What to Performance Test:
Choose Wisely

Derived from:

Microsoft patterns & practices
Performance Testing Guidance for Web Applications
By: J.D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, Dennis Rea
© 2007 Microsoft Corporation. All rights reserved.
http://www.codeplex.com/PerfTestingGuide

Scott Barber
Chief Technologist
PerfTestPlus, Inc.
Scott Barber, CTO, PerfTestPlus, Inc.
sbarber@perftestplus.com
www.perftestplus.com

Executive Director, Association for Software Testing
www.associationforsoftwaretesting.org

Co-Founder, Workshop On Performance and Reliability
www.performance-workshop.org

Beautiful Testing
oreilly.com/catalog/9780596159825

Performance Testing Guidance for Web Applications
www.codeplex.com/PerfTestingGuide
www.amazon.com/gp/product/0735625700
Primary Goal of this Course

To teach you how to *think about*, *organize*, and *manage* performance testing effectively, under time and resource constraints, by *wisely choosing* which tests to design and execute to based on your project *context*.
I Assume That You:

Test software performance or manage someone(s) who does.
Have at least some control over the design of your tests and some time to create new tests.
Have at least some influence over your test environment.
Are worried that your test process is spending too much time and resources on things that aren’t important AND/OR
Are worried that your test process doesn’t leave enough time and resources to determine what IS important.
Believe that good testing requires thinking.

Test under uncertainty, resource limitations and time pressure.
Have a major goal to find important problems quickly.
Want to get very good at testing software performance.
Credits

Some of the material in this presentation was inspired by *High Performance Web Sites: Essential Knowledge for Front-End Engineers*, by Steve Souders, O’Reilly, 2007.

Some of this material was developed for, or inspired by, *Performance Testing Guidance for Web Applications*, a Microsoft patterns & practices book by J.D. Meier, Scott Barber, Carlos Farre, Prashant Bansode, and Dennis Rea.

Many ideas in this presentation were inspired or enhanced by colleagues including Alberto Savoia, Roland Stens, Richard Leeke, Mike Kelly, Nate White, Rob Sabourin, Chris Loosley, Ross Collard, Jon Bach, James Bach, Jerry Weinberg, Cem Kaner, Dawn Haynes, Karen Johnson, the entire WOPR community, and by students who took previous versions of this course, back to 2001.

Most of the concepts in this presentation are derived from publications, presentations, and research written and/or conducted by Scott Barber.

This course has been heavily influenced by:


*Just-In-Time Testing* (Robert Sabourin, ©1998-2007 Amibug, Inc.)
Performance Testing: An empirical, technical investigation conducted to provide stakeholders with information about the quality of the product or service under test with regard to speed, scalability and/or stability characteristics.

Performance Investigation: A deliberate data-collection and data-interpretation activity typically focused on data related to speed, scalability, and/or stability of the product under test. The collected data are primarily used to assess hypotheses about the root cause of one or more observed performance issues.

Performance Validation: A deliberate activity that compares speed, scalability and/or stability characteristics of the product under test to the expectations of representative users of the product.
In Other Words:

I help and/or teach individuals and organizations to optimize software systems by balancing:

- Cost
- Time to market
- Capacity

while remaining focused on the quality of service to system users.
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.

**Instrument**
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.
Why Test Performance?

To choose tests wisely, we must know what we are testing for.

To determine compliance with requirements?
To evaluate release readiness?
To assess user satisfaction?
To assist in performance tuning?
To estimate capacity?
To validate assumptions?
To generate marketing statements?
“Just test it” Isn’t Enough

Do you know your performance testing mission?

Do you know the “Commander’s Intent”?

Can you find out?

Example from my days as a U.S. Army LT:

**Mission:** Secure hilltop 42 NLT 0545 tomorrow.

**Commander’s Intent:** It is my intent that the supply convoy safely cross the bridge spanning the gorge between hilltop 42 and hilltop 57 between 0553 and 0558 tomorrow.
COPE in PUBS
What Matters

Performance Criteria are *boundaries* dictated or presumed by someone or something that matters.

**Goals**: Soft Boundaries (User Satisfaction)

**Requirements**: Firm Boundaries (Business or Legal)

**Thresholds**: Hard Boundaries (Laws of Physics)

**Constraints**: Arbitrary Boundaries (Budget or Timeline)
ACQUIRE

A ccept
Use your active listening skills

C onverse
Validate and demonstrate interest

Q uestion
Probe, educate, and learn

U nderstand
Summarize priorities and value statements

I nvestigate
Repeat with other stakeholders, make comparisons, prototype, quantify, etc.

R estate
Return with testable, quantified criteria and/or conflicting criteria/priorities

E volve
Embrace and communicate changing criteria
Performance Testing Objectives

What we actually hope to gain by testing performance

Are sometimes completely unrelated to stated requirements, goals, thresholds, or constraints

Should be the main drivers behind performance test design and planning

Usually indicate the performance-related priorities of project stakeholders

Will frequently override goals in “go-live” decisions

Now we can design tests to help us achieve our objectives!
Do I need this test to:

Investigate or Validate/Verify

End-to-End or Component

response Times and/or Resources utilized

under Anticipated or Stressful conditions
## FIBLOTS

<table>
<thead>
<tr>
<th>Frequent</th>
<th>Common activities (get from logs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intensive</td>
<td>e.g. Resource hogs (get from developers/admins)</td>
</tr>
<tr>
<td>Business Critical</td>
<td>Even if these activities are both rare and not risky</td>
</tr>
<tr>
<td>Legal or Contract</td>
<td>SLA’s, Contracts and other stuff that will get you sued</td>
</tr>
<tr>
<td>Obvious</td>
<td>What the users will see and are mostly likely to complain about. What is likely to earn you bad press</td>
</tr>
<tr>
<td>Technically Risky</td>
<td>New technologies, old technologies, places where it’s failed before, previously under-tested areas</td>
</tr>
<tr>
<td>Stakeholder Mandate</td>
<td>Don’t argue with the boss (too much)</td>
</tr>
</tbody>
</table>
Communicating System Usage
<table>
<thead>
<tr>
<th>SCORN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td>Media, HTML, styles &amp; scripts – compress &amp; minify.</td>
</tr>
<tr>
<td><strong>Caching</strong></td>
<td>The end-user’s browser cache can be your best friend, or your worst nightmare, use it wisely.</td>
</tr>
<tr>
<td><strong>Order</strong></td>
<td>Get the load order of your scripts and styles wrong, and you’ll lose your users every time – even though response time hasn’t changed!</td>
</tr>
<tr>
<td><strong>Response Codes</strong></td>
<td>3, 4, &amp; 5xx series response codes on individual objects are bad things.</td>
</tr>
<tr>
<td><strong>Number</strong></td>
<td>When it comes to performance, less is more (usually).</td>
</tr>
</tbody>
</table>
## Performance Test Strategy

**Intent of Investigation:**
Collect configuration data for tuning. Collect data to assist in validating existing network.

**Prerequisites:**
Static prototype deployed on future production hardware.

**Tasks:**
Determine network bandwidth, validate firewalls & load balancer, evaluate web server settings.

**Tools & Scripts:**
Load generation tool, HTTP scripts to request objects of various sizes from a pool of IP addresses.

**External Resources Needed:**
Firewall, Load Balancer, Network Admins, network monitors, 20 IP addresses for spoofing.

**Risks:**
Schedule delay, availability of administrators, configuration of load generation tool for IP spoofing.

**Data of Special Interest:**
Network bandwidth & latency, load balancer effectiveness, resource consumption, response times.

**Areas of Concern:**
No internal expertise on load balancer configuration.

**Pass/Fail Criteria:**
Adequate available bandwidth, architectural assumptions validated.

**Completion Criteria:**
Critical data collected and assumptions validated.

**Planned Variants:**
1 to 20 IPs, volume of 1 to 500, size from 1Kb to 1mb, configuration settings.

**Execution Duration(s):**
6 days: 2 days ea. network & bandwidth, firewall and load balancer, web server configuration.
## Performance Testing Principles

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

Scott Barber
Chief Technologist
PerfTestPlus, Inc

E-mail: sbarber@perftestplus.com
Web Site: www.PerfTestPlus.com