



Performance Testing Challenges

First Presented for:

*Florida Institute of Technology
Computer Science Seminar*

Scott Barber
Chief Technology Officer
PerfTestPlus, Inc.



Agenda

“We need Performance Testing, but what is it?”

“I know you want it to be fast, but *how* fast?”

“What do you mean, what will the 500 users be doing?”

“*Houston, we have a problem...* but is it the application or the tool?”

“What do I do with all this data?”

“That’s a cool chart, but is the performance good or bad?”

“Does that mean we’re done?”



What is Performance Testing?

Performance Testing Determines...

- Speed
- Scalability
- Stability
- Confidence

...while focusing on

- User Expectations
- System Constraints
- Costs

Specifically, it answers:

- How many...?
- How much...?
- What happens if...?



What is Performance Testing?

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.

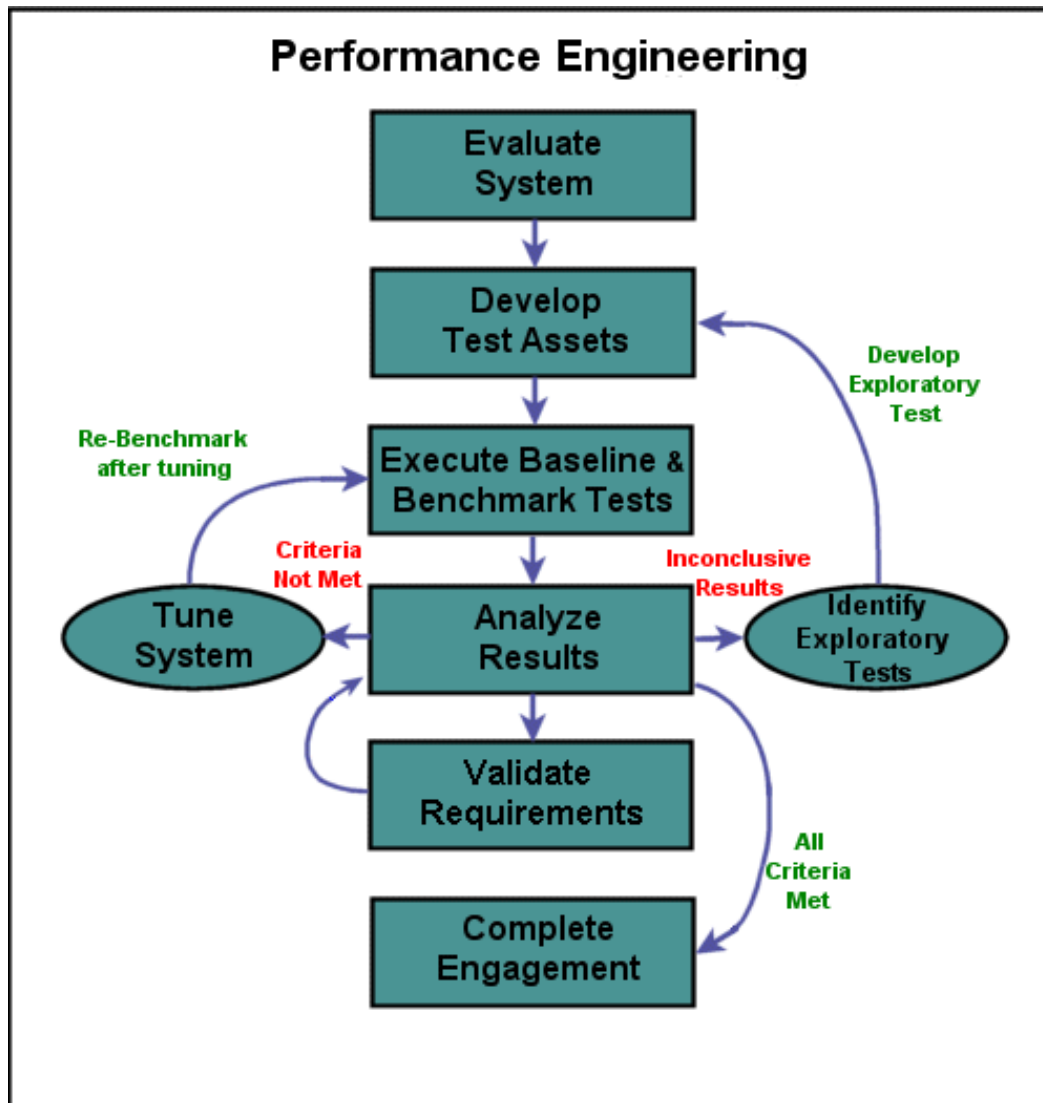
The peace of mind that it will work on go-live day alone justifies the cost of performance testing.



So how does it work?



What is Performance Testing?



Evaluate System
Develop Test Assets
Baselines and Benchmarks
Analyze Results
Tune
Identify Exploratory Tests
Execute Scheduled Tests
Complete Engagement



How Fast is Fast Enough?

There are no industry standards!

Based on system context, determine:

- User Psychology
- System Considerations
- Usage Considerations

Then assess:

- User Expectations
- Resource Limitations
- Stakeholder Expectations

Finally determine:

- Speed, Scalability and Stability Requirements for
 - User Experience measurements, not component metrics.
 - Batch type requirements.
 - Exception requirements.
 - Component metrics (use sparingly!).
- Composite Requirements for *your application!*



How Fast is Fast Enough?

What our clients can articulate usually aren't quantitative performance requirements.

The quantitative performance requirements our clients know, they usually can't articulate.

We performance analysts can articulate quantitative performance requirements, but we usually don't know what they are.

And that is why determining performance requirements is an iterative process."

*Derived from experience reports and discussions presented at the Workshop On Performance and Reliability #1 (WOPR1), Oct 2003. Attendees were Calvin Arnason, James Bach, Scott Barber, Ross Collard, Linda Hamm, Douglas Hoffman, Paul Holland, David Jewell, Chris Johnson, Philip Joung, Nancy Landau, Jude McQuaid, Alan Newman, Alexander Podelko, Robert Sabourin, Bill Schonger, Andrew Sliwkowski and Roland Stens



What Do The Simulated Users Do?

Using **Documentation**, **Interviews** and **Exploration**...

Determine the System's **Purpose** and what **Activities** it is used for.

Then **Measure** or **Estimate**:

- All possible user activity.
- How often each activity is accomplished.
- All “types” of users.
- What activities are most performance intensive.
- Other user community modeling information.

So you can model “**Real**” users.

Why model real users?



What Do The Simulated Users Do?

Because:

Results from inaccurately modeled tests are nearly always inaccurate, and often lead to incorrect decisions.

The only way to predict actual user experience (end-to-end response time) is to execute tests using realistic User Community Model(s).

Extrapolating expected performance based on incomplete models doesn't work.

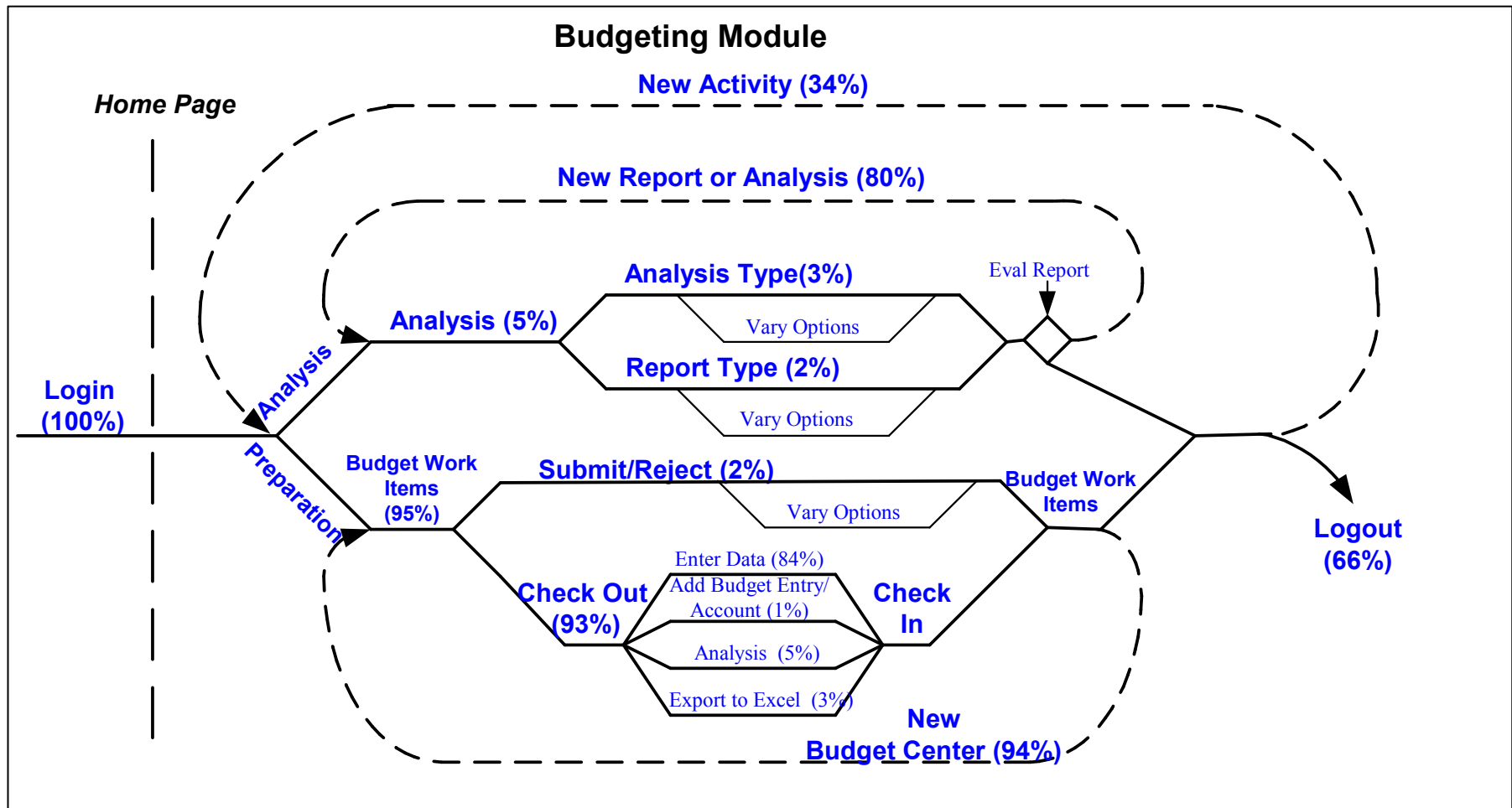
"The one thing that matters the most is not how your site behaves under theoretical or simulated conditions, but how well it works when you plug it into the wall and let everyone come hit your box from all across the world"

— Serdar Yegulalp in "Website Stress-Testing"



What Do The Simulated Users Do?

Build a visual model



Load Generation Tools Ain't Perfect

Popular Tools

- Mercury (LoadRunner, Astra LoadTest)
- Rational Robot (previously Performance Studio)
- Segue SilkPerformer
- Empirix E-Test Suite
- Radview WebLoad
- Compuware QALoad
- OpenSTA
- Microsoft Web Application Stress Tool



Load Generation Tools Ain't Perfect

How They (generally) Work

- Capture protocol level traffic.
- Use scripting language to replay that traffic.
- Do not interact with client system on playback.
- Do not actually invoke browsers on playback.
- Can simulate many users from many machines (IP Spoofing).
- Do not evaluate correctness of returned page, just codes and sizes.
- If done correctly, no application tier can tell the difference.



Load Generation Tools Ain't Perfect

Load generation tools...

- do not interact with client side portions of the application.
- do not natively evaluate correctness of returned pages.
- often do not handle conditional navigation.
- do not handle abandonment well.

Scripting concepts

- Record – *EDIT* – playback
- Add data variance
- Add delays
- Add conditional logic
- Add code to evaluate correctness of key pages
- Add abandonment functions

Scripting is *not* as easy as the tool vendors would like you to believe!



Collecting Data is Hard, Analysis Harder

Things “only the Perf-Guy” Thinks About

Statistical Significance

- Percentiles
- Standard deviations
- Common sense

Sample Size - How many measurements are enough?

Outliers - “...rare occurrence outside of 3 standard deviations...” What?!?

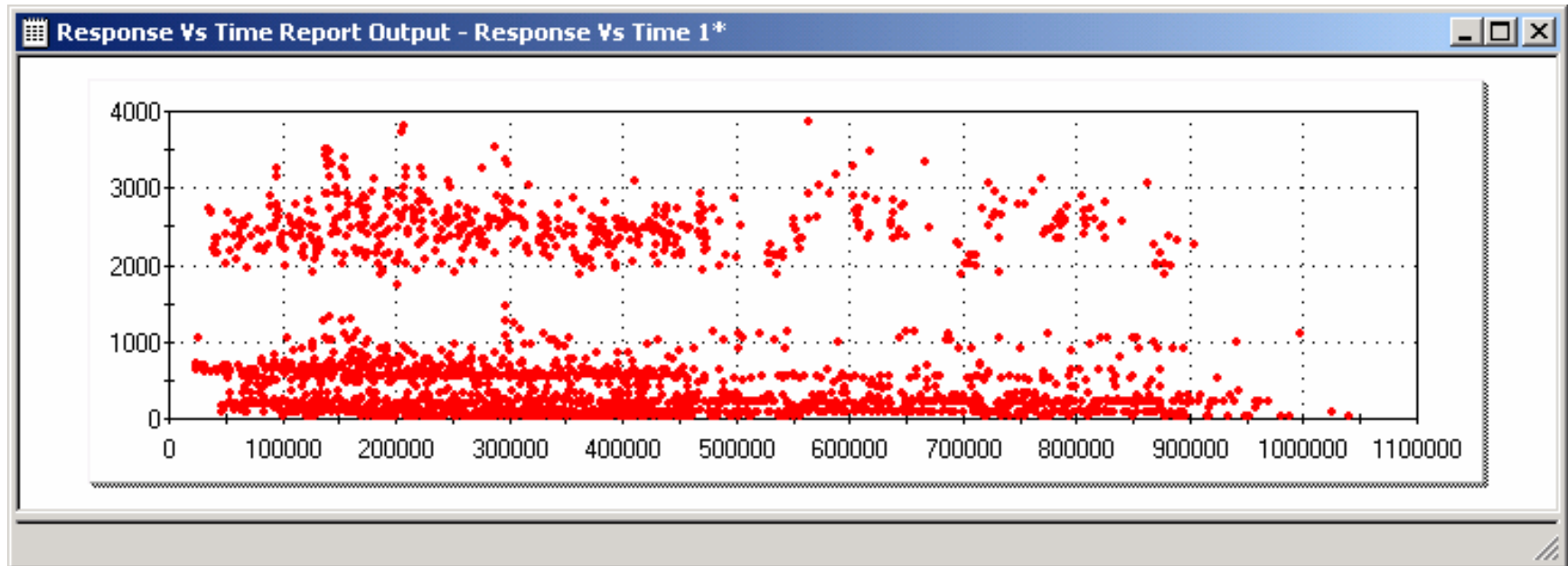
Confidence Interval - “How sure are you this is what users will experience in production?”

Identifying Real Issues vs. Symptoms



Collecting Data is Hard, Analysis Harder

Typical Tool Data Output:



Cool Chart, What Does it *MEAN*?!?

Why are clear charts so important?

Most people will never read performance test results docs.

Most people don't really understand the underlying components to performance.

It is our job to make it easy for them to understand, and understand quickly.

Being skilled at graphical presentation of technical information is critical for us to help others understand the message we are delivering.

Improper or confusing charts and tables can lead to wrong decisions that cause lost \$ and ruined reputations.



Cool Chart, What Does it *MEAN*?!?

What Stakeholders Want

Answers... NOW! (They might not even know the question)

To understand information intuitively.

Simple explanations of highly technical information.

To be able to make decisions quickly and have the information to support those decisions.

“Trigger phrases” to use with other stakeholders.

Concise summaries and conclusions.

Recommendations and options.



Cool Chart, What Does it *MEAN*?!?

What Stakeholders (usually) Get

Complex (verbal) technical descriptions.

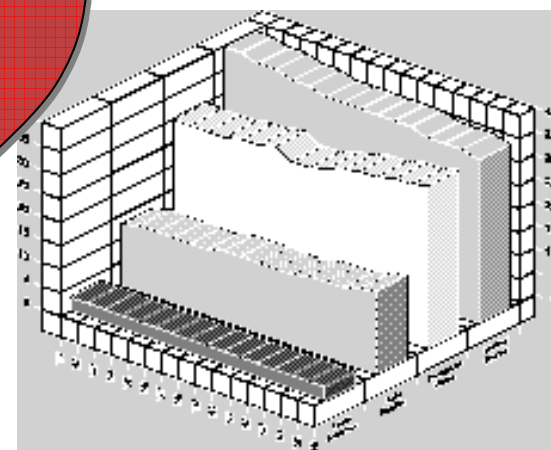
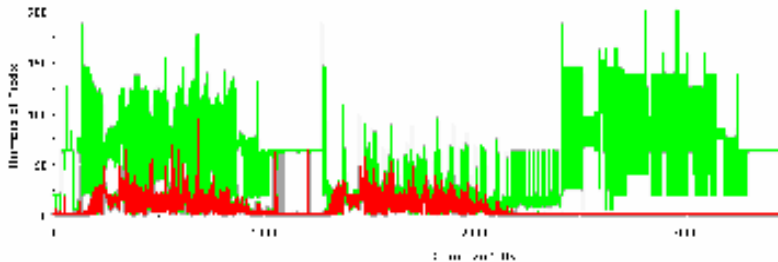
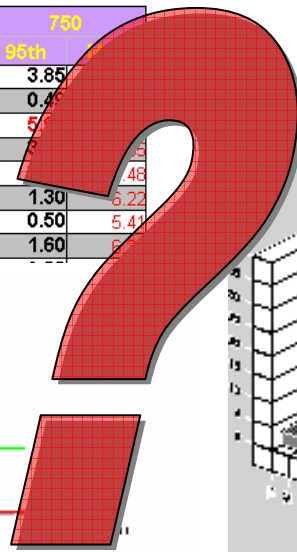
Complex technical graphics.

Raw data.

i.e. they get confused!



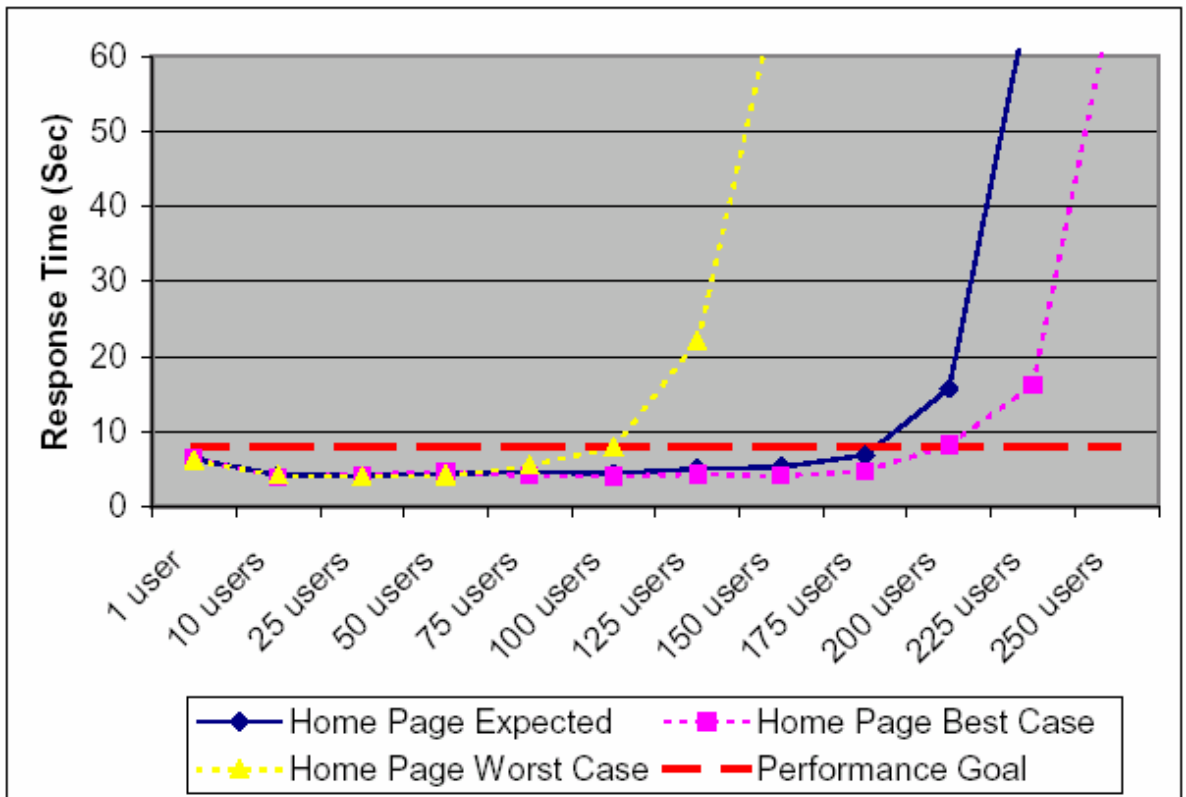
Timer Name	Baseline		250		500		750	
	55th	Max	55th	Max	55th	Max	55th	Max
ec_Main_Page	10.46	18.09	6.41	8.22	6.33	8.33	3.85	5.41
ec_logon_help	0.98	0.98	0.56	0.59	0.55	0.55	0.48	0.48
ec_login	5.35	7.92	6.66	11.84	6.75	17.03	5.67	10.61
quick_learns	6.66	6.67	5.91	10.98	5.92	11.02	5.67	10.61
view_quick_learn	15.66	17.11	5.53	10.72	3.89	10.61	4.8	10.61
view_faq_window	2.45	2.45	1.47	1.52	1.53	1.66	1.30	6.22
view_faq	0.67	0.67	0.60	0.63	0.58	0.69	0.50	5.41
view_ec_status	8.08	12.55	1.73	6.66	1.80	1.86	1.60	6.22



Cool Chart, What Does it *MEAN*?!?

The best general purpose performance graph I know...

Response Time Degradation Curve



Are we Done?

Not conducting performance testing is very risky.

Conducting performance testing improves the application and gives stakeholders confidence on go-live day.

Using a methodology improves success.

Capturing requirements is difficult, but worth while.

Performance requirements:

- come from many different sources.
- should be based in reality.
- should focus on the end-users.
- need context.

Performance tests must simulate real users to be meaningful.

Visual user community modeling helps clients understand performance testing.



Are we Done?

Few people actually understand performance testing, but most people **THINK** they do.

The load generation tool should not drive the test design.

Most tools make scripting look easy, but it rarely is.

Results aren't always what they seem.

Scatter charts hold a wealth of analytical information.

The performance degradation curve will show your stakeholders what they need to know in a single glance.

Actual performance in production is your legacy.



Where to go for more information

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

<http://www.TestingReflections.com> (QA Blog Collection)

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

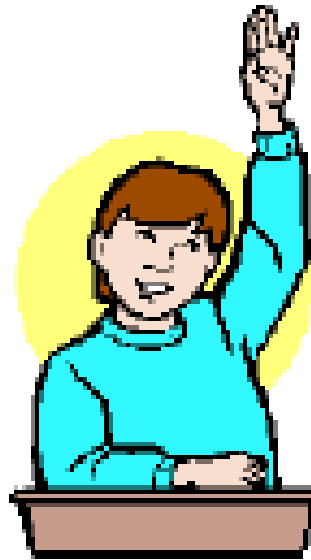
<http://www.loadtester.com> (Good articles and links)

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)



Questions



Contact Info

Scott Barber
PerfTestPlus, Inc

E-mail:

sbarber@perftestplus.com

Web Site:

www.perftestplus.com

