
 - 3) Test Environment:
 - A. Who provides hardware and software for the test environment
 - B. Who establishes test environment
 - C. Who controls test environment
 - D. Who provides server support for the test environment
 - E. Who provides client support for the test environment
 - F. Who provides LAN support for the test environment
 - G. Who provides DBA support
 - Application level DBA support
 - Server DBA support
 - 4) Configuration management
 - A. Who provides control over the software configuration items
 - B. Who provides the software releases to testing
 - C. Who provides control over test configuration items
 - 5) Management responsibilities
 - A. Who approves the test plans
 - B. Who approves the test schedule
 - C. Who resolves ambiguities within the performance requirements
 - D. Who is responsible for requesting the test environment
 - E. Who provides application training for test group
 - F. Who provides supporting software training for test resources
 - 6) Notify each participant of their assigned role and establish Points of Contact(POC) for test results
 - A. Who will receive the response time and throughput results
 - B. Who will receive the server capacity test results
 - C. Who will receive the transaction/message volume test results
 - 7) Identify the test discrepancy reporting procedures (section 4.1.5)
 - A. Who is authorized to create a Test Discrepancy Report (TDR)

Enterprise Performance Testing

- B. What accesses are required for the Configuration Management Information System (CMIS)
 - C. Routing of a discrepancy/problem report
 - D. Who has action to authorize the correction of a problem, and when
 - E. Who provides the impacts of a problem upon the application and project schedule
 - F. Who is responsible for documenting the action of correcting a problem, and the configuration items that would be impacted in the TDR within CMIS
 - G. Identify the discrepancy/problem report standards
- f. Identify the test environment (section 3.x.1, 3.x.2, 3.x.3, 3.x.4, 3.x.5)
 - 1) Review the architectural design specification
 - 2) Define the software and hardware needed to perform testing of this component
 - 3) Client environment
 - 4) Server environment
 - 5) Web environment
 - 6) Database instances necessary for testing
 - 7) Controls, communications, security accesses
 - 8) Space/size requirements
 - 9) Testing software tools
 - 10) Identify licensing requirements for software
 - 11) Identify location that software and hardware will be needed for testing
 - 12) Identify timeline that software and hardware will be needed for testing
 - 13) Identify administration responsibilities and maintenance responsibilities of the test environment
 - g. Identify test sites (Section 3.x)
 - 1) How many test environments
 - 2) Is the test environment different from development environment
 - 3) Where will the test environment(s) reside
 - 4) How long is the environment(s) needed
 - 5) What size are the environment(s)
 - 6) What security access is needed to get to the data or to perform the test
 - 7) What tools are needed on the test environment to perform my test
 - 8) Do I need special permission to use tool in the test environment
 - h. Identify pass/fail criteria
 - 1) Review ORD
 - 2) Review TEMP
 - 3) Baseline Performance Parameters
 - i. Determine performance test deliverables
 - 1) Identify test deliverables
 - 2) Define format and content
 - j. Document initial test plan
 - 1) Distribute plan for review and approval
 - 2) Update plan as required

6. Verification:

- a. Test management review of test plan and determination of feasibility to perform test within time and dollar constraints
- b. Project management review

7. Exit Criteria:

- a. Approved Initial performance test plan

8. Measures:

- a. Effort involved in defining enterprise performance testing strategy

Enterprise Performance Testing

Task Name: Analyze Test Requirements
Component: Enterprise Performance Test

Task Number: T-EPT-002
Category: Software Engineering

1. Task Name: Analyze Enterprise Performance Test Requirements

2. Purpose:

Analyze the information gathered during the “develop test strategy” task and identify the business functions and modules, workload and performance requirements that will be tested during the enterprise performance test.

3. Roles:

Analysis of the enterprise performance requirements is the responsibility of the team responsible for enterprise performance testing.

4. Entrance Criteria:

- a. Approved initial performance test plan
- b. Test Plan, scripts, stubs and drivers from Unit, Software Integration and Functional Testing
- c. Business Process Model
- d. Functional Hierarchy Diagram Model
- e. Dataflow Diagram Model
- f. CRUD activity for each function and module
- g. Technical Architecture
- h. Performance assessment criteria (requirements / parameters)

5. Procedures:

- a. Confirm availability of performance measurement tools and methods to be used
- b. Identify business process **scenarios** from unit, software integration and functional testing
 - 1) Review application information
 - A. Performance test operational environment
 - B. User community, surveys, etc.
 - C. Business process model
 - D. Criticality of business processes
 - E. Workload information
 - 2) Determine the performance criteria to be validated
 - A. Review performance criteria defined for the application
 - 3) Determine business processes to be included in the performance test
 - A. Review performance criteria by function
 - B. Determine business processes that are affected by defined performance criteria
 - C. Evaluate user types and roles
 - D. Evaluate functions by business process (critical processes vs. performance)
 - 4) Determine test scripts to be used in the performance test
 - A. Review test cases and scripts from unit, software integration and functional testing
 - B. Review test data from unit, software integration and functional testing
 - C. Review automated test scripts available for each function
 - D. Determine any changes required to existing scripts
 - E. Identify test scripts and data that must be created to run the performance test
 - Identify functions to be executed that do not have unit, software integration or functional test script
 - Identify project level personnel to assist in the script creation
 - Identify performance team personnel to assist in the script creation
 - Identify data required to execute function
 - Identify application steps required to successfully execute function with identified data

Enterprise Performance Testing

- Document the steps for executing the function
- 5) Identify load profiles
 - A. Users
 - B. Business processes and functions
 - C. Identifying frequency each function is used
 - By activity
 - By user role
 - By time period
 - D. Estimate workload
 - E. Estimate resources for each profile
- 6) Evaluate Load profiles
- 7) Identify acceptable load profile test results
- 8) Identify risks and contingencies associated with not testing a load profile
- 9) Prioritize load profiles
- 9) Estimate time required to create and implement each load profile

- c. Update test plan
 - 1) Distribute plan for review, approval and selection of load profiles to be automated and tested
 - 2) Update plan as required

6. Verification:

- a. Test management review of test plan and determination of feasibility to perform test within time and dollar constraints
- b. Project management review

7. Exit Criteria:

- a. Updated performance test plan

8. Measures:

- a. Number of performance test scenarios
- b. Number of performance test load profiles
- c. Effort spent in performance test analysis task

Enterprise Performance Testing

Task Name: Design Test

Task Number: T-EPT-003

Component: Enterprise Performance Test

Category: Software Engineering

1. Task Name: Design Enterprise Performance Test

2. Purpose:

Design the automated performance tests that will be used to conduct enterprise performance testing. Components of the performance test design consists of identifying the elementary, reusable and batch test scripts that are to be automated and used to execute the performance test scenarios.

3. Roles:

Design of the enterprise performance test is the responsibility of the team responsible for enterprise performance testing.

4. Entrance Criteria:

- a. Products from enterprise performance test analysis task
- b. Test Plan, scripts, stubs and drivers from Functional and Unit Testing
- c. Test data from Functional and Unit Testing

5. Procedures:

- a. Determine test script design
 - 1) Physical implementation of Detail scripts
 - A. Sharable scripts
 - B. Parameterization
 - C. Scripts to be automated using WinRunner (GUI virtual user)
 - D. Scripts to be automated using Database Virtual User generator (Database Virtual user)
 - E. Scripts to be manually executed (cannot be automated such as batch jobs, FTP processes)
- b. Determine Batch script design
 - 1) Script hierarchy
 - 2) Detail scripts assigned to batch scripts
- c. Determine strategy for approximating production size database
- d. Determine test scenario design
 - 1) Distribution of GUI and Database virtual user
 - 2) Script Instrumentation
 - A. Rendezvous
 - B. Start / Stop timing transactions
 - C. Think time
- e. Determine test cycle execution design
 - 1) Test data
 - A. Refresh
 - B. Backup / recover
 - C. Export / import
 - 2) Test result files
 - A. Graphical presentation files
 - B. Scenario detail logs
 - 3) Application releases
 - 4) Configuration changes

Enterprise Performance Testing

- f. Assign resources
 - 1) GUI MAP Development
 - 2) LoadRunner Scenarios
 - 3) WinRunner Virtual User Scripts
 - 4) Database Virtual User Scripts
 - 5) Supporting software
 - A. Drivers
 - B. Stubs
 - C. Audit Log reports
- g. Update test plan

6. Verification:

- a. Test management review of test plan and determination of feasibility to perform test within time and dollar constraints
- b. Project management review

7. Exit Criteria:

- a. Updated performance test plan

8. Measures:

- a. Time spent in enterprise performance test design task

Enterprise Performance Testing

Task Name: Construct Test

Task Number: T-EPT-004

Component: Enterprise Performance Test

Category: Software Engineering

1. Task Name: Construct Enterprise Performance Test

2. Purpose:

Construct the automated performance tests that will be used to conduct enterprise performance testing. Components of the performance test consists of automated elementary and batch test scripts and scenarios that are to be constructed and used to execute the performance tests.

3. Roles:

Construction of the enterprise performance test software is the responsibility of the team responsible for enterprise performance testing.

4. Entrance Criteria:

- a. Test plan, scripts, stubs and drivers from Functional and Unit Testing
- b. Test data from Functional and Unit Testing

5. Procedures:

- a. Establish automated test script development test environment (DFAS)
 - 1) Build database tables
 - 2) Load current application
 - 3) Populate database with test data
 - 4) Establish server-side Unix system to execute database virtual users
 - 5) Install Mercury software on servers
 - 6) Confirm availability of client systems required to execute performance test
 - 7) Confirm availability of Mercury software and licenses required to execute performance test
- b. Construct automated test scripts / scenarios
 - 1) Develop WinRunner virtual user scripts to perform each identified function
 - 2) Build test parameterization data
 - 3) Develop Database virtual user scripts using WinRunner scripts
 - 4) Build scenarios to be executed (contains appropriate virtual user scripts and host machines)
 - A. Build Rendezvous and transaction points in WinRunner and Database virtual user scripts
- c. Construct Supporting software
- d. Debug scripts and scenarios
 - 1) Execute scripts and scenarios
 - 2) Perform QA
 - 3) Fix problems
 - 4) Confirm outputs
- e. Peer review of scripts and scenarios
- f. Management review of scripts and scenarios
- g. Establish performance test environment (DISA)
 - 1) Build database tables
 - 2) Load current application
 - 3) Populate database with data that approximates production size database
 - 4) Establish server-side Unix system to execute database virtual users
 - 5) Install Mercury software on servers

Enterprise Performance Testing

- 6) Confirm availability of client systems required to execute performance test
- 7) Confirm availability of Mercury software and licenses required to execute performance test

6. Verification:

- a. Project management review

7. Exit Criteria:

- a. Automated performance test available for execution

8. Measures:

- a. Number of automated test scripts/scenarios developed
- b. Time spent in enterprise performance test construct task

Enterprise Performance Testing

Task Name: Execute Test

Component: Enterprise Performance Test

Task Number: T-EPT-005

Category: Software Engineering

1. Task Name: Execute Enterprise Performance Test

2. Purpose:

Execute the performance test scripts and scenarios using the automated testing tools WinRunner and LoadRunner.

3. Roles:

Execution of the enterprise performance test is the responsibility of the team responsible for enterprise performance testing. In addition, the following individuals may be involved to evaluate the results of the test.

Application DBA - Evaluate Oracle Utility Reports for Evaluate Oracle Utility Reports for Client /Server traffic; I/O, memory and CPU contention; and SQL Metrics

Physical Database DBA - Evaluate Utility Reports for I/O contention; evaluate file and table / index placement strategy.

Oracle Administrator - Evaluate Oracle Utility Reports for Client/Server traffic; I/O, memory and CPU contention; and SQL Metrics.

Sniffer Expert – Evaluate SQL Statement Count, Telcom Messages Frame Count Sent and Received and Frame Size.

4. Entrance Criteria:

- a. Metrics / System / SAR analysis software
- b. Oracle utilities to produce report file "Report.TXT"
- c. Sniffer on Test Client
- d. Sniffer Analysis Skills
- e. Approved Test Plan (S-PM-004)
- f. Completed / Accepted Test Readiness Review Report (S-SE-001)
- g. Test Discrepancy Report Standard (S-SE-011)
- h. Testing Analysis Report Standard (S-SE-014)
- i. Completed Test CIs (Scripts & Data)
- j. Established DISA Testing Environment
- k. Available Mercury software and licenses
- l. Available Client systems

5. Procedures:

- a. Establish testing event in CMIS
- b. Enter start date for Testing Event in CMIS
- c. Run Performance Test cycles according to script / plan
- d. Evaluate Performance Measurement results
 - 1) Create graphical reports
 - 2) Review Scenario detail logs and graphics
- e. Identify Performance Test Discrepancy Reports in CMIS

Enterprise Performance Testing

- f. Forward TDR to appropriate personnel
- g. Repeat test cycle as required
- h. Enter completion date for testing event in CMIS
- i. Prepare Test Analysis Report
 - 1) Distribute report for review and acceptance
- j. Place test products under Configuration Management Control

6. Verification:

- a. Management review of performance reports

7. Exit Criteria:

- a. Resolved performance issues
- b. Completed Testing Analysis Report (S-SE-014)
- c. Updated Test Library

8. Measures:

- a. Time spent in enterprise performance test execution task
- b. Number of test CIs completed