



Collaborative Performance Testing & Tuning Teams

First Presented for:

Software Test & Performance Conference
Dec 7-9, 2004 Baltimore, MD

Scott Barber
Chief Technology Officer
PerfTestPlus, Inc.



Agenda

Introduction

Why Tune Collaboratively?

The Testing and Tuning Team

How to Build the Team

Want More Information?

Summary/Questions



Introduction

This presentation is adapted from *Beyond Performance Testing*: Part 11 located at <http://www-106.ibm.com/developerworks/rational/library/> (RDW) and <http://www.perftestplus.com/>.



Introduction

Traditionally, performance testing **stops after** the performance tester(s) identifies the **bottlenecks**, then the developers begin a **separate activity of performance tuning** – each without the involvement of the other.

If you're new to performance testing, you may be surprised to hear that there is often a lot of **resistance to** the idea of **collaborative tuning**. A common argument is that there should be a **clear division** of tasks between testing and tuning.

Experience shows that this **separation of tasks is a mistake**. A better approach is to create a **collaborative testing and tuning team**.



Why Tune Collaboratively?

Bringing Tester and Developer Mindsets Together

- ▶ **Developers and Testers Think Differently**

- ▶ Testers tend to look for ways to make the application perform incorrectly in every possible situation.
- ▶ Developers tend to try to make the application perform correctly in the situations in which they envision the application being used.

- ▶ **This Thought Difference...**

- ▶ can be combative.
- ▶ can be collaborative and mutually beneficial.
- ▶ come together by developers finding the tester to be a valuable resource.



Why Tune Collaboratively?

Seeing the Big-Picture View Alongside the Detail View

- ▶ **Developers during Tuning**
 - ▶ Detail oriented
 - ▶ Cause focused
- ▶ **Performance Testers during Tuning**
 - ▶ Big picture oriented
 - ▶ Effect focused
- ▶ **Pros and Cons**
 - ▶ Adversarial when seen separately
 - ▶ Checks and Balances when viewed together
 - ▶ Both views needed for complete system view



Why Tune Collaboratively?

Streamline the Tuning Cycle

▶ Traditional

- ▶ Roughly 1 week per change/theory
- ▶ Full of “red-tape”

▶ Collaborative (Streamlined)

- ▶ Roughly 2 days per change/theory
- ▶ “Red-tape” minimized
- ▶ Based on direct interaction & trust between tester and developer
- ▶ Results over process (Agile)



The Testing and Tuning Team

Project Manager

- ▶ Performance-Related Focus
 - ▶ Responsible for end to end system
 - ▶ Ultimately responsible for both testing and tuning
- ▶ Contribution to Collaborative Tuning Team
 - ▶ Sets the tone for collaboration
 - ▶ Dictates the priority of performance issues
 - ▶ Manages resources available



The Testing and Tuning Team

Lead Developer/Architect

- ▶ **Performance-Related Focus**
 - ▶ Responsible for system design providing appropriate performance
 - ▶ Also sets priorities on performance issues.
- ▶ **Contribution to Collaborative Tuning Team**
 - ▶ Task assignments
 - ▶ Encourages direct communication between developers and testers
 - ▶ Needs to know when to call in outside experts



The Testing and Tuning Team

Developers/Administrators/DBAs

- ▶ **Performance-Related Focus**
 - ▶ Component level performance
 - ▶ Performance of integration points
 - ▶ Do actual tuning
- ▶ **Contribution to Collaborative Tuning Team**
 - ▶ Open to input and two way communication with testers
 - ▶ Control feedback loop with testers
 - ▶ Need to use the performance testers resources to tune effectively



The Testing and Tuning Team

Test Manager

- ▶ **Performance-Related Focus**
 - ▶ Champion of good performance
 - ▶ Develop procedures to enable collaborative teamwork
- ▶ **Contribution to Collaborative Tuning Team**
 - ▶ Get performance tester involved with developers
 - ▶ Start performance testing early and often
 - ▶ Allow performance tester to bypass red-tape in favor of collaboration and experimentation



The Testing and Tuning Team

Performance Tester

- ▶ **Performance-Related Focus**
 - ▶ Develop the overall performance test strategy
 - ▶ Collect and quantify the performance requirements
 - ▶ Determine and document the user community model(s)
 - ▶ Create scripts representing the user community model(s)
 - ▶ Execute the scripts and analyzing the results
 - ▶ Work with the developers as part of the collaborative testing and tuning team
- ▶ **Contribution to Collaborative Tuning Team**
 - ▶ Collection, analysis and reporting of results
 - ▶ Communicate in language of developers
 - ▶ Data collector for developer experiments



The Testing and Tuning Team

Outside Experts

- ▶ Performance-Related Focus
 - ▶ As needed basis
 - ▶ Almost always expert tuners
- ▶ Contribution to Collaborative Tuning Team
 - ▶ Tune components without in house expertise
 - ▶ Familiar with integration issues
 - ▶ Generally promote collaboration between testers and tuners



How to Build the Team

Know the Technologies

- ▶ Speak in language developers understand
- ▶ Be able to intelligently speculated both causes and effects
- ▶ Understand simple vs. challenging issues
- ▶ Be able to offer viable options to difficult experiments



How to Build the Team

Attend Meetings

- ▶ Best place to gain technical understanding
- ▶ Where relationships are forged
- ▶ Respect gained for showing interest and involvement
- ▶ Lines of communication are opened



How to Build the Team

Educate Your Team

- ▶ Most team members don't understand critical performance issues and simulations
- ▶ Much popular information is wrong
- ▶ Team may not be used to collaborative methods



How to Build the Team

Ask Questions

- ▶ Intelligent questions lead to both understanding and respect
- ▶ Help you determine who is most likely to assist with technical issues
- ▶ Seal your position as part of the team



How to Build the Team

Offer Assistance

- ▶ Developers need data, help them get it
- ▶ Help developers verify experiments
- ▶ Make your tools available
- ▶ Performance tools can be extremely valuable for...
 - ▶ Data generation
 - ▶ Database population
 - ▶ Comparing configurations
 - ▶ Reproducing scenarios



How to Build the Team

Be Available and Approachable

- ▶ Don't be “out of sight/out of mind”
- ▶ Accept the challenge of supporting results
- ▶ Don't hide behind initial test strategy
- ▶ Demonstrate a desire to assist.



How to Build the Team

Tear Down the QA-versus-Development Barrier

- ▶ Don't allow "us vs. you" attitude
- ▶ Work to eliminate red-tape
- ▶ Don't just report unverified bugs – work with development team to verify it
- ▶ Make the effort to become part of the team



Want More Information?

Information adapted from User Experience, not Metrics: Parts 6, 8, 9 and 10 and Beyond Performance Testing: Parts 6, 7, 8, 9, 10 located at <http://www-106.ibm.com/developerworks/rational/library/> (RDW) and <http://www.PerfTestPlus.com>

Good sources for additional information about Performance Testing:

- ▶ <http://www.PerfTestPlus.com> (Methodology, Templates, Articles, Presentations)
- ▶ <http://www.loadtester.com> (Good articles and links)
- ▶ http://www.keynote.com/resources/resource_library.html (Good articles and statistics)

Graphical Presentation of Information – Edward Tufte, PhD. <http://www.edwardtufte.com> (Books and seminars)



Summary

Teams are built, they don't just happen.

Performance testers have to start the collaboration by being conversant with developers.

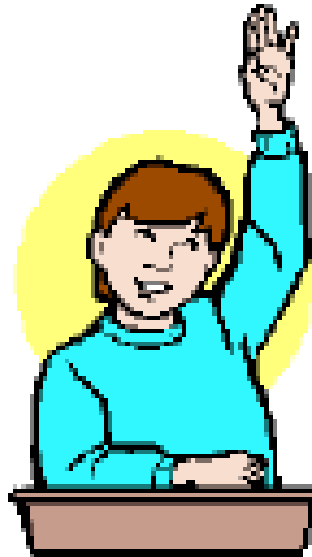
Developers will help you test better if you help them develop better.

Red-tape is poisonous.

Report symptoms, then help find causes.



Questions



Contact Info

Scott Barber

Chief Technology Officer

PerfTestPlus, Inc

E-mail:

sbarber@perftestplus.com

Web Site:

www.PerfTestPlus.com

