Introduction to Performance Testing

First Presented for:

PSQT/PSTT Conference

Washington, DC May, 2003

Scott Barber

Chief Technology Officer

PerfTestPlus, Inc.

Agenda

Why Performance Test?

What is Performance related testing?

Intro to Performance Engineering Methodology

Where to go for more info

Summary / Q&A

Why Performance Test?

Speed - Does the application respond quickly enough for the intended users?

Scalability – Will the application handle the expected user load and beyond? (AKA Capacity)

Stability – Is the application stable under expected and unexpected user loads? (AKA Robustness)

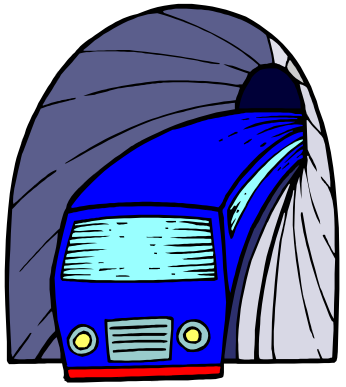
Confidence – Are you sure that users will have a positive experience on go-live day?

Speed

User Expectations
- Experience
- Psychology
- Usage

System Constraints
- Hardware
- Network
- Software

Costs
- Speed can be expensive!



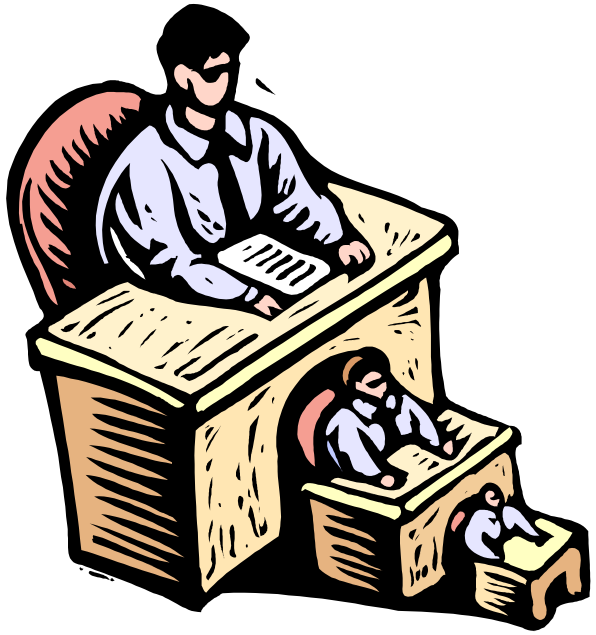
Scalability

How many users…

- before it gets “slow”?
- before it stops working?
- will it sustain?
- do I expect today?
- do I expect before the next upgrade?

How much data can it hold?

- Database capacity
- File Server capacity
- Back-up Server capacity
- Data growth rates



Stability

What happens if…

- there are more users than we expect?
- all the users do the same thing?
- a user gets disconnected?
- there is a Denial of Service Attack?
- the web server goes down?
- we get too many orders for the same thing?

Confidence

If you know what the performance is…

- you can assess risk.
- you can make informed decisions.
- you can plan for the future.
- you can sleep the night before go-live day.

The peace of mind that it will work on go-live day alone justifies the cost of performance testing.

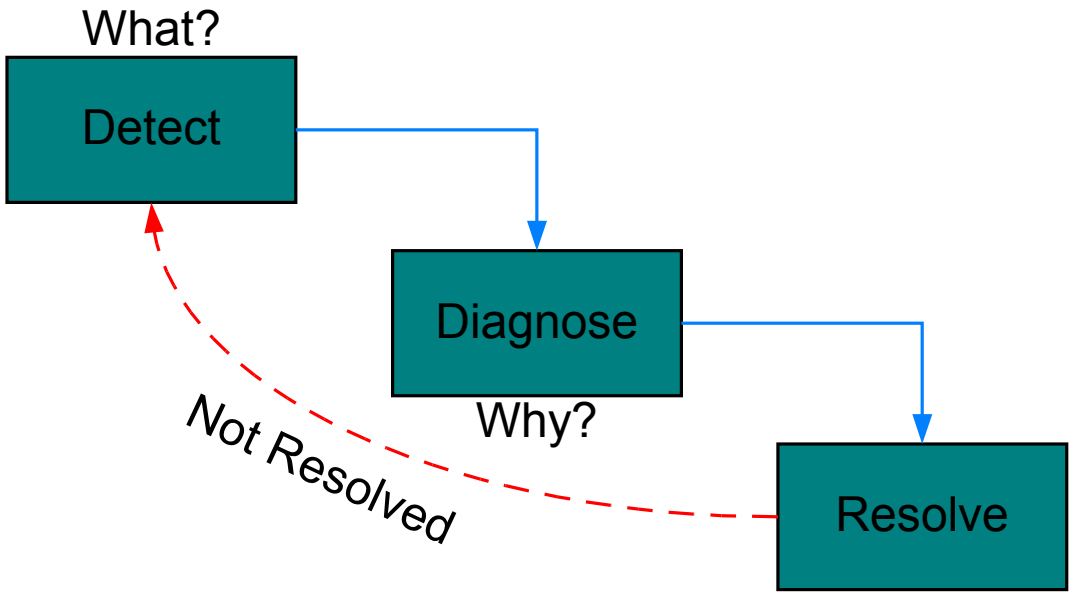
What is Performance Related Testing?

Performance Validation

Performance Testing

Performance Engineering

Compare & Contrast



Performance Validation

"Performance validation is the process by which software is tested with the intent of determining if the software meets pre-existing performance requirements. This process aims to evaluate compliance."

Primarily used for…

- determining SLA compliance.
- IV&V (Independent Validation and Verification).
- validating subsequent builds/releases.

Performance Testing

"Performance testing is the process by which software is tested to determine the current system performance. This process aims to gather information about current performance, but places no value judgments on the findings."

Primarily used for…

- determining capacity of existing systems.
- creating benchmarks for future systems.
- evaluating degradation with various loads and/or configurations.

Performance Engineering

"Performance engineering is the process by which software is tested and tuned with the intent of realizing the required performance. This process aims to optimize the most important application performance trait, user experience."

Primarily used for…

- new systems with pre-determined requirements.
- extending the capacity of old systems.
- "fixing" systems that are not meeting requirements/SLAs.

Compare and Contrast

Validation and Testing:

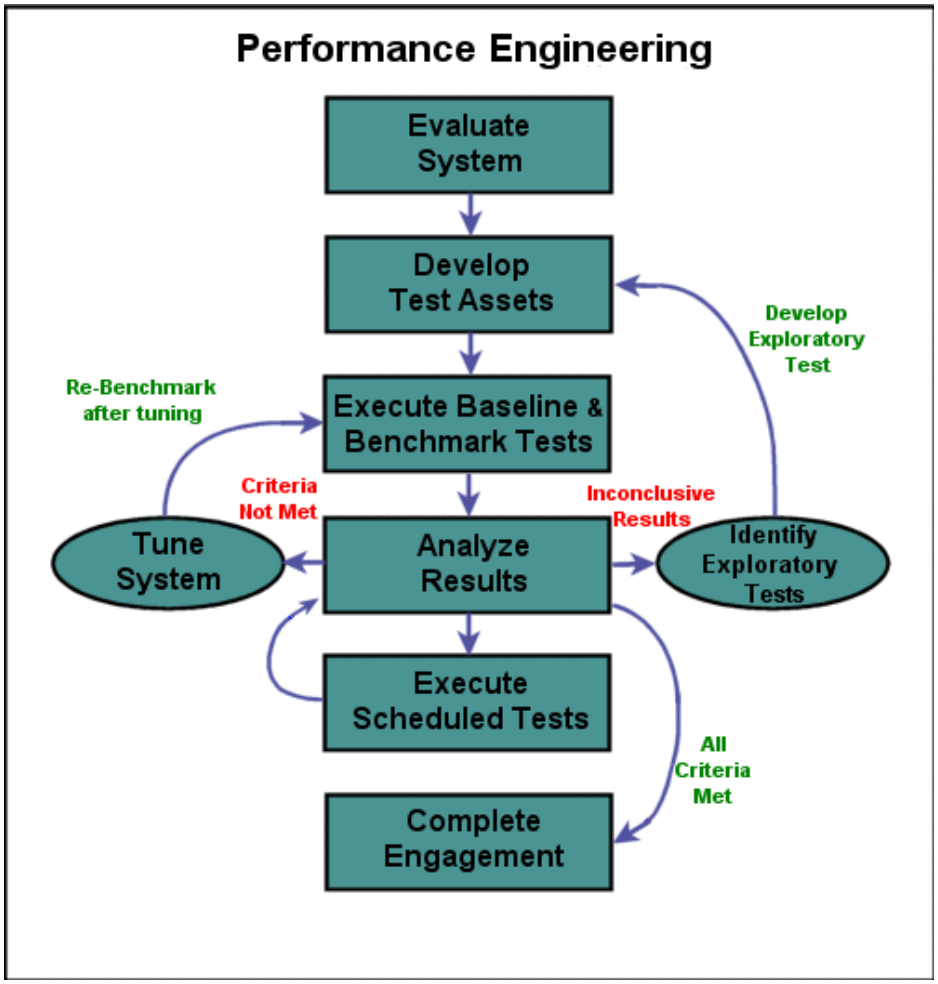
- Are a subset of Engineering.
- Are essentially the same except:
  - Validation usually focuses on a single scenario and tests against pre-determined standards.
  - Testing normally focuses on multiple scenarios with no pre-determined standards.
- Are generally not iterative.
- May be conducted separate from software development.
- Have clear end points.

Compare and Contrast

Engineering:

- Is iterative.
- Has clear goals, but ‘fuzzy’ end points.
- Includes the effort of tuning the application.
- Focuses on multiple scenarios with pre-determined standards.
- Heavily involves the development team.
- Occurs concurrently with software development.

Intro to PE Methodology



Evaluate System

Develop Test Assets

Baselines and Benchmarks

Analyze Results

Tune

Identify Exploratory Tests

Execute Scheduled Tests

Complete Engagement

Evaluate System

Determine performance requirements.

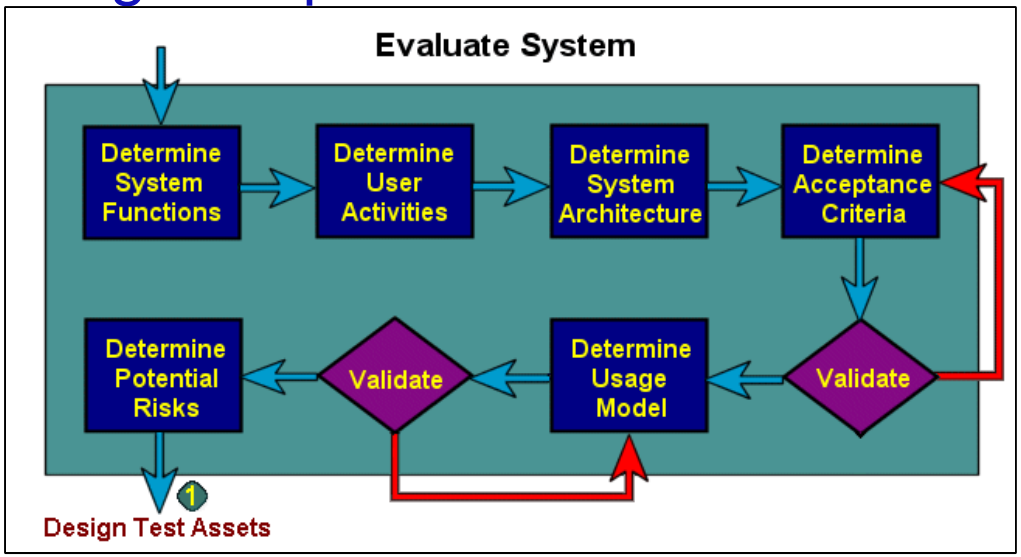
Identify expected and unexpected user activity.

Determine test and/or production architecture.

Identify non-user-initiated (batch) processes.

Identify potential user environments.

Define expected behavior during unexpected circumstances.



Develop Test Assets

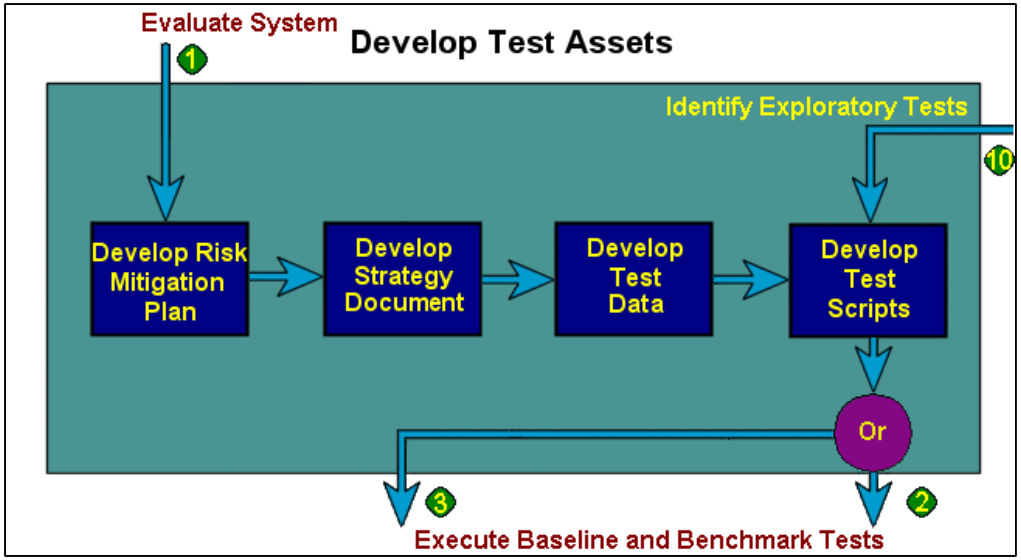
Create Strategy Document.

Develop Risk Mitigation Plan.

Develop Test Data.

Automated test scripts:

- Plan
- Create
- Validate

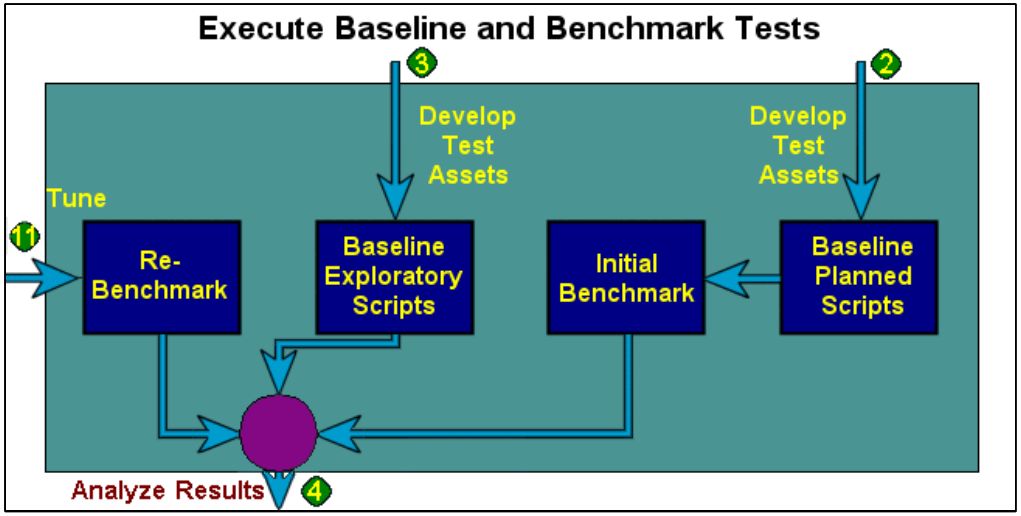


Baseline and Benchmarks

Most important for iterative testing.

Baseline (single user) for initial basis of comparison and 'best case'.

Benchmark (15-25% of expected user load) determines actual state at loads expected to meet requirements.



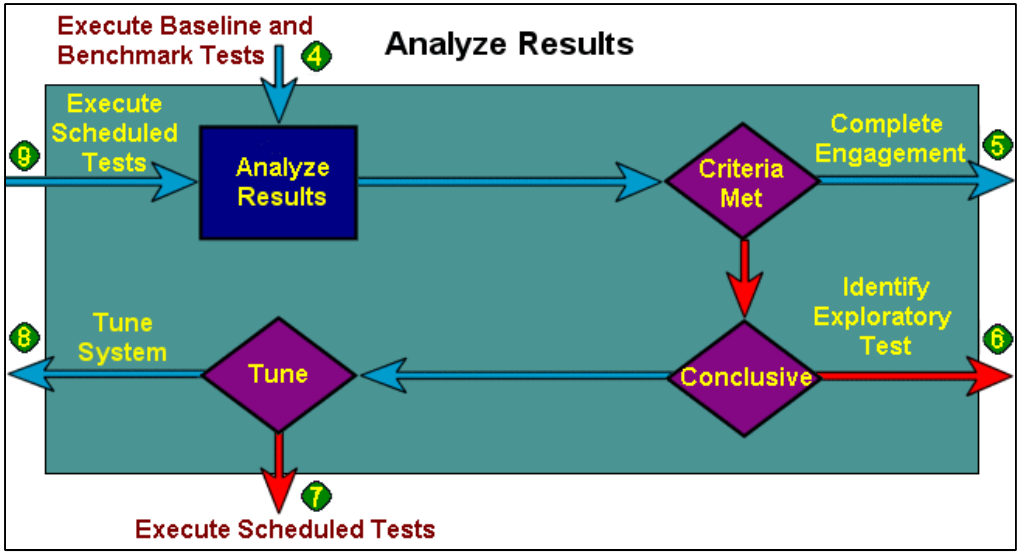
Analyze Results

Most important.

Most difficult.

Focuses on:

- Have the performance criteria been met?
- What are the bottlenecks?
- Who is responsible to fix those bottlenecks?
- Decisions.



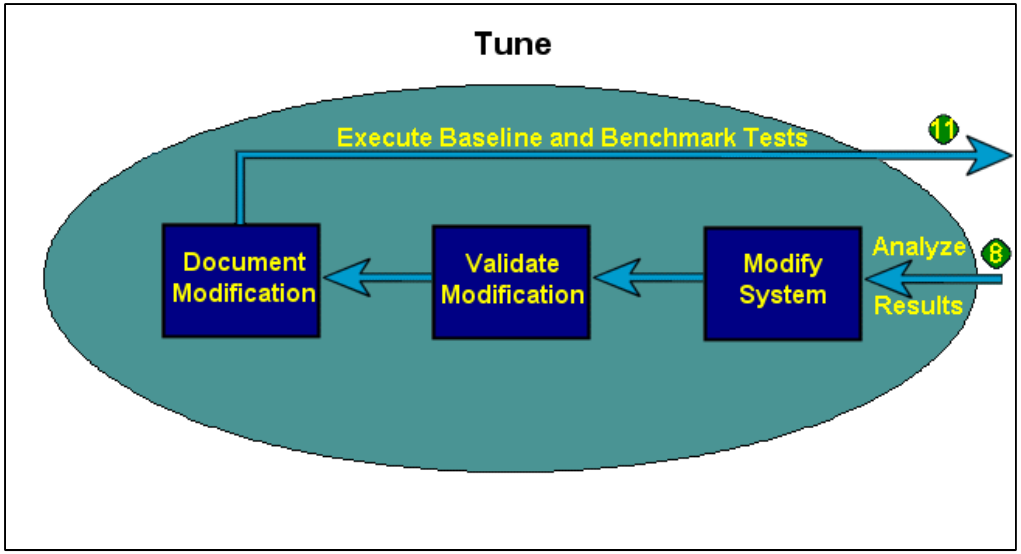
Tune

Engineering only.

Highly collaborative with development team.

Highly iterative.

Usually, performance engineer ‘supports’ and ‘validates’ while developers/admins ‘tune’.



Identify Exploratory Tests

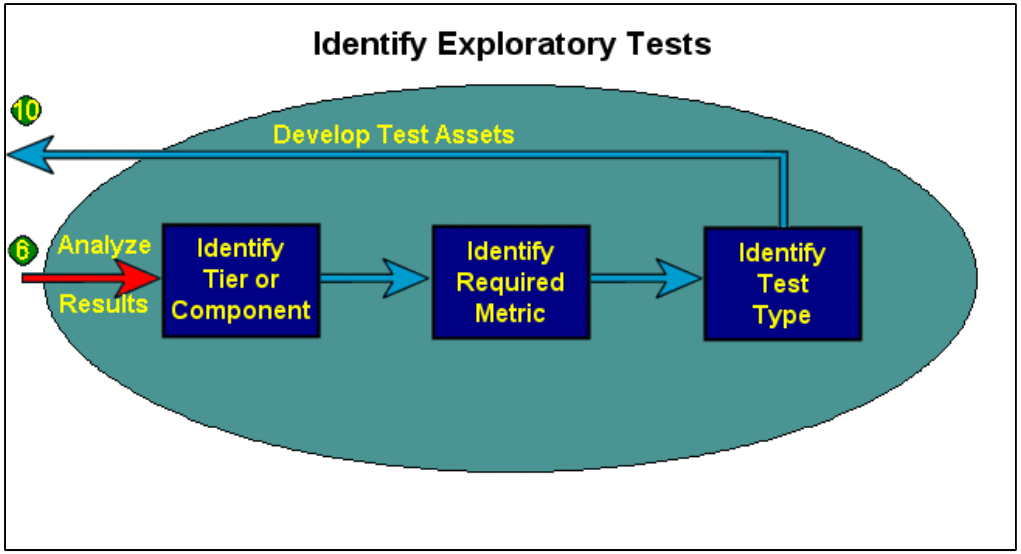
Engineering only.

Exploits known bottleneck.

Assists with analysis & tuning.

Significant collaboration with ‘tuners’.

Not robust tests – quick and dirty, not often reusable/relevant after tuning is complete.

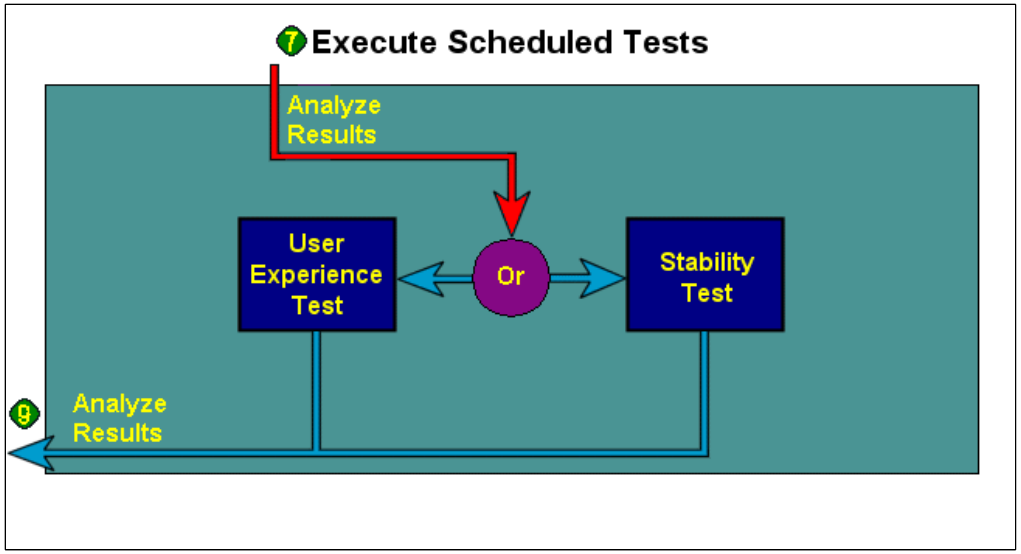


Execute Scheduled Tests

Only after Baseline and/or Benchmark tests.

These tests evaluate compliance with documented requirements.

Often are conducted on multiple hardware/configuration variations.



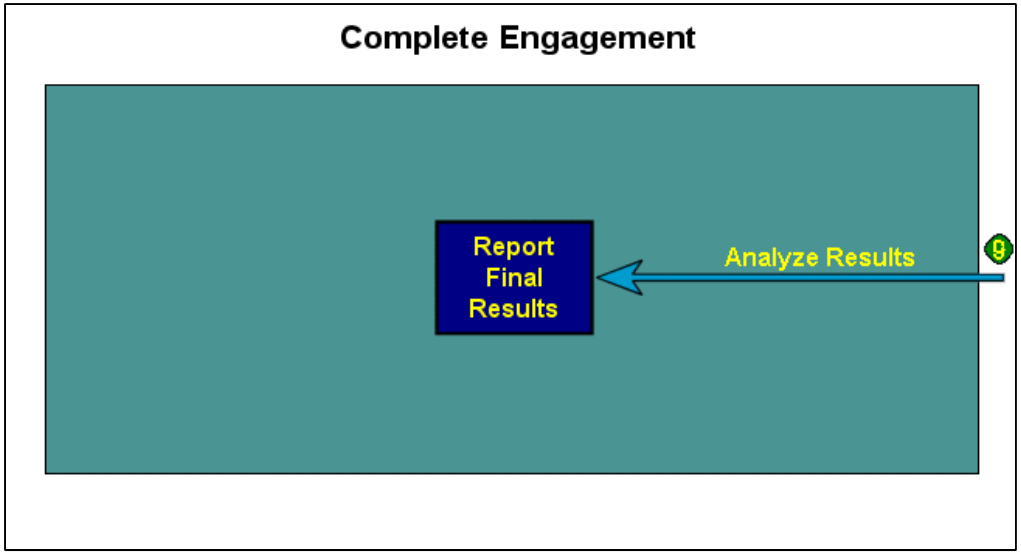
Complete Engagement

Document:

- Actual Results
- Tuning Summary
- Known bottlenecks not tuned
- Other supporting information
- Recommendation

Package Test Assets:

- Scripts
- Documents
- Test data



Where to go for more information

http://www.PerfTestPlus.com (My site)

http://www.QAForums.com (Huge QA Forum)

http://www.loadtester.com (Good articles and links)

http://www.segue.com/html/s_solutions/papers/s_wp_info.htm (Good articles and statistics)

http://www.keynote.com/resources/resource_library.html (Good articles and statistics)

Summary

We test performance to:

- Evaluate Risk.
- Determine system capabilities.
- Determine compliance.

Performance Engineering Methodology:

- Ensures goals are accomplished.
- Defines tasks.
- Identifies critical decision points.
- Shortens testing lifecycle.

Questions and Contact Information

Scott Barber

Chief Technology Officer

PerfTestPlus, Inc.

E-mail: *sbarber@perftestplus.com*

Web Site: *www.PerfTestPlus.com*