Quick, Easy, Cheap & Valuable Performance Testing…

…for Every Member of the Team

Created for:

STANZ 2010

The Duxton Hotel
Wellington City, NZ
23 August, 2010

The Swissotel Sydney
Sydney, AU
26 August, 2010

Scott Barber

Chief Technologist

PerfTestPlus, Inc.

Scott Barber

CTO, PerfTestPlus, Inc.
sbarber@perftestplus.com
www.perftestplus.com

Association for Software Testing
ast
Life Member

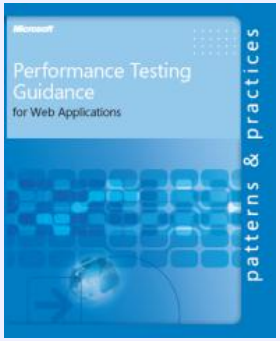
Co-Founder:
Workshop On Performance and Reliability
www.performance-workshop.org

Co-Author:

Beautiful Testing

oreilly.com/catalog/9780596159825

Performance Testing Guidance for Web Applications



www.codeplex.com/PerfTestingGuide
www.amazon.com/gp/product/0735625700

What is Performance Testing (Part 1)

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 with the purpose of assessing value or mitigating risk.

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.

What is Performance Testing (Part 2)

What mom tells people:

I help people make websites go fast.

What I tell people:

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.*

What is Performance Testing (Part 3)

In effect:

Performance testing helps ***stakeholders*** make ***decisions*** regarding product ***value*** and project ***risk***; Specifically ***value*** and ***risk*** related to ***speed***, ***scalability***, and ***stability*** attributes of a ***system*** and it's ***components*** throughout the ***product life-cycle***.

"Let's face the truth, performance testing

***IS* rocket science."**

--*Dawn Haynes*

… but even rocket science involves

***SOME* easy stuff.**

--Addendum added by: *Scott Barber*

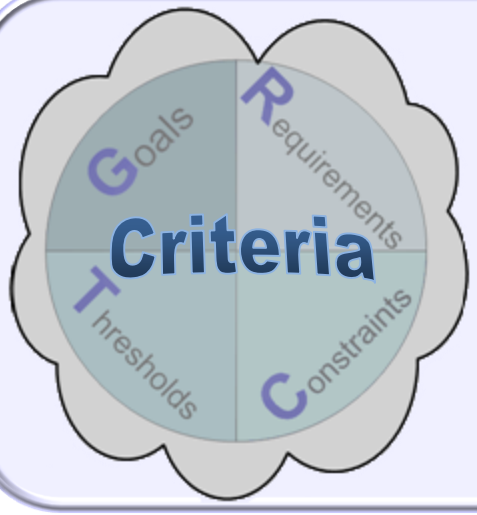
Quick, Easy, Cheap & Valuable

Item #1

Make Performance a:

Thoughts on Priority

- Focusing on performance adds value and mitigates risk from “bar napkin to delete key”.
- Get performance in the dev, test, & delivery plans.
- Don’t let performance fall off the plate.
- Be the advocate, even if it makes you “annoying”.



Goals: Soft Boundaries (User Satisfaction)

Requirements: Firm Boundaries (Business or Legal)

Thresholds: Hard Boundaries (Laws of Physics)

Constraints: Arbitrary Boundaries (Budget or Timeline)

Item #2

Ask lots of:

Thoughts on Questions

- How is the performance today?
- How will this [change] effect performance?
- Go to dev, test, & management meetings – and ask those questions.
- Advocate performance through questions.
- Earning the “annoying” label for asking questions is ok, but becoming “annoying” via your response to their answers isn’t.

"Ok, that stuff is 'quick', 'easy' & 'cheap'

and I can see the value,

*but what about the *testing*?!*

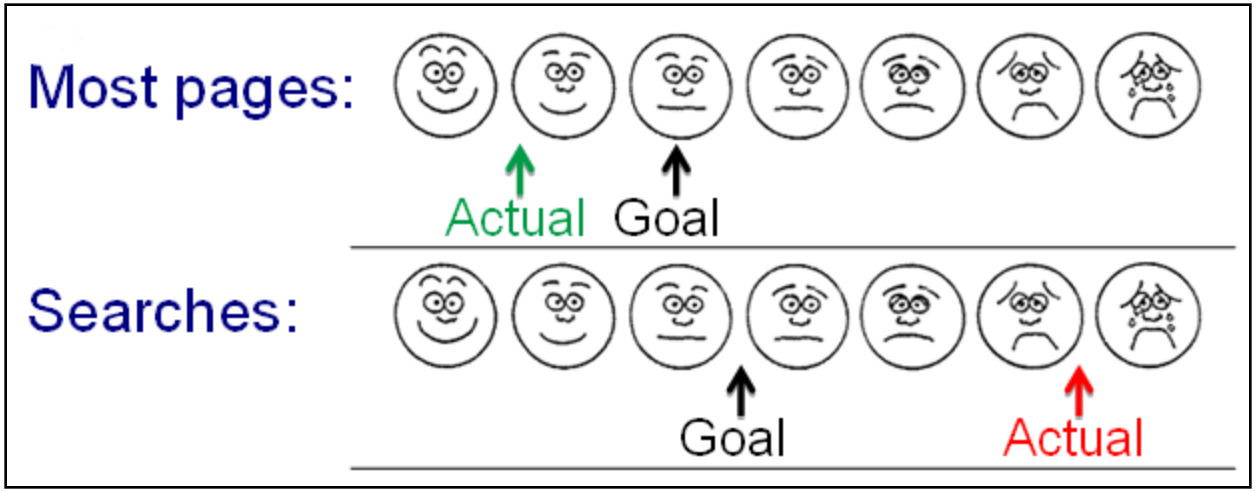
Quick, Easy, Cheap & Valuable

Item #3

Am I annoyed?

Thoughts on Annoyance

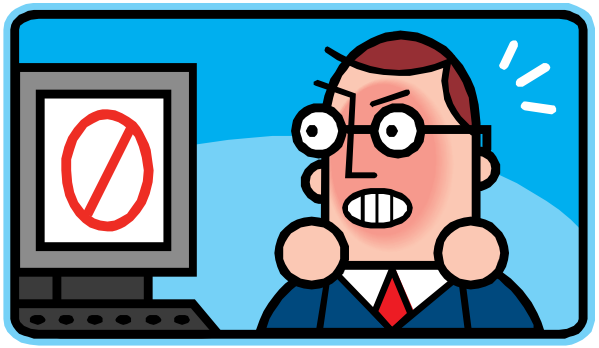
- Why am I annoyed?
- How annoyed am I?
- Does this annoy me all the time, or just sometimes?
- What impact is this likely to have on product value?
- Advocate something better.



Quick, Easy, Cheap & Valuable

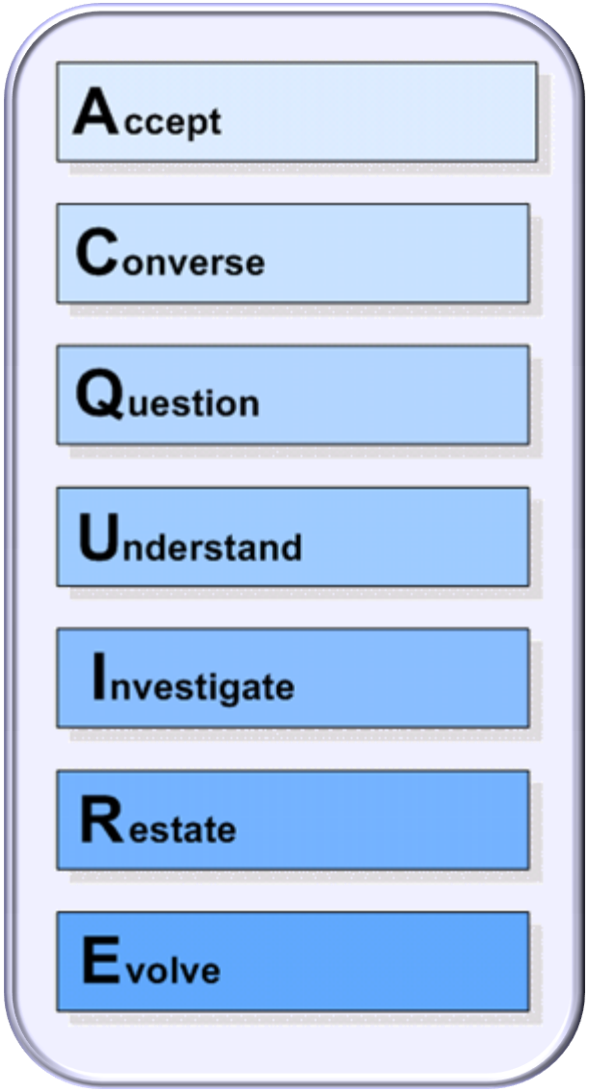
Item #4

Who else is annoyed?



More Thoughts on Annoyance

- Who matters?
- How do I get their feedback?
- Are they annoyed with performance, or workflow, or, or, or…?
- Advocate something better.



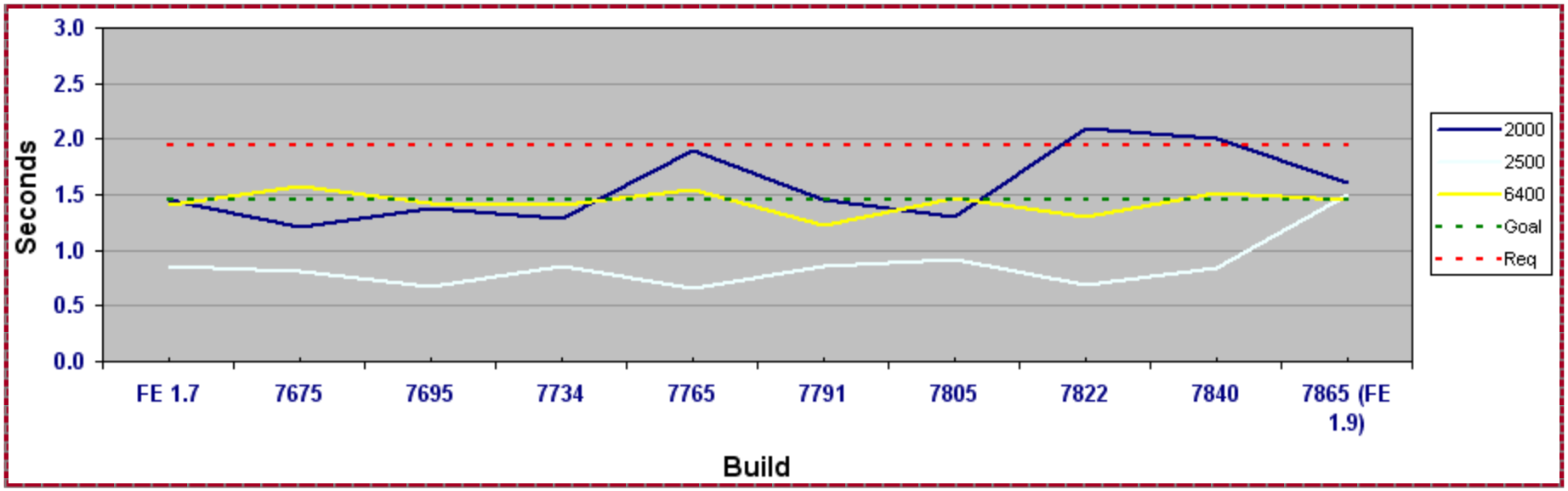
Item #5

Determine:

How Fast?

Speedy Speed Collection

- http://www.websiteoptimization.com/services/analyze/
- http://www.websitepulse.com/help/tools.php
- http://webwait.com/



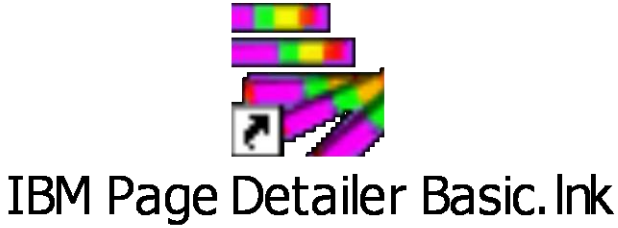
If you don't have speed targets, don't fret...

Trends are trendy!

Speedy Speed Collection (and more)

- *Microsoft Visual Round Trip Analyzer*

- *IBM Page Detailer Basic*



Item #6

Research the

How are “leaders” doing?

- *Keynote Systems*
- *Gomez Benchmarks*
- *WebMetrics*
- *The eService Index*

Quick, Easy, Cheap & Valuable

Item #7

Test the Front-End with

What is SCORN, anyway?

Size — Media, HTML, styles & scripts – compress & minify.

Caching — The end-user's browser cache can be your best friend, or your worst nightmare, use it wisely.

Order — 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!

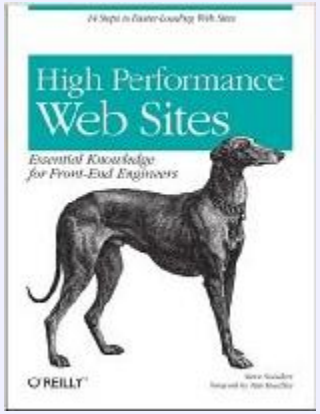
Response Codes — 3, 4, & 5xx series response codes on individual objects are bad things.

Number — When it comes to performance, less is more (usually).

SCORN References

- *High Performance Web Sites: Essential Knowledge for Front-End Engineers*, by Steve Souders, O'Reilly, 2007.
- *Yahoo! YSlow for Firebug*
- *Page Speed*
- *Right Click -> View Source and other Tips for Performance Testing the Front End*, by Scott Barber, for AST Update, 2007.

High Performance Web Sites: Essential Knowledge for Front-End Engineers



www.amazon.com/dp/0596529309

Quick, Easy, Cheap & Valuable

Item #8

Proceed with caution,
the following is only easy for hard-core

Easy Stuff for Geeks

- *Firefox Performance Tester's Pack*
- *Web Site Test Tools and Site Management Tools*
- *Fiddler*
- *Web Development Helper*

ReCap

- Make performance a *priority*.
- Ask lots of *questions*.
- Am I *annoyed*?
- *Who else* is annoyed?
- Determine *how fast*.
- *Research* the competition.
- Test the front-end with *SCORN*.
- Embrace your inner *geek*.

Questions

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.

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, and the entire WOPR community.

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

Many ideas were improved by students of various courses taught by Scott Barber, back to 2001.

Contact Info

Scott Barber

Chief Technologist

PerfTestPlus, Inc

E-mail:

sbarber@perftestplus.com

Web Site:

www.PerfTestPlus.com