Performance Testing for Managers

First Presented:

Software Test & Performance Conference

November 2005, New York, NY

Most Recently Revised for:

Software Test & Performance Conference

October 2007, Boston, MA

Based on:

“High Performance Testing”

Better Software Magazine, *May/June 2005*

Scott Barber, Chief Technologist

PerfTestPlus, Inc.

Agenda

Overview

People

Tools

Process

Results

Summary

Overview

As an activity, performance testing is widely misunderstood, particularly by executives and managers.

This misunderstanding can lead to a variety of difficulties -- including outright project failure.

Overview

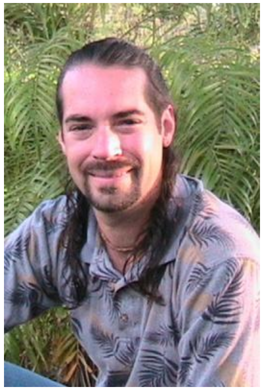
Managers and executives do not need to understand the technical details of performance testing to make good decisions or effectively manage performance testing projects.

They do need to understand what performance testing is, what it is not and what value it adds.

Overview

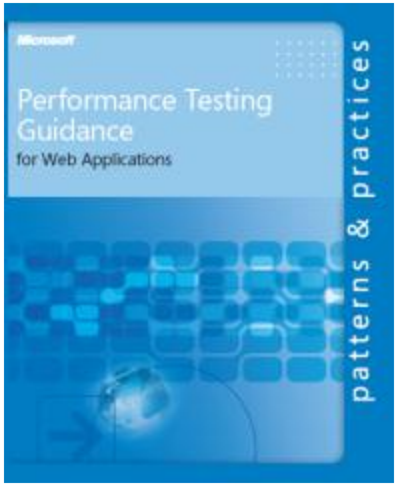
Learning, understanding, and applying the nuggets of knowledge in this presentation to your performance testing projects will improve your chances for success.

Performance Testing for Managers



Scott Barber, (A.K.A. *“The Perf Guy”)*
sbarber@perftestplus.com

Chief Technologist, PerfTestPlus, Inc.
www.perftestplus.com



Microsoft
Performance Testing Guidance
for Web Applications
patterns & practices

Vice President & Executive Director,
Association for Software Testing
www.associationforsoftwaretesting.org

Contributing Author & Expert Advisor
Performance Testing Guidance for Web Applications
www.codeplex.com/PerfTestingGuide

People

“There is no such thing as a ‘junior performance tester’…

but there are people who are new to performance testing.“

--Scott Barber

A (Good) Performance Tester Is:

A senior member of the team.

A “mid-level” everything.

Consultative, collaborative, competent, self-managing, trustworthy, and dependable.

Multi-disciplinary.

A member of a network of peers to collaborate with about performance testing challenges.

A (Good) Performance Tester Will:

Help you verbalize and quantify both business and technical goals & objectives.

Collect and present goal achievement data intuitively.

Speak intelligently about risks & implications related to:

- The Business
- The Application/System
- The Project
- Users of the application/system

Advise managers about performance and performance testing to help them make good decisions.

Interviewing Performance Testers:

Expect candidates to address questions typically asked of:

- Developers
- System administrators
- DBAs
- Technical managers

Expect candidates to relate personal experiences about:

- Solving complex technical problems in specific technical detail
- Addressing business risks through their testing
- Leading or coordinating cross-functional teams

Expect candidates to ask a ***lot*** of questions.

Managers Set Expectations:

You are not interested in “fools with tools.”

The performance tester is your personal advisor.

You will review strategies, whiteboards, results & deliverables – and ask questions about them.

You want to be educated along the way.

Encourage performance testers to develop relationships:

- With performance testers
- And other specialists
- Internally and externally
- And to leverage those relationships to solve challenges

Tools

"Enterprise grade load generation tools are designed to look easy in sales demos.

Don't be fooled."

--Scott Barber

Tool Related Tips for Managers:

The tool is a load-generator & data collector, not a performance tester.

The tool was not created for your system, making it work for you will be hard.

Commit to the performance tester's tool(s) of choice.

Avoid "tool-driven test design."

Do not accept "canned" reports as analysis.

Allow the use of supplemental open-source or free tools.

Cough up $35 when your performance tester asks to license share-ware when the trial expires.

Trust your performance testers, but not the tools.

A (Good) Performance Tester Is:

A master with their tool(s) of choice.

Able to extend the tool to implement the test design correctly (but it will take time).

Able to pick up a new tool quickly.

Not defensive when the tool is challenged.

Equipped with a virtual tool-box filled with tools that you have likely never heard of.

Frequently overheard complaining about the inadequacies of the tool… when their not bragging about the cool new features.

Process

"Only performance testing at the conclusion of system or functional testing

is like

ordering a diagnostic blood test after the patient is dead."

--Scott Barber

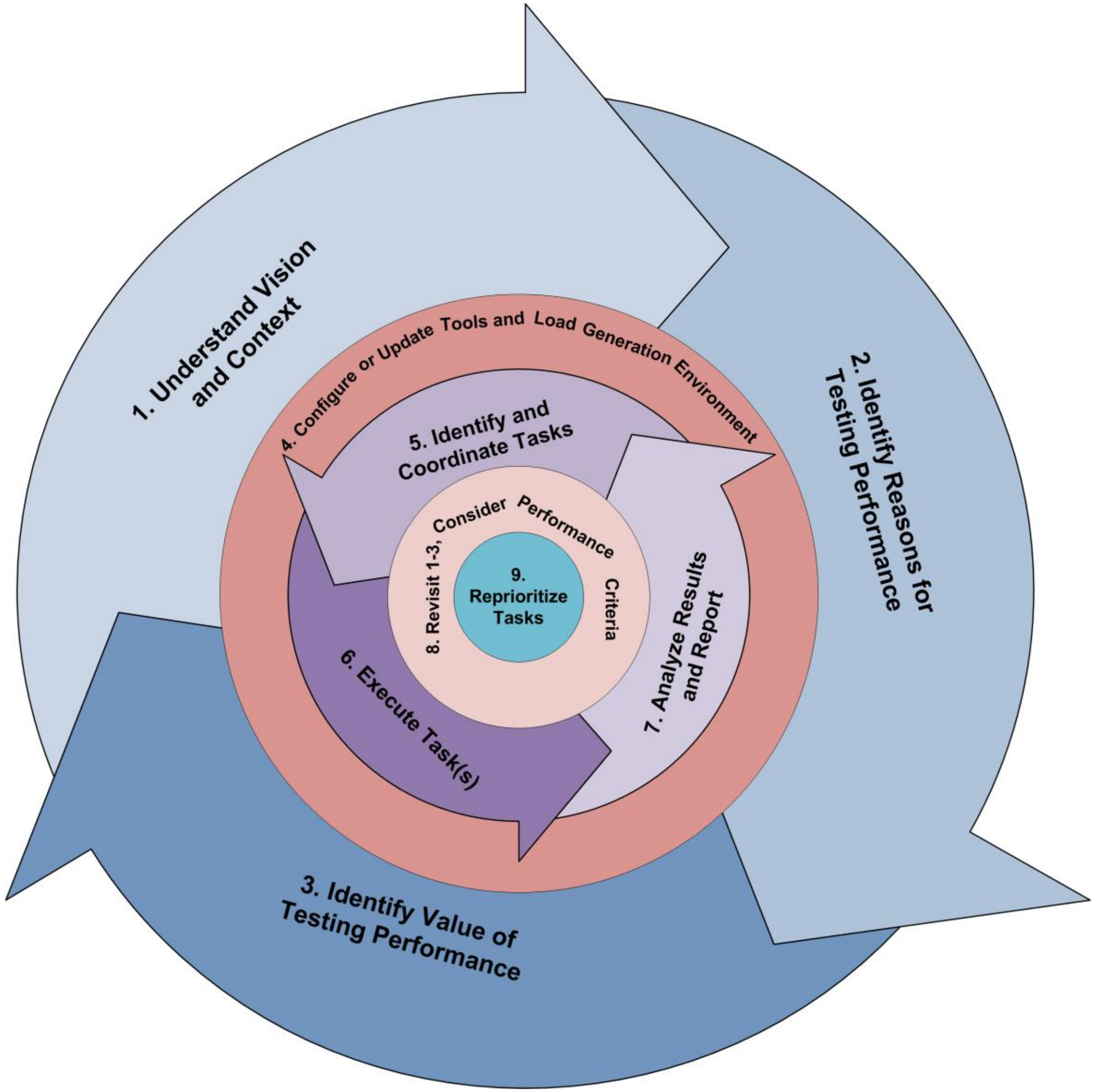
Performance Testing Principles:

Context	Project context is central to successful performance testing.
Criteria	Business, project, system, & user success criteria.
Design	Identify system usage, and key metrics; plan and design tests.
Install	Install and prepare environment, tools, & resource monitors.
Script	Script the performance tests as designed.
Execute	Run and monitor tests. Validate tests, test data, and results.
Analyze	Analyze the data individually and as a cross-functional team.
Report	Consolidate and share results, customized by audience.
Iterate	"Lather, rinse, repeat" as necessary.

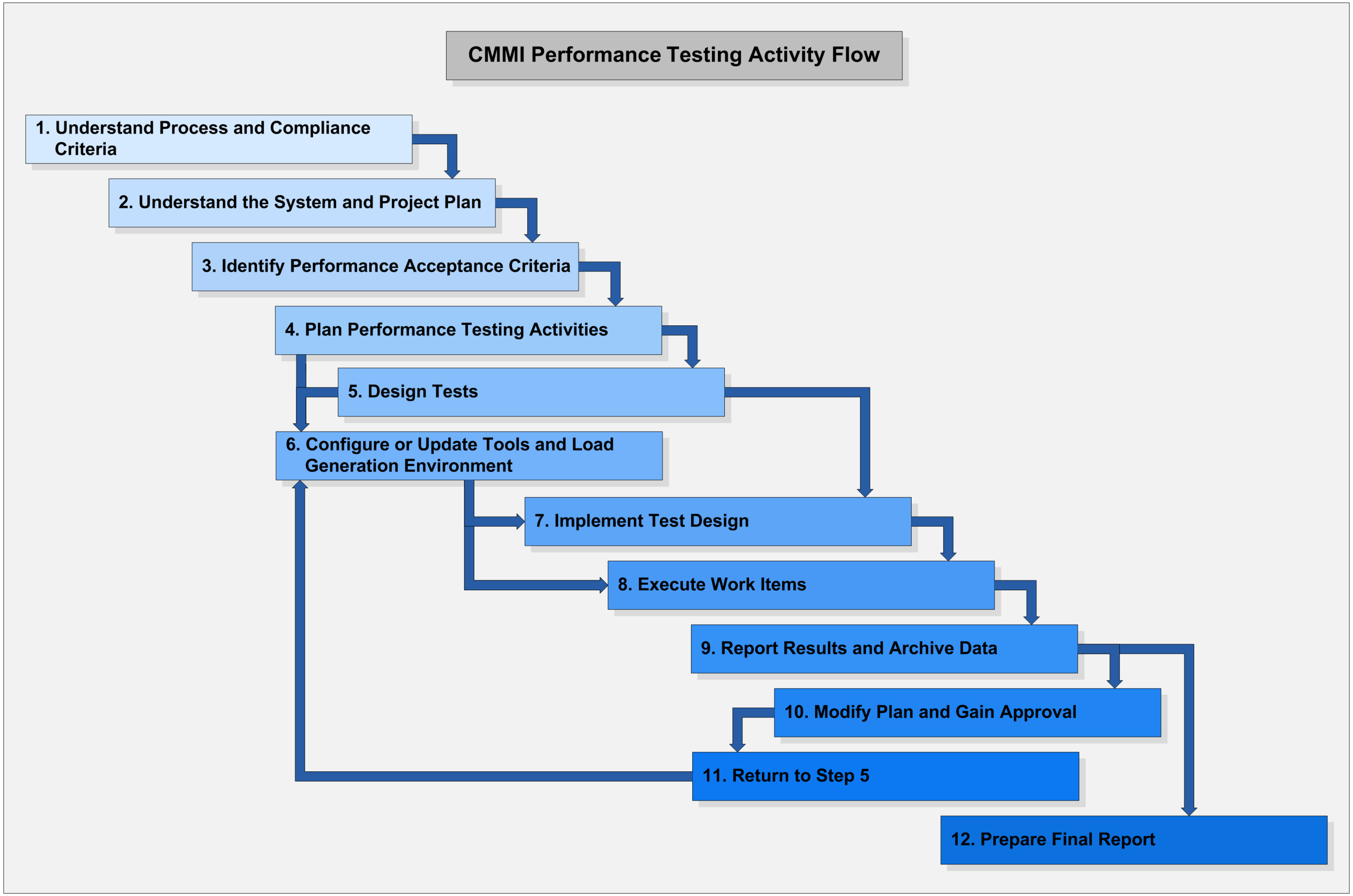
Agile Performance Testing:

Agile Performance Testing Activities

1. Understand Project Vision and Context
2. Identify Reasons for Testing Performance
3. Identify Value of Testing Performance
4. Configure or Update Tools and Load Generation Environment
5. Identify and Coordinate Tasks
6. Execute Task(s)
7. Analyze Results and Report
8. Revisit 1-3, Consider Performance Criteria
9. Reprioritize Tasks


1. Understand Vision and Context
2. Identify Reasons for Testing Performance
3. Identify Value of Testing Performance
4. Configure or Update Tools and Load Generation Environment
5. Identify and Coordinate Tasks
6. Execute Task(s)
7. Analyze Results and Report
8. Revisit 1-3, Consider Performance Criteria
9. Reprioritize Tasks

CMMI Performance Testing:


CMMI Performance Testing Activity Flow
1. Understand Process and Compliance Criteria
2. Understand the System and Project Plan
3. Identify Performance Acceptance Criteria
4. Plan Performance Testing Activities
5. Design Tests
6. Configure or Update Tools and Load Generation Environment
7. Implement Test Design
8. Execute Work Items
9. Report Results and Archive Data
10. Modify Plan and Gain Approval
11. Return to Step 5
12. Prepare Final Report

A (Good) Performance Tester Will:

Pair with developers & architects on early performance tasks.

Work within the constraints of the schedule.

Speed up the process of finding root causes and fixing performance issues detected late in the project life-cycle.

Know “delivery” is a business decision based on risks, costs, and benefits.

Help management weigh those risks, rather than trying to be “release traffic cops.”

Know that releasing the software is no reason to stop performance testing.

Process Related Tips for Managers :

Performance test from kick-off through roll-out.

Encourage the development team to use the performance tester's skills and resources as a development tool.

Performance testing is a primary role – not an additional duty.

Do not assign the performance tester to the test manager.

Plan to keep testing performance after the first release.

Prepare for performance monitoring in production.

Do not trust performance results until validated in production.

Plan to push a performance maintenance releases prior to the first expected peak load.

Realize that performance has a tendency to change over time.

Results

"Linear extrapolation of performance test results is, at best, black magic.

Don't do it (unless your name is Connie Smith, PhD. or Daniel Menasce, PhD.)"

--Scott Barber

A (Good) Performance Testers Will:

Know that linear extrapolation is, at best, black magic.

Not predict production performance based on measurements of a system that isn't identical to production.

Like when you ask to see the data.

Present data, results, and findings clearly and intuitively.

Find "prove it" to be a fair, reasonable, and anticipated request.

Results Related Tips for Managers:

Demand to review the assumptions, calculations, and data that contribute to any predictions.

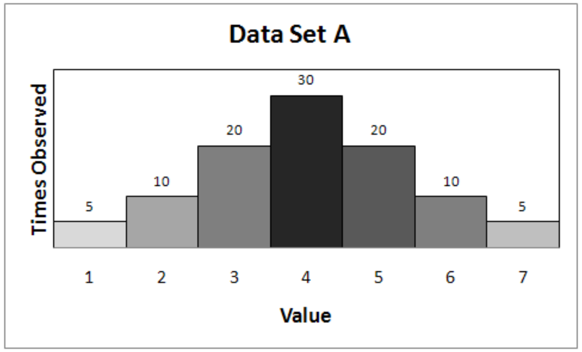
Ensure that your project plan involves at least one load test on production hardware.

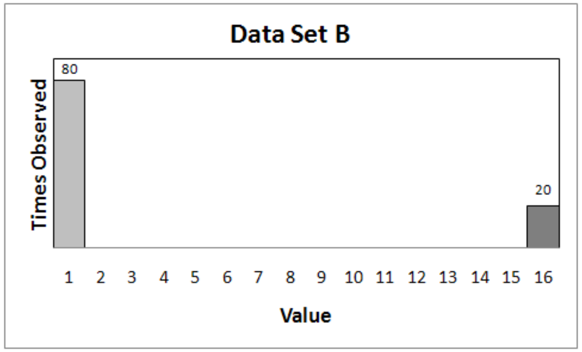
Do not accept defensiveness and/or arrogance in response to a request to justify or prove predictions.

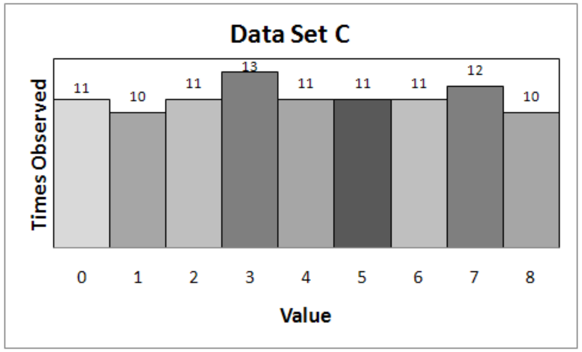
Know (or learn) some statistics.

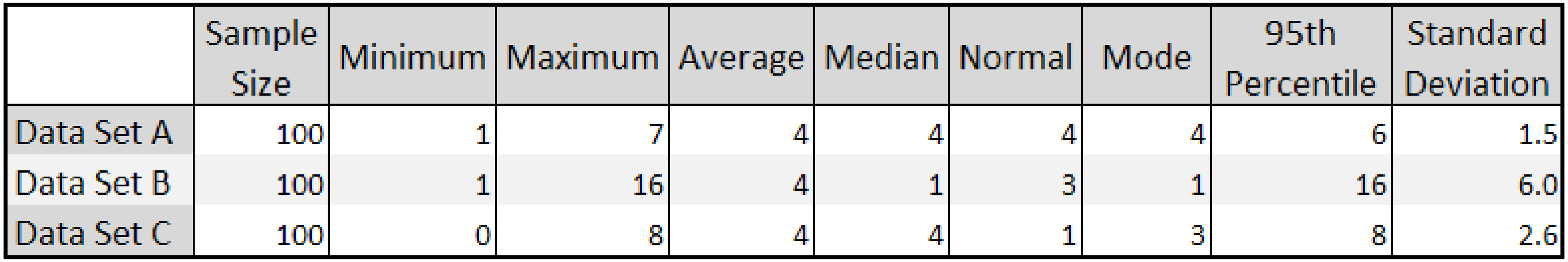
With every presented result ask “How many users will this make unhappy?”

Statistics Summary for Managers:









	Sample Size	Minimum	Maximum	Average	Median	Normal	Mode	95th Percentile	Standard Deviation
Data Set A	100	1	7	4	4	4	4	6	1.5
Data Set B	100	1	16	4	1	3	1	16	6.0
Data Set C	100	0	8	4	4	1	3	8	2.6

All three have an average of 4.

Which has the “best” performance”?

How do you know?

Summary

Experienced performance testers relate to executives and managers, not only “geeks.”

Performance testing can start adding value with the “bar napkin.”

Deliver based on risks. Stop testing when goals are met. Don’t confuse the two.

Expect skills and experience from a performance tester. There is no such thing as a “junior” performance tester.

The best tool for the job is the one that your performance tester recommends.

Don’t accept extrapolated results – they are wrong.

Credits

Adapted from "High-Performance Testing," *Better Software Magazine*, May/June 2005 issue located at http://www.perftestplus.com/pubs.htm.

Enhanced with material from *Performance Testing Guidance for Web Applications*, a Microsoft patterns & practices publication by J.D. Meier, Scott Barber, Carlos Farre, Prashant Bansode, and Dennis Rea available at http://www.codeplex.com/perftestingguide

Questions

Contact Info

Scott Barber

Chief Technologist

PerfTestPlus, Inc.

E-mail:

sbarber@perftestplus.com

Web Site:

www.PerfTestPlus.com