

Rapid Performance Testing: *No Load Generation Required*

Rapid Performance Testing

By:

Scott Barber
Chief Technologist
PerfTestPlus, Inc.



Chief Technologist, PerfTestPlus, Inc.

sbarber@perftestplus.com

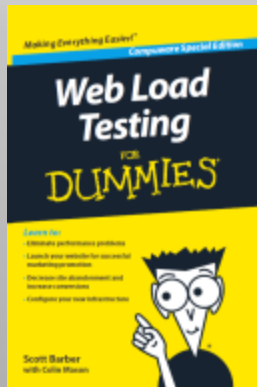
www.perftestplus.com

[@sbarber](https://twitter.com/sbarber)

Co-Founder: Workshop On Performance and Reliability

www.performance-workshop.org

Author:



Co-Author:



Contributing Author:



Books: www.perftestplus.com/pubs

About me: about.me/scott.barber

Performance Testing

Overview/Refresher

What is Performance?

System or application characteristics related to:

Speed:

- responsiveness
- user experience



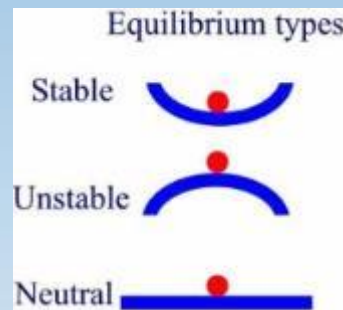
Scalability:

- capacity
- load
- volume



Stability:

- consistency
- reliability
- stress



What is Performance Testing?

Testing designed to:

Inform *software system optimization* by balancing:

- Cost
- Time to market
- Capacity

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

Performance vs. Load Testing?

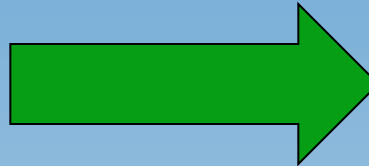
Performance is to Load

as

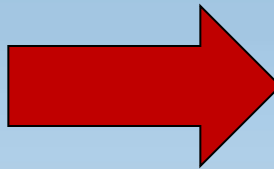
Rectangle is to Square

The Performance Lifecycle is:

Conception to Headstone



Not

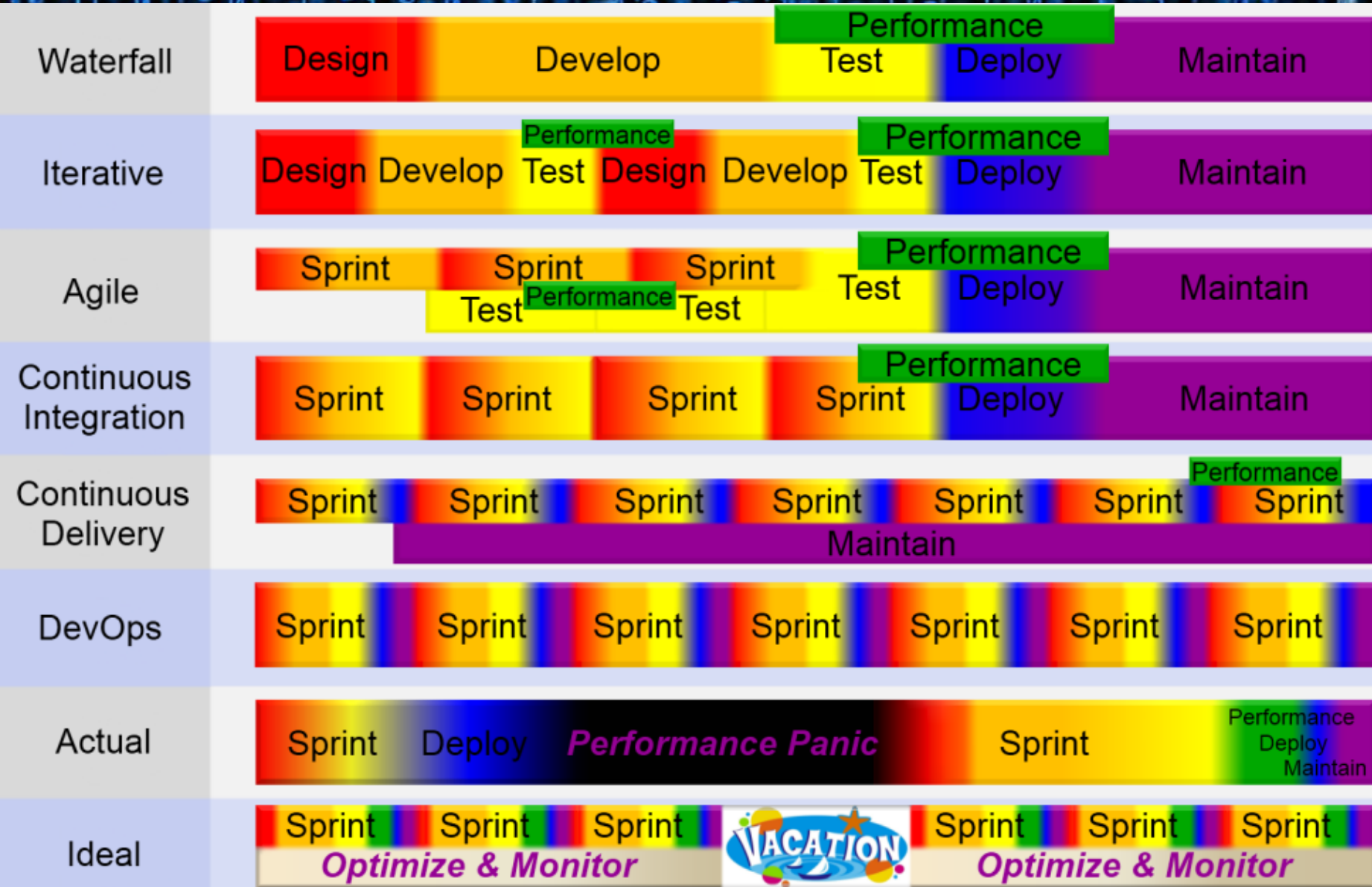


Cradle to Grave

Who is Responsible?

Everyone

Where does it fit in the SDLC?



Personally, I'm fond of...

Performant Application Delivery



Triad





Preventing Poor Performance with
a *little* work...
every day...
from *every* one.



Heuristic Production Simulation

**Rapid
Performance
Testing**

Key Points

- Entire Lifecycle
- Entire Team Focused
- Individual, Role, & Team Accountability & Responsibility
- Continuous & Collaborative



“DevOps”

- Code Profiling
- Requirements Revision
- Design Validation
- Budgets
- Units & Components
- Monitoring
- Tuning



Perf Team

- Load/Stress Testing
- Environment Config
- Capacity Planning
- Concurrency Checks
- Situation Analysis



Individual

- Snapshots
- Comparisons
- Niche Concerns
- Mini-Field Studies (UAT)
- “What-if...”

Rapid Performance Testing

... to address *random*
performance questions.

Evolved from:

*“What have we **got**?
What do we **want**?
How do we get there...?”*

--Bob Barber (Scott's dad)

*... as **quickly, simply, and**
cheaply as possible?*

*--Addendum added by: **Scott Barber***

Which is...

*...a common man's way of expressing the **problem solving approach that classical engineers employ.***

- **Given:** *“What have we got?”*
- **Find:** *“What do we want?”*
- **Solve:** *“How do we get there?”*

Premise

Value Begins with Clear Objectives

What **value** do we hope to gain?

*RPT questions are often **not** known requirements, goals, thresholds, or constraints*

Value should be the **main driver** behind performance test design and planning

*RPT questions often indicate the true **priorities** of stakeholders*

RPT answers will frequently **override requirements** in “go-live” decisions

Attribution

RPT is:

Inspired by Rapid Software Testing

Consistent with Rapid Software Testing themes

Sanctioned by James Bach, Michael Bolton & the RST instructors to as a specific implementation of the Rapid Testing Methodology

For more information about RST, visit:

http://www.satisfice.com/info_rst.shtml

What is it?

An approach to respond to a specific performance-related question after 4 or fewer hours of team effort with 1 or more of:

A) The answer

B) A partial answer

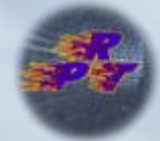
- **To determine the value of additional effort**
- **The level of effort to provide the answer**

C) Better questions to address the underlying concern



Conceptual Approach

1. *Receive Question*
2. *Generate Test Coverage Outline (TCO) (~20 min)*
3. *Transform TCO into Rapid Strategy (~20 min)*
4. *Execute Strategy (~2.5 hrs)*
5. *Consolidate/Analyze Data (~30 min)*
6. *Report Results (~20 min)*



Approach Discussion (Part 1)

1. *Receive Question*

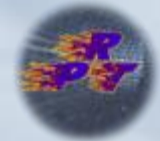
- *Clarify the question*
- *Understand the driver(s) behind the question*

2. *Generate Test Coverage Outline (TCO) (~20 min)*

- *Simplest path to (partial) answer(s)*
- *Comprehensive path to (partial) answer(s)*

3. *Transform TCO into Rapid Strategy (~20 min)*

- *Only tasks that fit in time box*
- *Stick to tasks requiring available resources*



Approach Discussion (Part 2)

4. *Execute Strategy (~2.5 hrs)*

- *Snapshots are your friends*
- *Anecdotal is sufficient*

5. *Consolidate/Analyze Data (~30 min)*

- *Identify patterns*
- *Confirm patterns (time permitting)*

6. *Report Results (~20 min)*

- *Answer(s)*
- *Time/Effort to answer(s)*
- *Follow-on questions of interest*

Addendum

Jessica's Story

The Bottom Line



+



+



=>

***A practical & holistic
framework for delivering
Performant Systems***

Questions?



Demo & Resources

Scott's "Secret RPT CheatSheet":

<http://qikpad.co.uk/p/RPT>

Rapid Strategy Examples:

<http://www.mindmeister.com/267329863?t=UkxsZQfDKF>

<http://www.mindmeister.com/267331147?t=D8tCoW05En>

<http://www.mindmeister.com/267331063?t=AiWwSdPusv>

Demo:

GTMetrix.com

WhichLoadsFaster.com

Contact Info

Scott Barber

about.me/scott.barber

Chief Technologist
PerfTestPlus, Inc.

E-mail:

sbarber@perftestplus.com

Web Site:

www.PerfTestPlus.com

Blog:

scott-barber.blogspot.com

Twitter:

[@sbarber](https://twitter.com/sbarber)