Fully Integrating Performance Testing:

Into Agile Development

Created for:

Agile
TESTING DAYS

By:

Scott Barber

Chief Technologist

PerfTestPlus, Inc.

PerfTestPlus

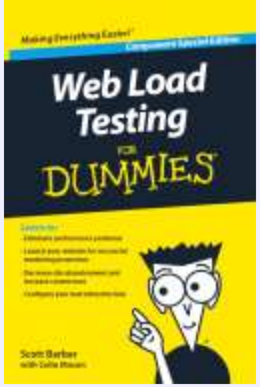
Fully Integrating Performance Testing: *Into Agile Development*

Association for Software Testing ast Life Member

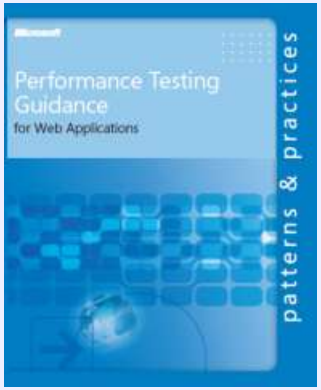
Scott Barber

Chief Technologist, PerfTestPlus, Inc.
sbarber@perftestplus.com
www.perftestplus.com
@sbarber
Co-Founder: Workshop On Performance and Reliability
www.performance-workshop.org

Author:



Co-Author:



Contributing Author:

How to Reduce the Cost of Software Testing
Edited by Matthew Heusser and Govind Kulkarni

Books: www.perftestplus.com/pubs

About me: about.me/scott.barber

Notes

This tutorial is a sub-set of 7 days of instruction (about the same as a 1 semester university course)

I do not believe in "Best Practices" (I believe in problem solving via experience, experiment & education)

Everything in this tutorial is based on personal experience in *some* context (but not yours)

Do adapt concepts to your situation/context

I *like* questions! (and challenges)

Do follow–up with me (what works for you… or doesn't)

To Get The Most From This Class

Ask questions (helping you understand is very important to me)

Do not be shy (did I mention that I *like* to be challenged and interact with you?)

Don't worry about what "the boss" will or won't allow (at least not before you I finish explaining)

Do laugh at my jokes (or groan – so I know if it didn't translate well)

Do network with others who are here (I bet they have great ideas too)

WARNING

- I speak 1 language (not very "naturally")
- I speak quickly (especially when I'm telling stories)
- I choose words very carefully (that may or may not translate well)
- Some suggestions are harsh (use at your own risk)

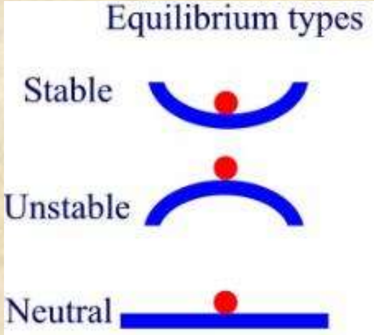
<Background>

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?

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.*

Performance vs. Load Testing?

Performance is to Load

as

Rectangle is to Square

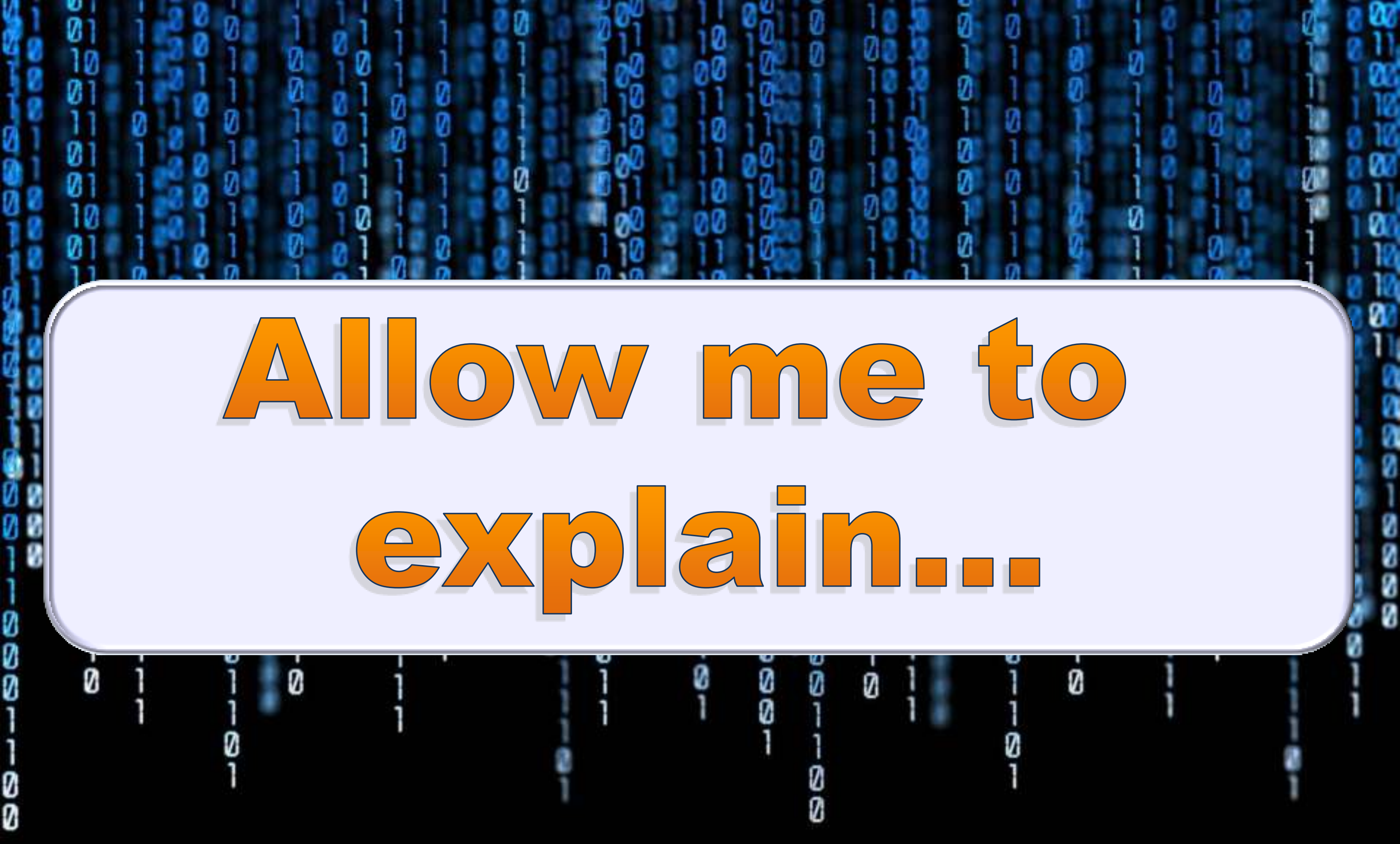
</Background>

Who is Responsible?

Everyone

Where Does it Fit?

Everywhere



Allow me to explain...

To Prevent Poor Performance…

To Prevent a Wreck

Left and Right
You Must Check

www.cap.gov

… don’t just react when it happens.

An Ounce of Prevention…

Copyright © Randy Glasbergen.
www.glasbergen.com
GLASBERGEN

"What fits your busy schedule better, exercising one hour a day or being dead 24 hours a day?"

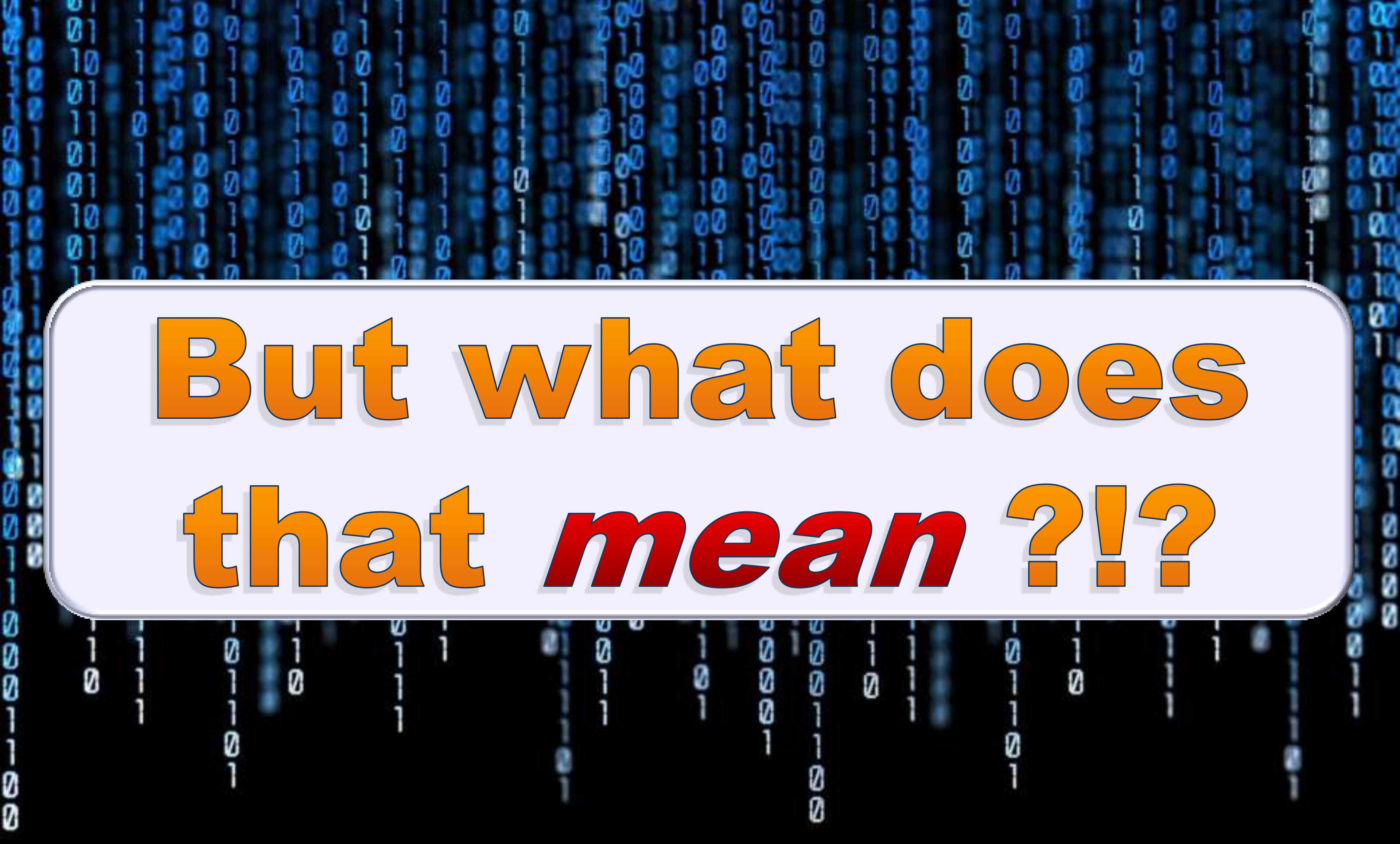
Which implies that...

The Performance Lifecycle is:

Conception to Headstone

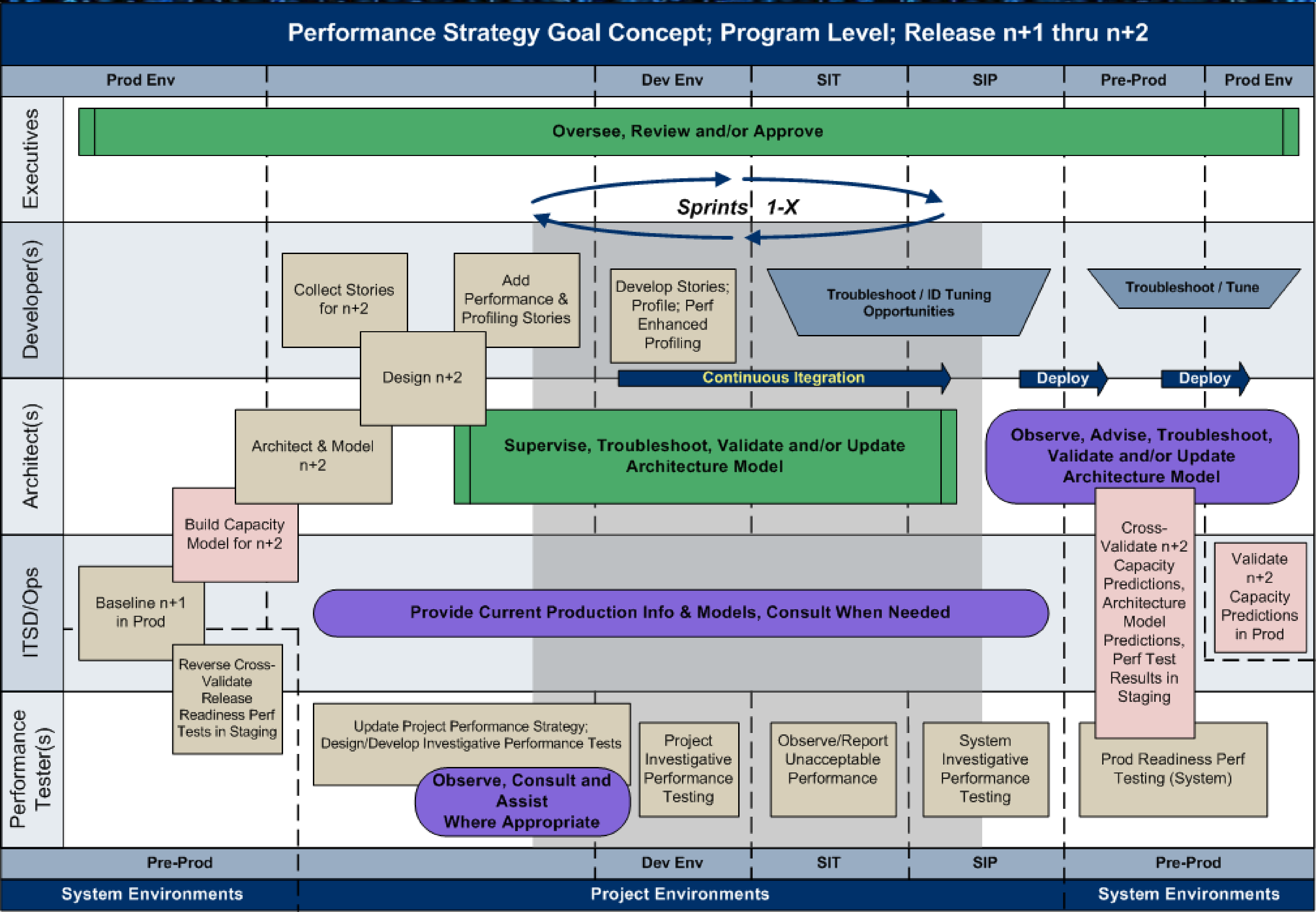
Not

Cradle to Grave



But what does that *mean* ?!?

…Do you like Swim Lanes?



Performance Strategy Goal Concept; Program Level; Release n+1 thru n+2

Prod Env | Dev Env | SIT | SIP | Pre-Prod | Prod Env

Executives

Oversee, Review and/or Approve

Sprints 1-X

Developer(s)

Collect Stories for n+2

Add Performance & Profiling Stories

Develop Stories; Profile; Perf Enhanced Profiling

Troubleshoot / ID Tuning Opportunities

Troubleshoot / Tune

Design n+2

Continuous Itegration

Deploy

Deploy

Architect(s)

Architect & Model n+2

Supervise, Troubleshoot, Validate and/or Update Architecture Model

Observe, Advise, Troubleshoot, Validate and/or Update Architecture Model

Build Capacity Model for n+2

ITSD/Ops

Baseline n+1 in Prod

Provide Current Production Info & Models, Consult When Needed

Cross-Validate n+2 Capacity Predictions, Architecture Model Predictions, Perf Test Results in Staging

Validate n+2 Capacity Predictions in Prod

Performance Tester(s)

Reverse Cross-Validate Release Readiness Perf Tests in Staging

Update Project Performance Strategy; Design/Develop Investigative Performance Tests

Observe, Consult and Assist Where Appropriate

Project Investigative Performance Testing

Observe/Report Unacceptable Performance

System Investigative Performance Testing

Prod Readiness Perf Testing (System)

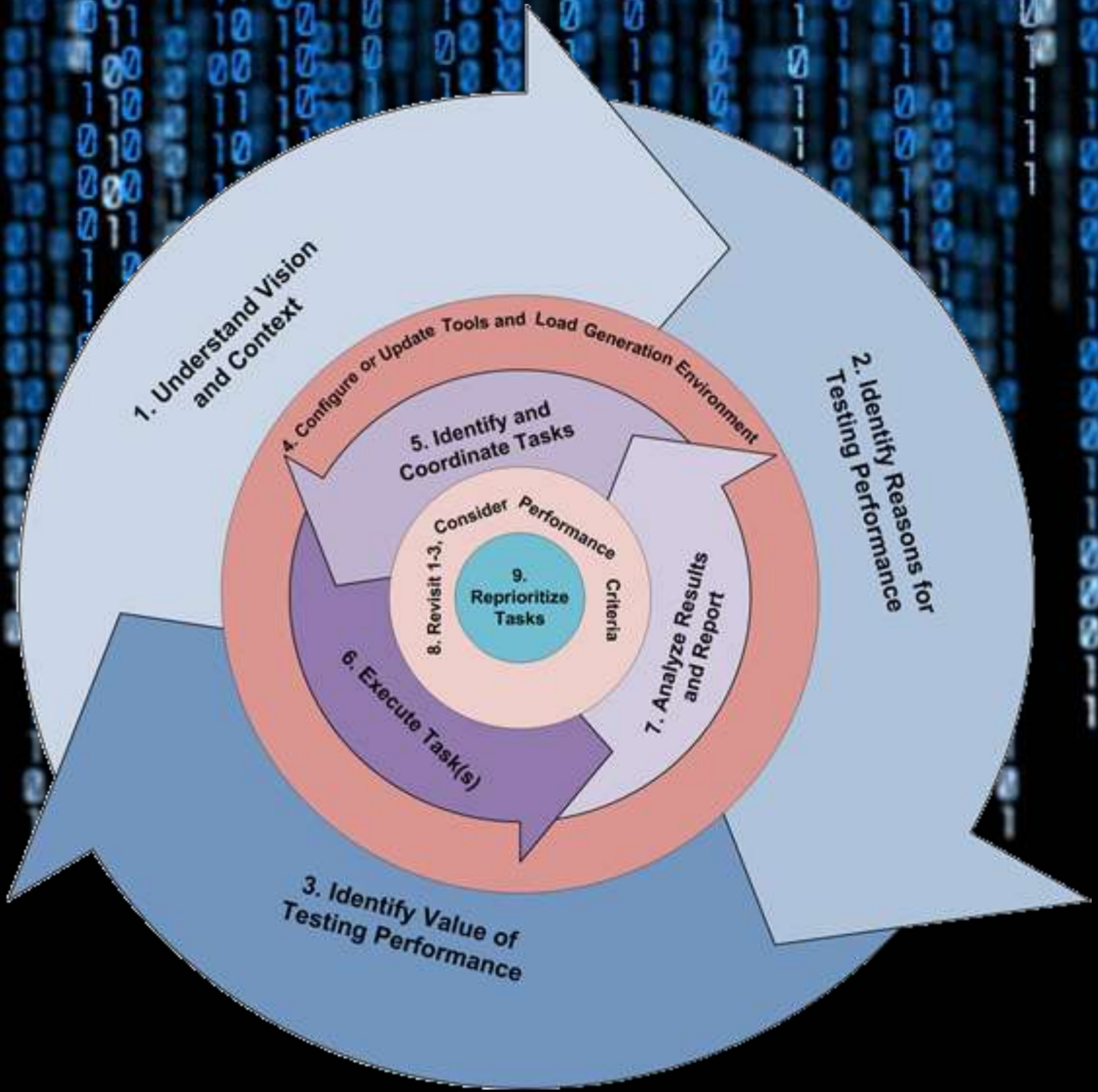
Pre-Prod | Dev Env | SIT | SIP | Pre-Prod

System Environments | Project Environments | System Environments

…Or maybe Circles?

Agile Performance Testing Activities

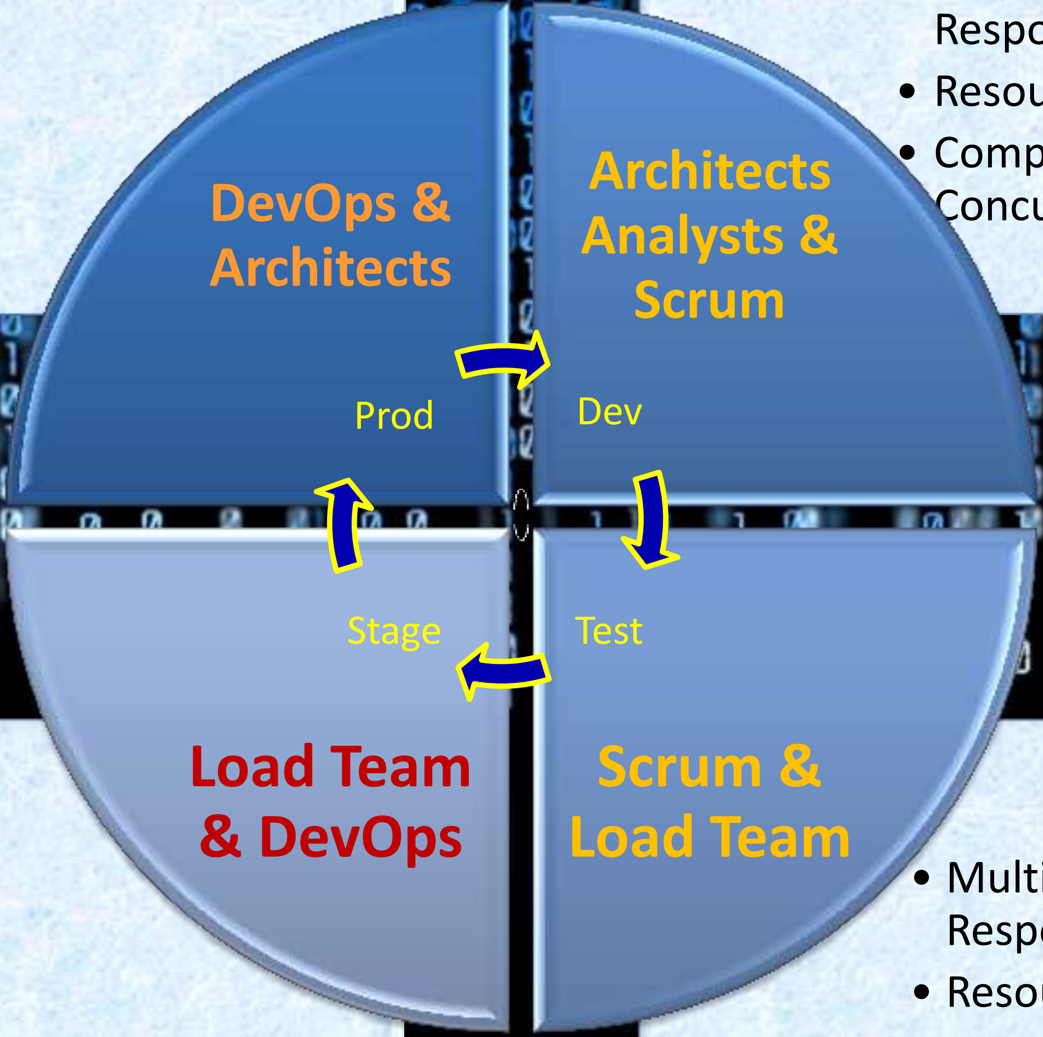
1. Understand Project Vision and Context
2. Identify Reasons for Testing Performance
3. Identify Value of Testing Performance
4. Configure or Update Tools and Load Generation Environment
5. Identify and Coordinate Tasks
6. Execute Task(s)
7. Analyze Results and Report
8. Revisit 1-3, Consider Performance Criteria
9. Reprioritize Tasks


1. Understand Vision and Context
2. Identify Reasons for Testing Performance
3. Identify Value of Testing Performance
4. Configure or Update Tools and Load Generation Environment
5. Identify and Coordinate Tasks
6. Execute Task(s)
7. Analyze Results and Report
8. Revisit 1-3, Consider Performance Criteria
9. Reprioritize Tasks

…Or Annotated Pie Charts?

- Current Capacity
- Projections
- Scalability Plan
- Reliability

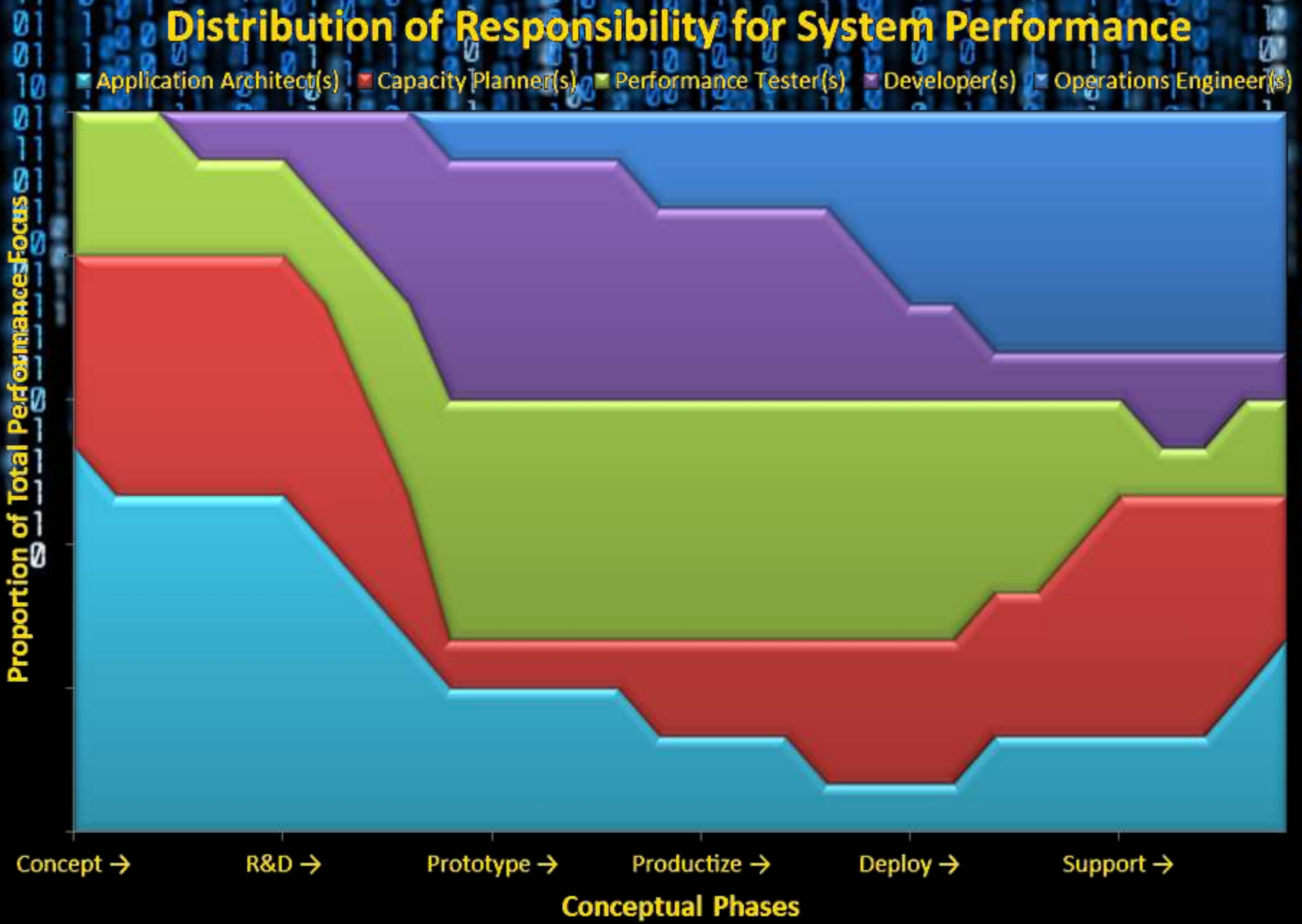
- Single User Responsiveness
- Resource Management
- Component-Level Concurrency



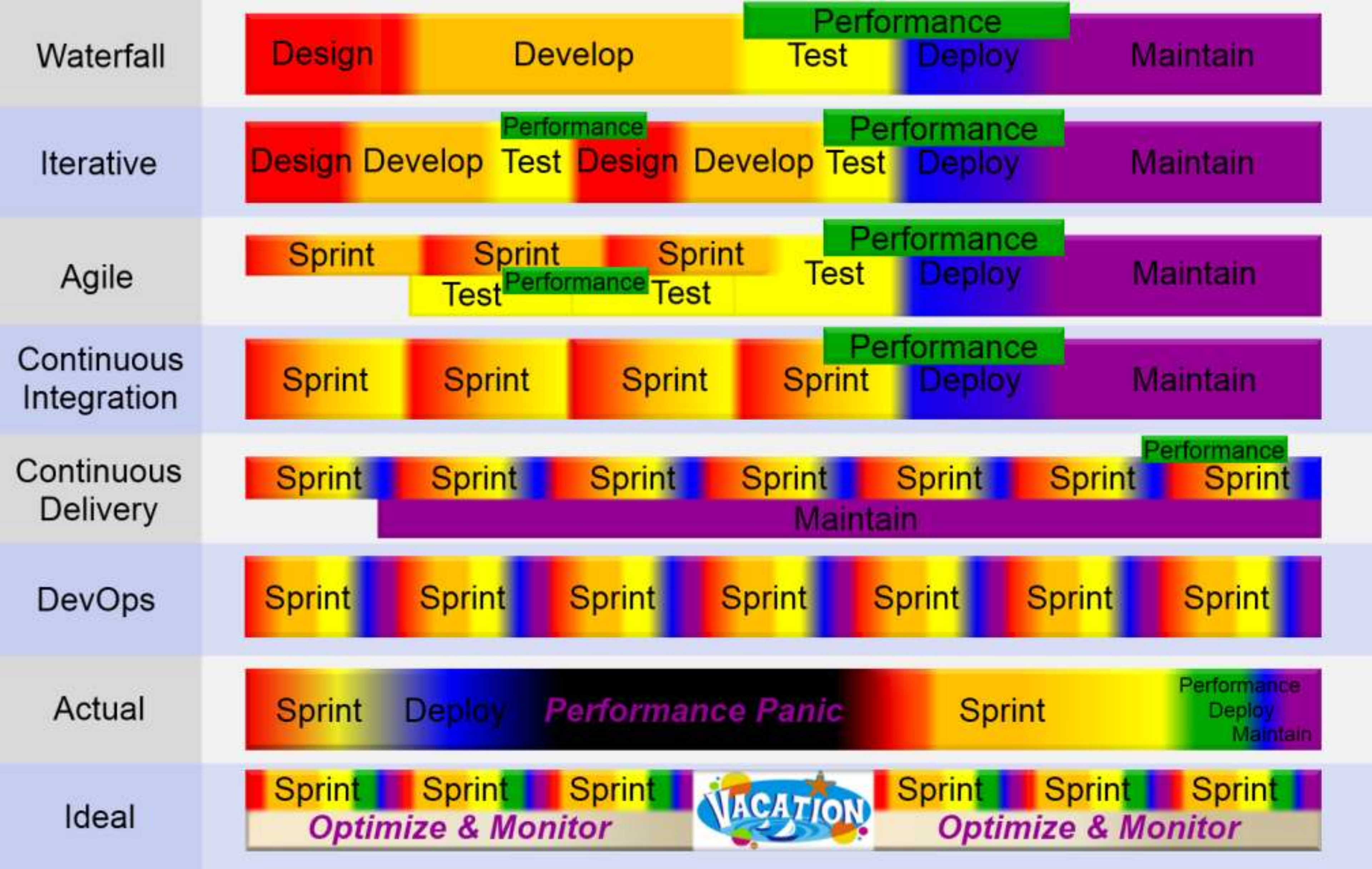
- Prepare for Prod
- Tune/Optimize
- Predict
- Early Warning Identification

- Multi-User Responsiveness
- Resource Trending
- Limit Identification

…Relativistic Comparisons?



…How about Colors?



Waterfall: Design | Develop | Performance Test | Deploy | Maintain

Iterative: Design | Develop | Performance Test | Design | Develop | Performance Test | Deploy | Maintain

Agile: Sprint | Test | Sprint | Performance | Sprint | Test | Test | Performance Deploy | Maintain

Continuous Integration: Sprint | Sprint | Sprint | Sprint | Performance Deploy | Maintain

Continuous Delivery: Sprint | Sprint | Sprint | Sprint | Sprint | Sprint | Performance Sprint | Maintain

DevOps: Sprint | Sprint | Sprint | Sprint | Sprint | Sprint | Sprint

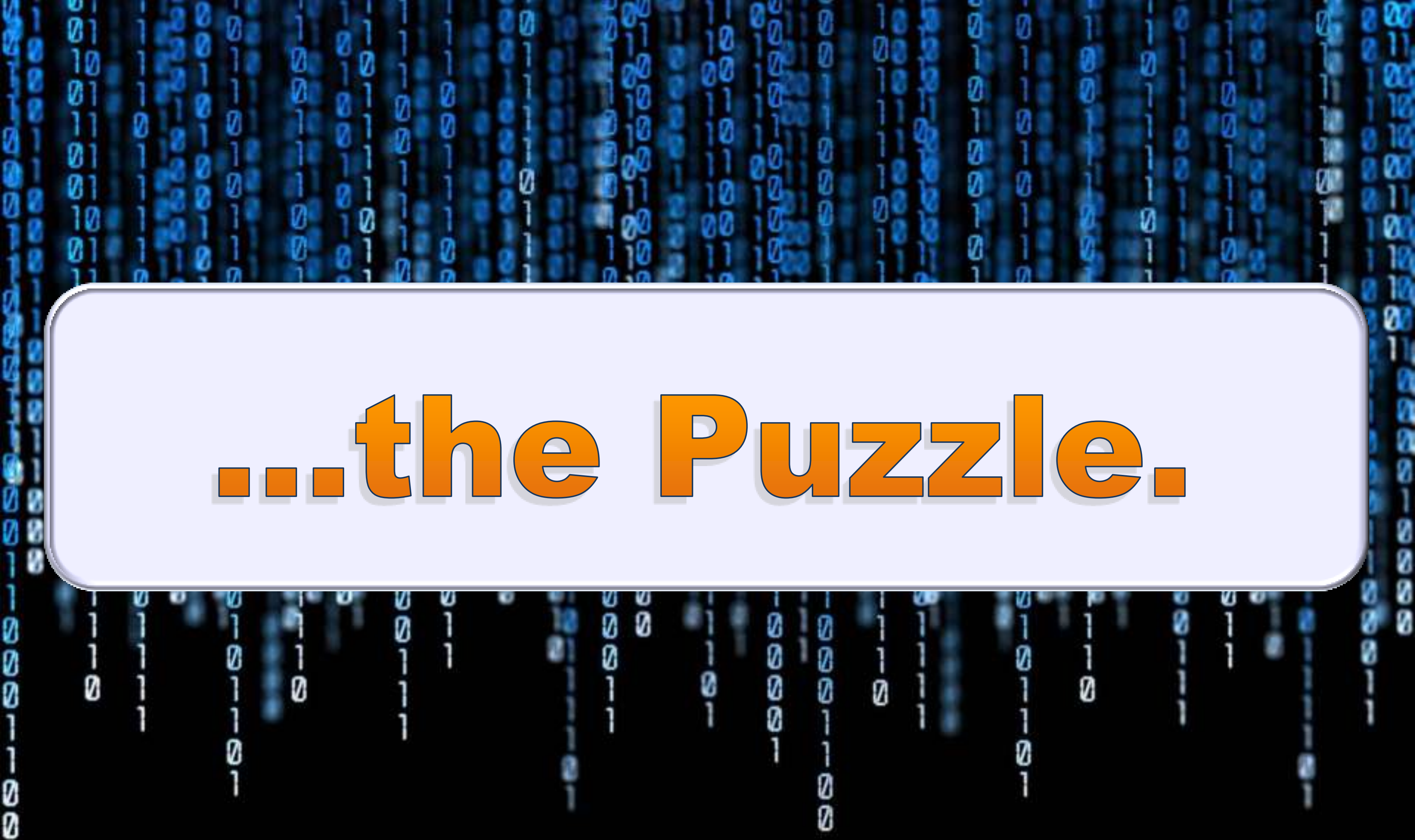
Actual: Sprint | Deploy | *Performance Panic* | Sprint | Performance Deploy Maintain

Ideal: Sprint | Sprint | Sprint | *Optimize & Monitor* | VACATION | Sprint | Sprint | Sprint | *Optimize & Monitor*

I’ve tried all of those…

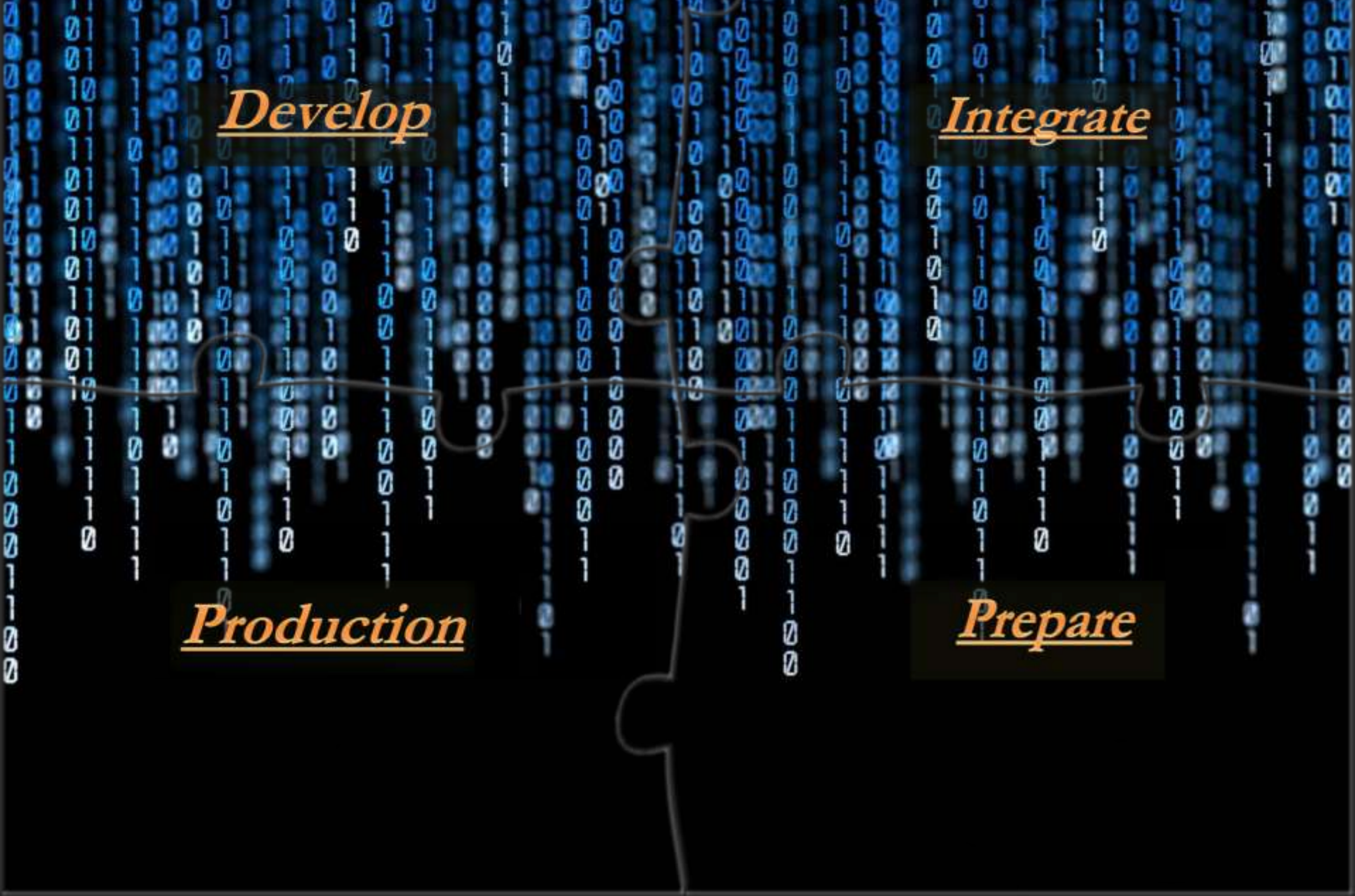
... But now

I prefer...



...the Puzzle.

Application Delivery Activities

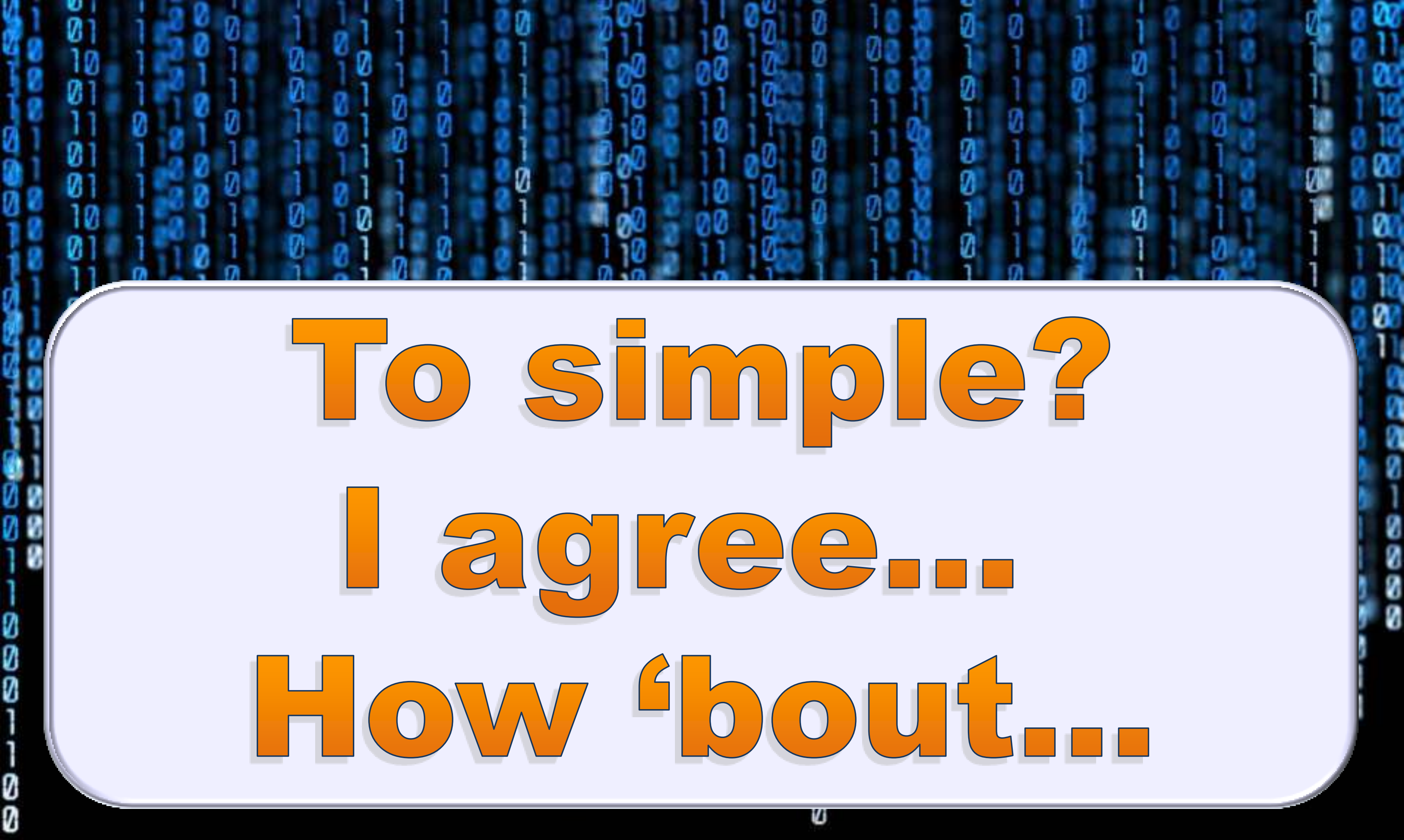


Develop

Integrate

Production

Prepare



To simple?
I agree…
How ‘bout…

Develop

(Sprint, Iteration, etc.)

Goals/Budgets

Unit Tests/Trends

Story Acceptance

Build Validation/Regression

Integrate

(Build, Merge, Promote, etc.)

Features

Target Load Profiles

Peak Load Profiles

Weakness Identification

Production

Deployments

Maintenance

Rapid Response

Data Forward

Prepare

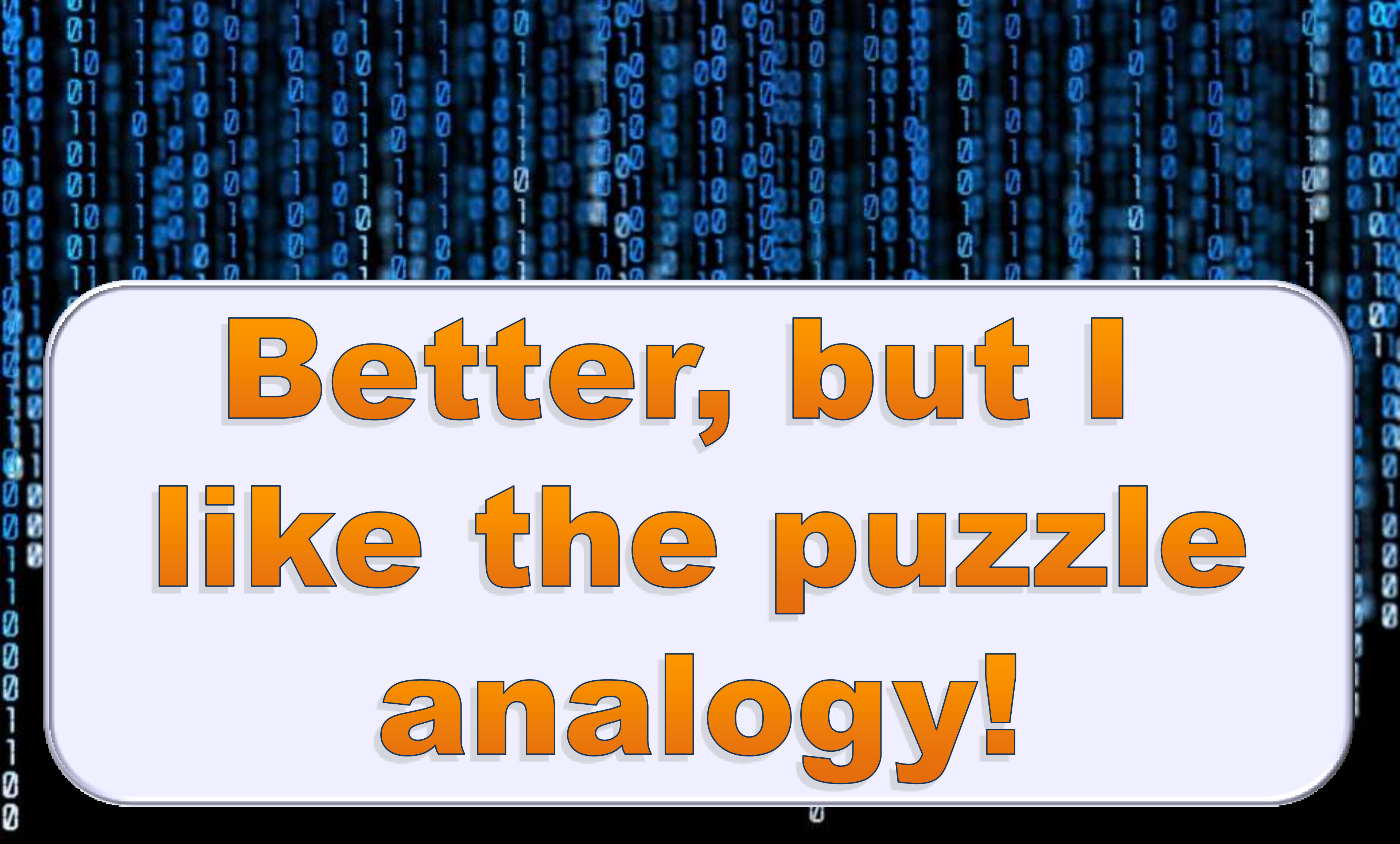
(Major Release, Configure, Certify, etc.)

External Integrations

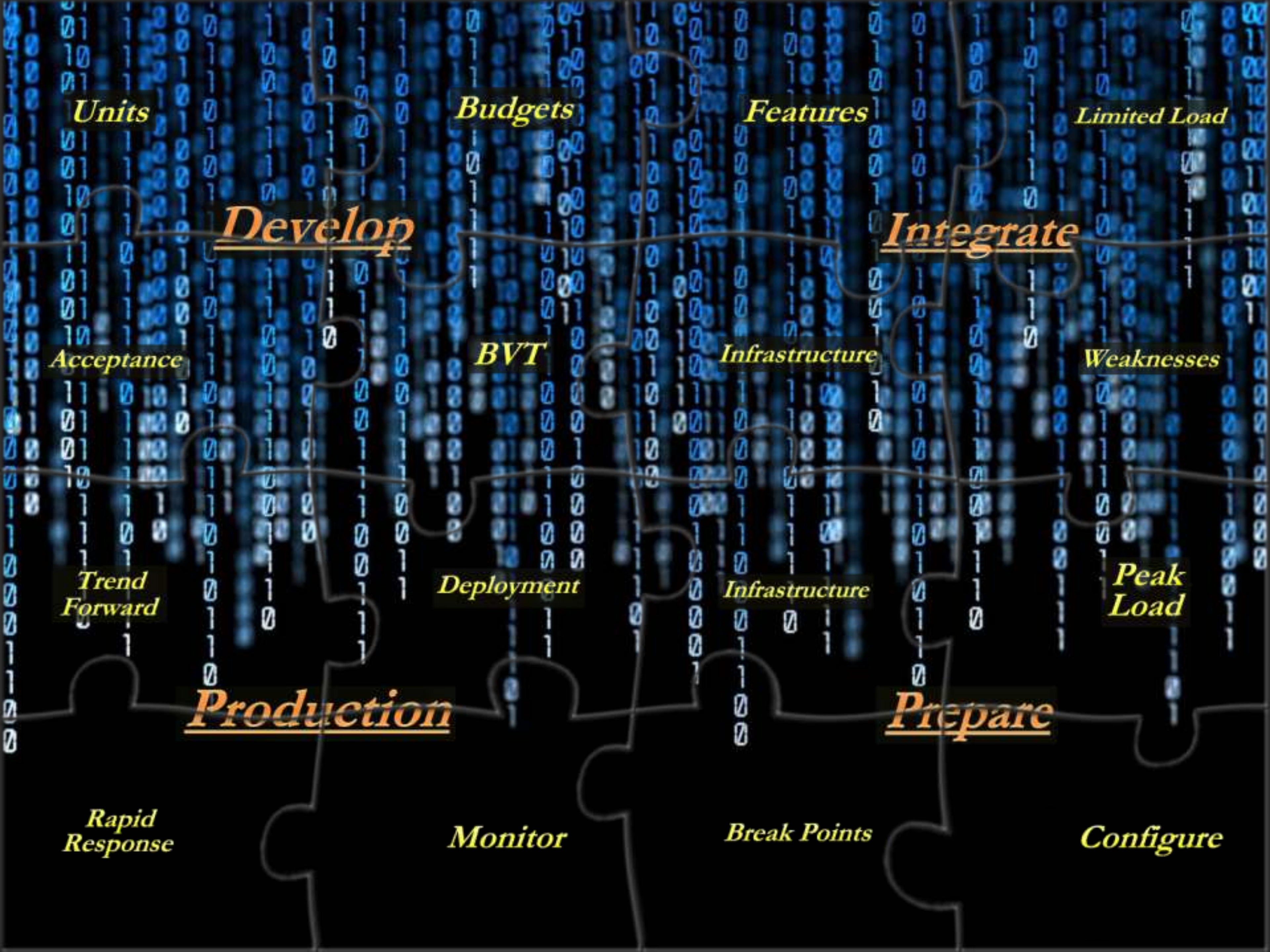
Infrastructure/Tuning

Break Point Identification

Capacity Planning



Better, but I like the puzzle analogy!



Develop

Units

Budgets

Acceptance

BVT

Integrate

Features

Limited Load

Infrastructure

Weaknesses

Production

Trend Forward

Deployment

Rapid Response

Monitor

Prepare

Infrastructure

Peak Load

Break Points

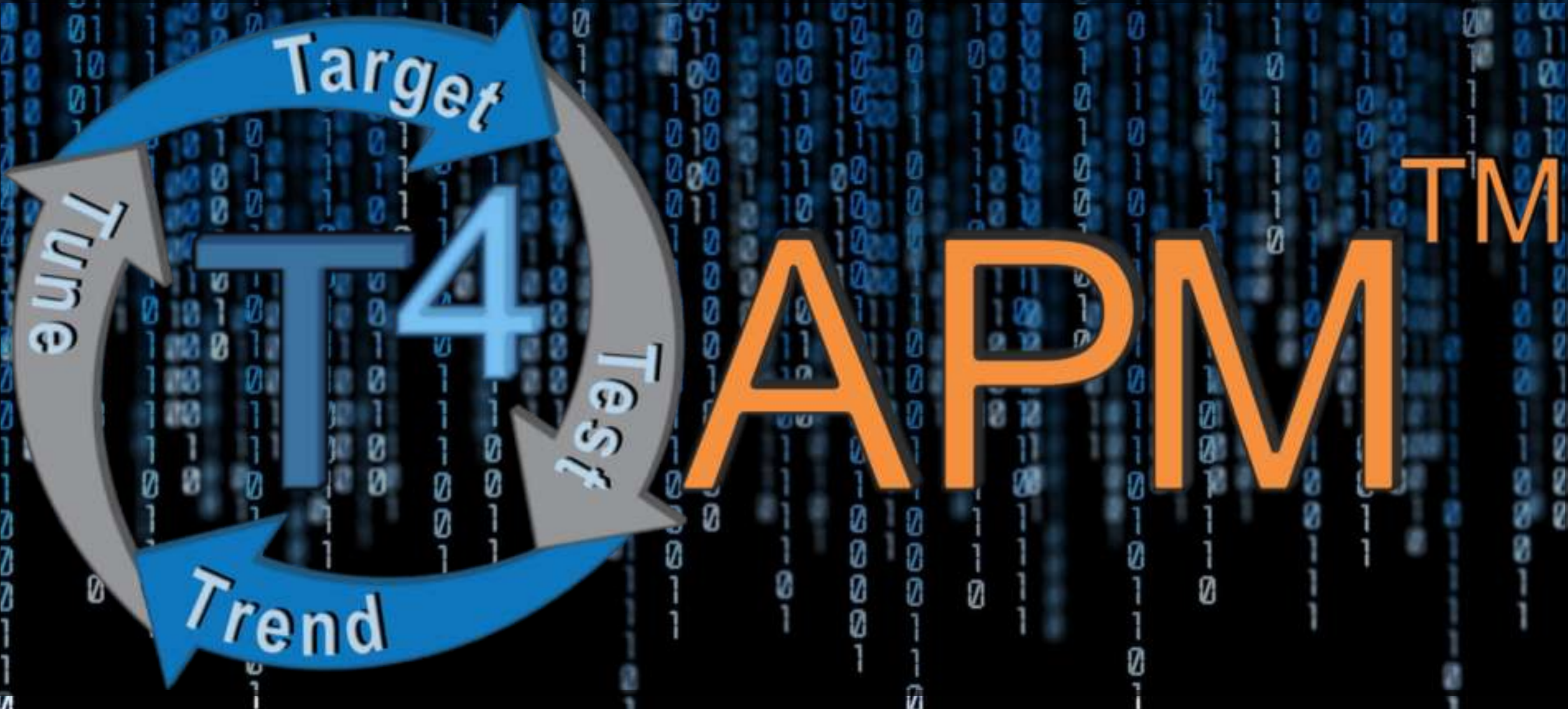
Configure

...and your team can improve performance on each puzzle piece!

... with just
a *little* work...
every day...
from *every* one.

Want to know how?

With:



Target

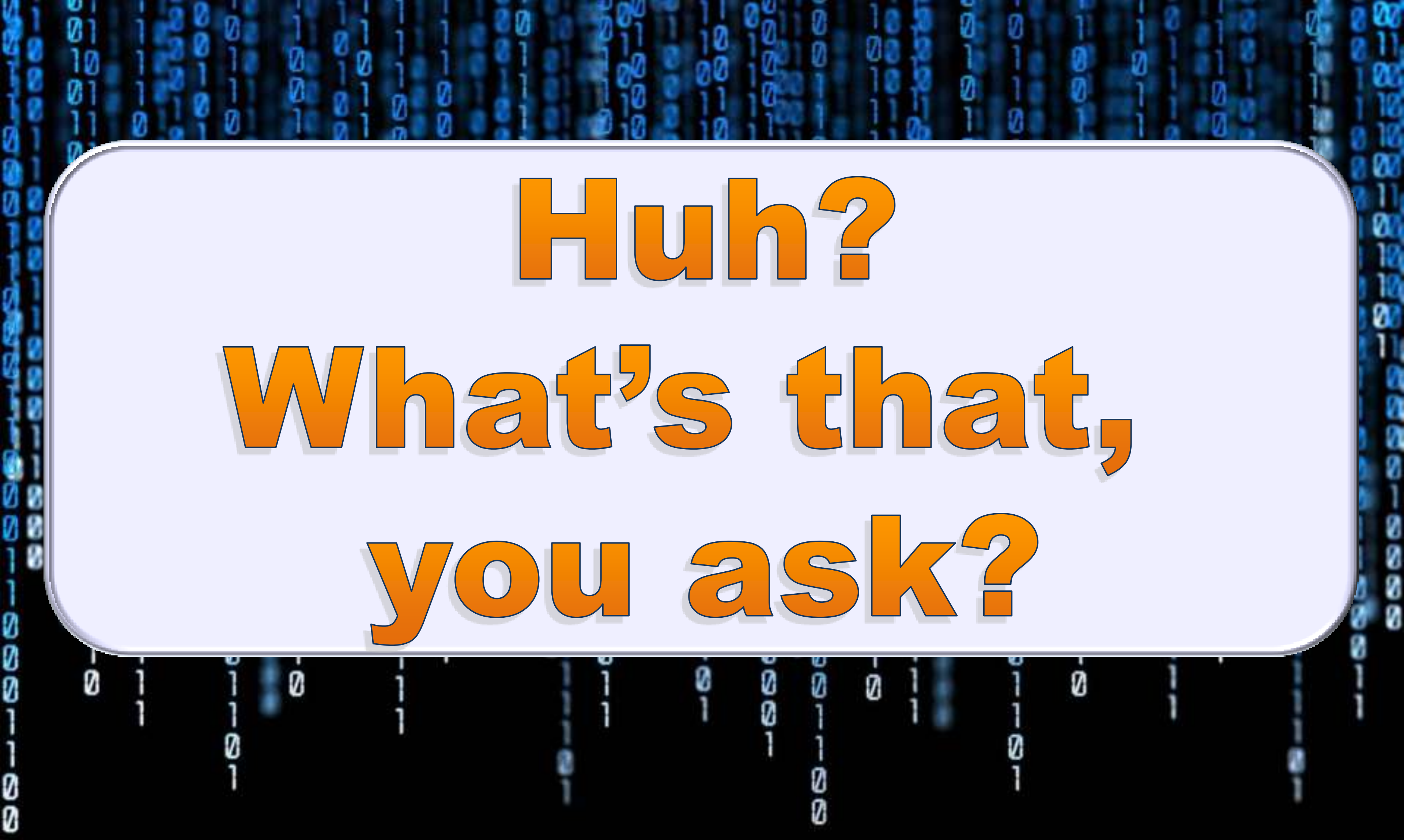
Test

Trend

Tune

T^4 APM™

A *“Test-Driven”* Application Performance Management Model

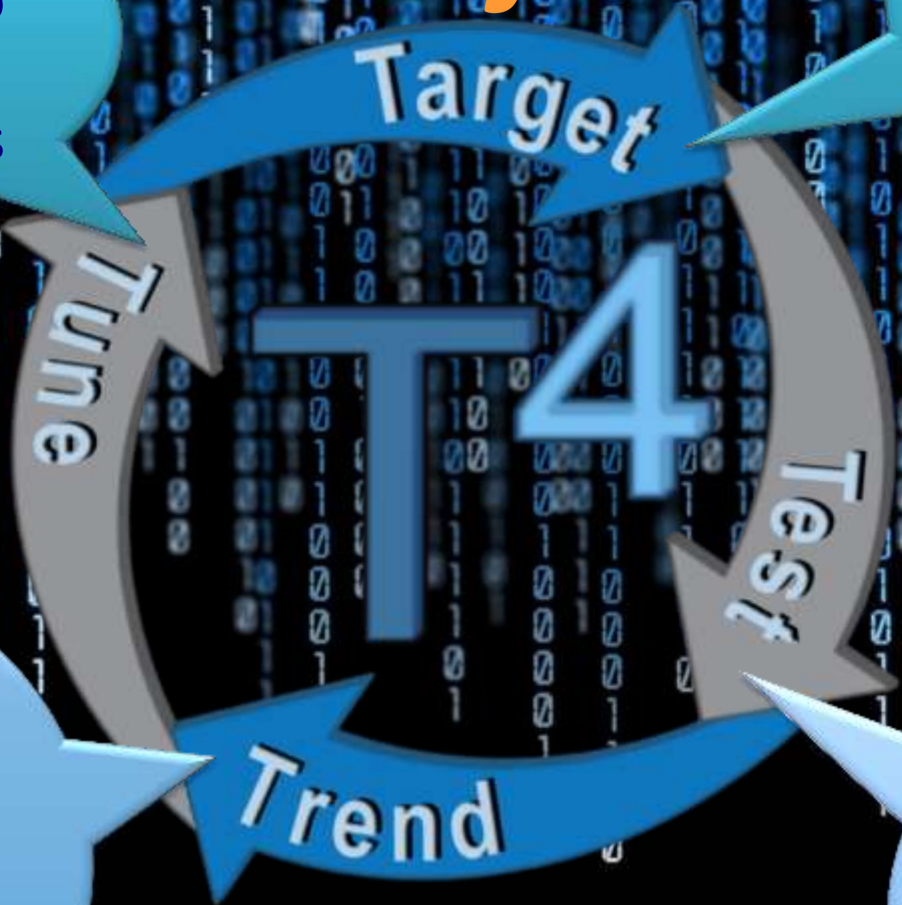


Huh?
What’s that, you ask?

The four T's stand for…

Proactive
Micro & Macro
Establish Goals
Update Targets

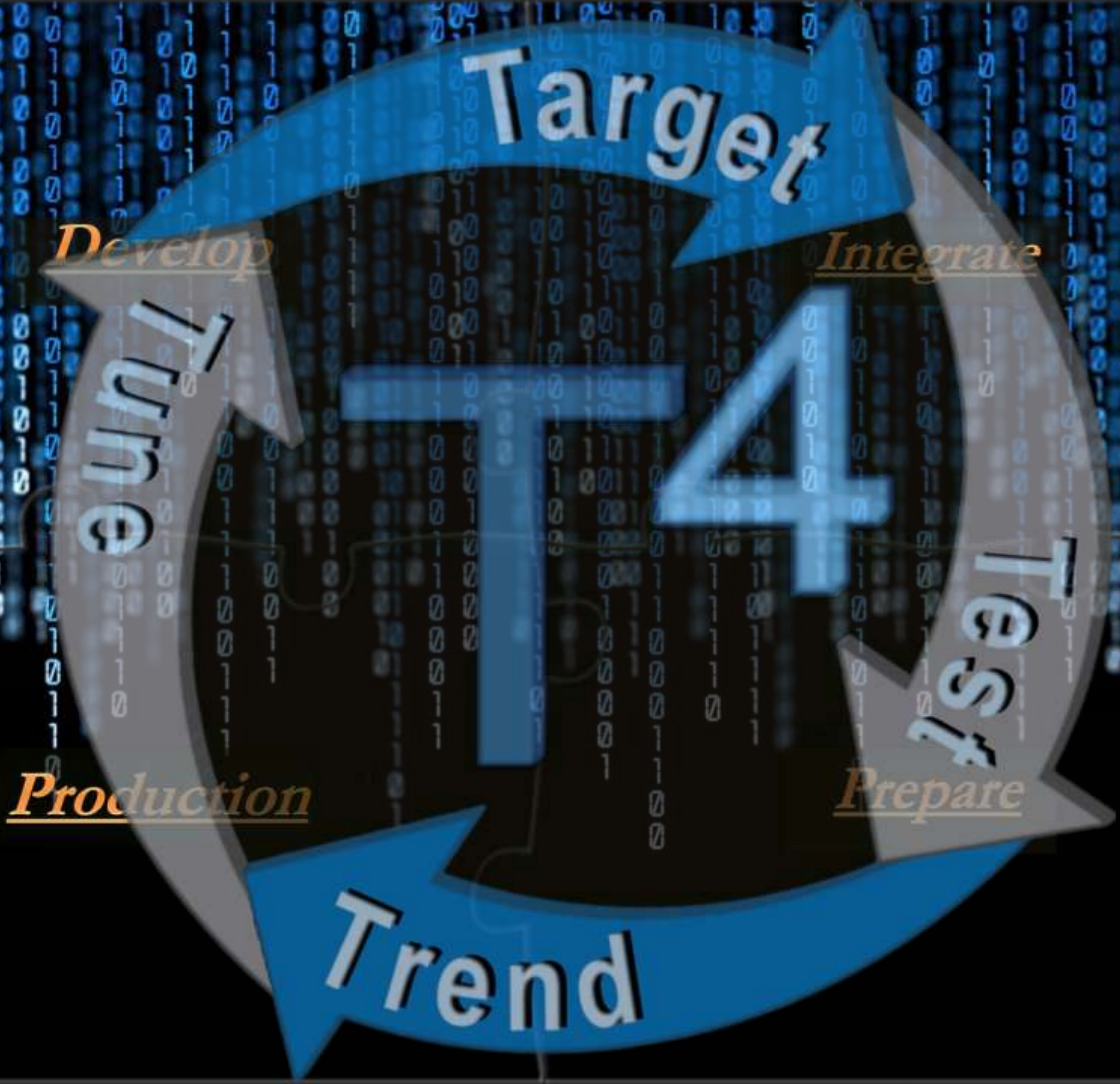
Units
Stories
Tiers
Resources
Goals



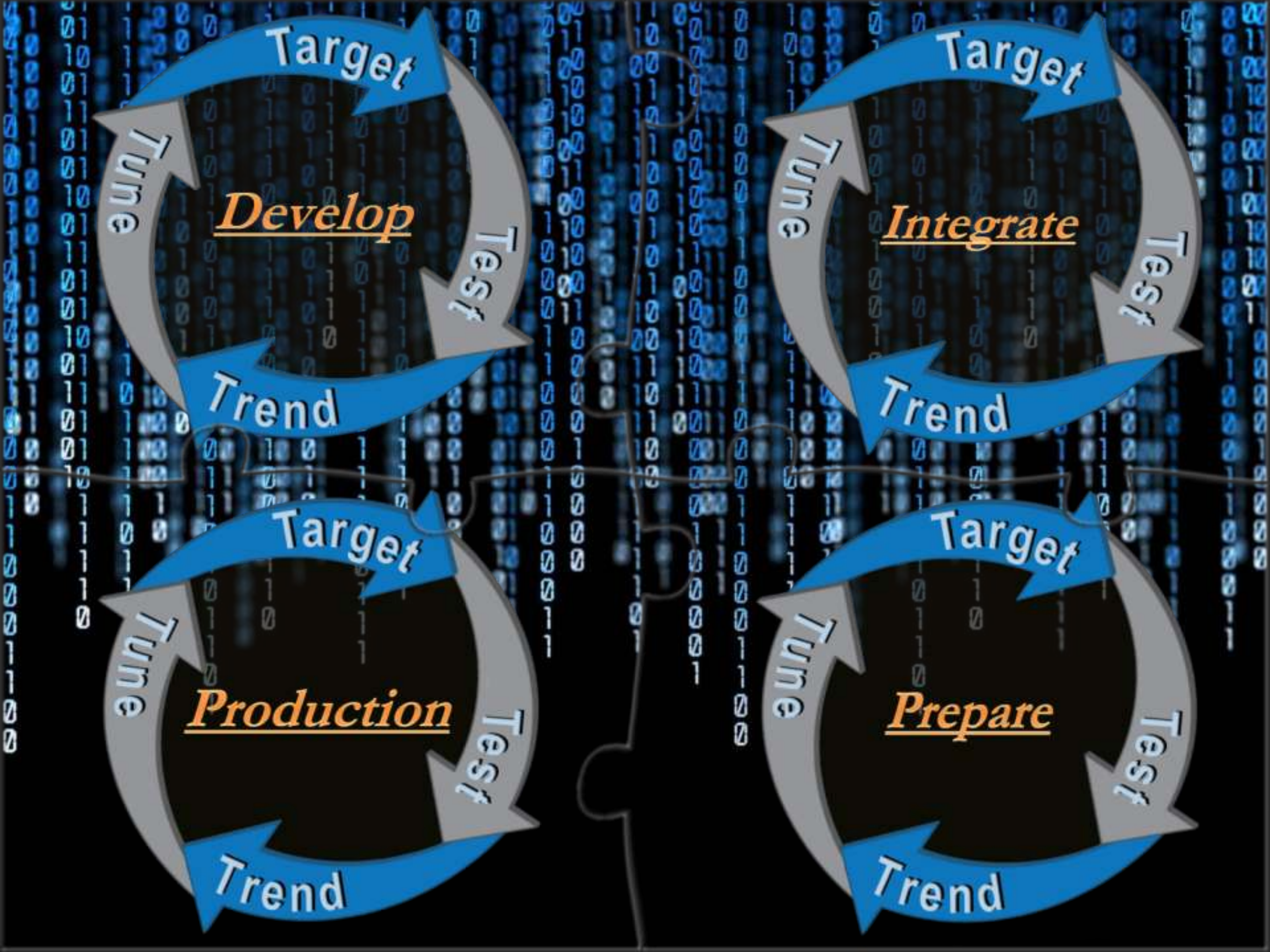
Dev & Prod
Times
Resources
Sizes
Frequencies
Dashboard!

Assess
Compare
Investigate
Accept
Answer

... and you can apply them like...


Target
Integrate
Test
Prepare
Trend
Production
Tune
Develop
T4

... or ...



Develop

Target

Test

Trend

Tune

Integrate

Target

Test

Trend

Tune

Production

Target

Test

Trend

Tune

Prepare

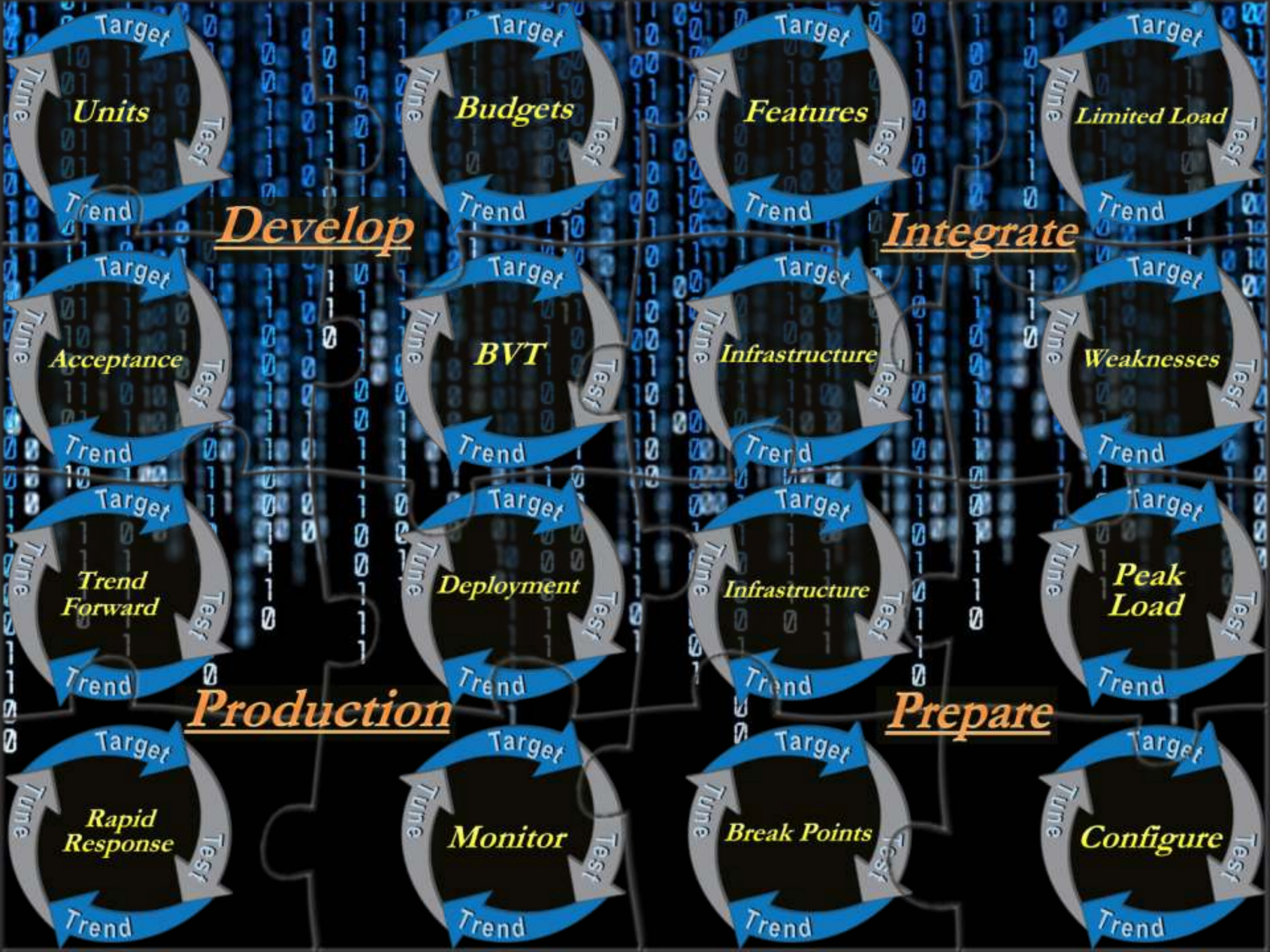
Target

Test

Trend

Tune

... or ...



Develop

Units

Budgets

Acceptance

BVT

Integrate

Features

Limited Load

Infrastructure

Weaknesses

Production

Trend Forward

Deployment

Rapid Response

Monitor

Prepare

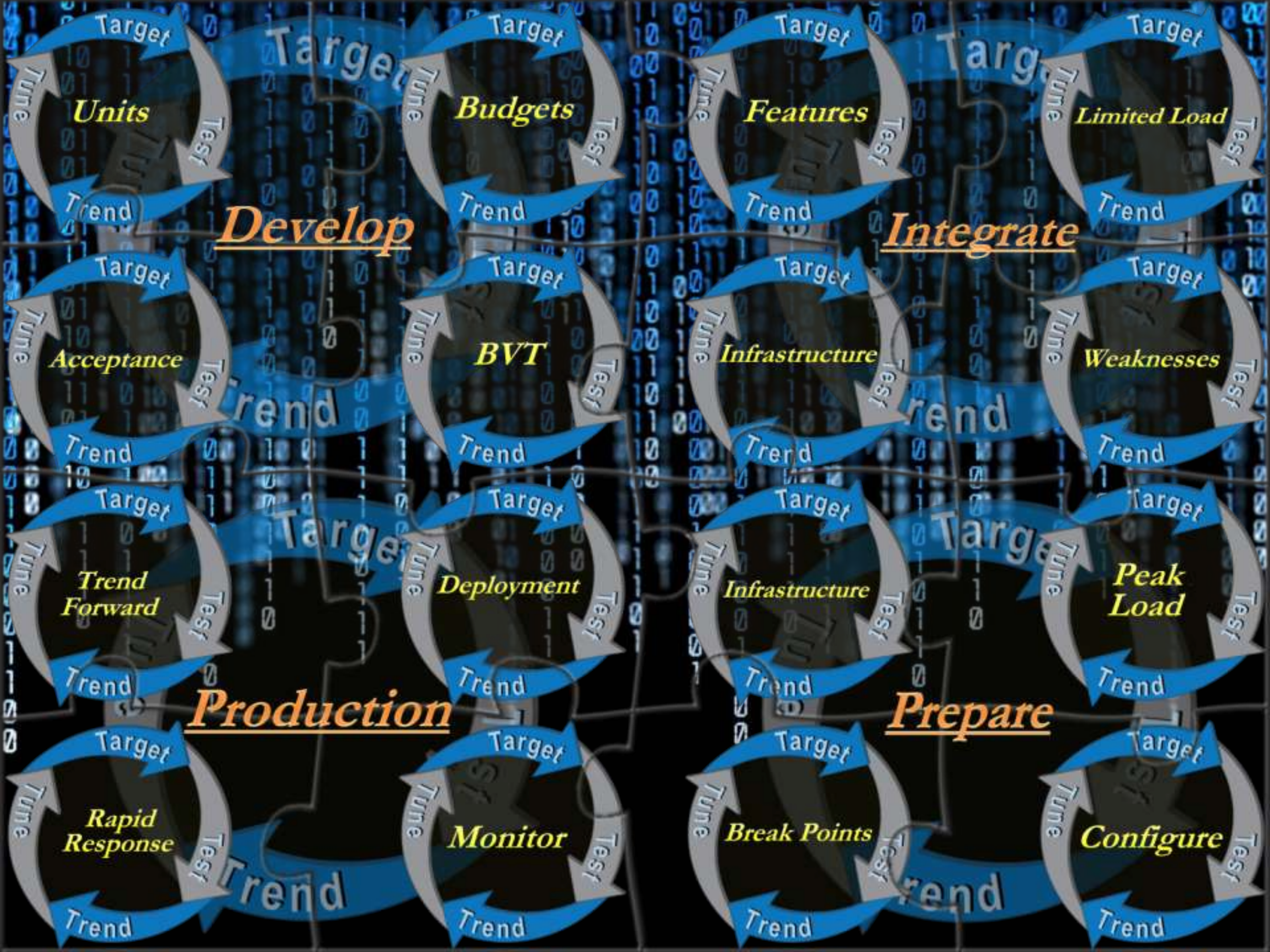
Infrastructure

Peak Load

Break Points

Configure

... but *and*
is better ...



Develop

Units

Budgets

Acceptance

BVT

Integrate

Features

Limited Load

Infrastructure

Weaknesses

Production

Trend Forward

Deployment

Rapid Response

Monitor

Prepare

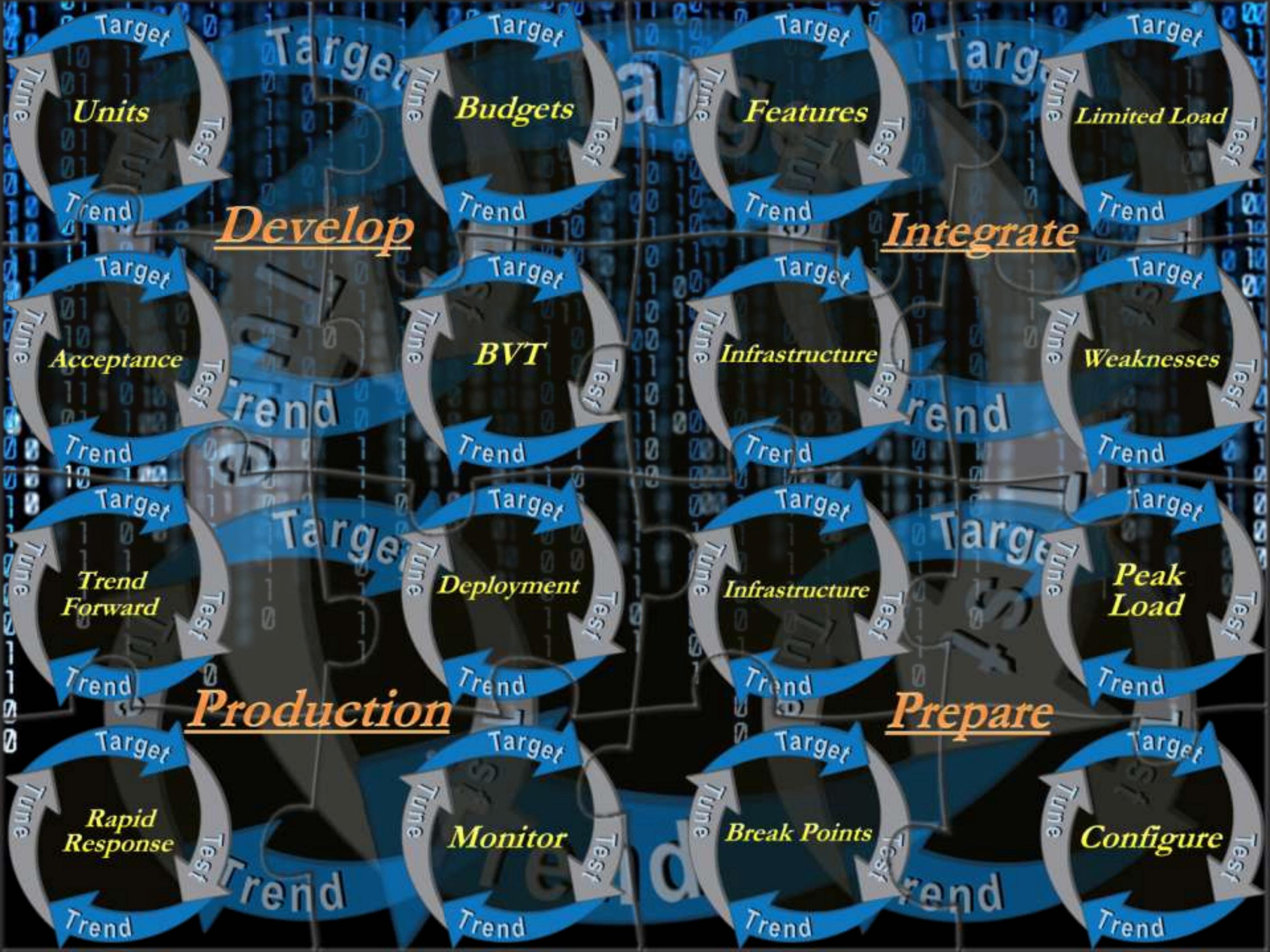
Infrastructure

Peak Load

Break Points

Configure

… and more *and*s are even better!



Develop

Units

Budgets

Acceptance

BVT

Integrate

Features

Limited Load

Infrastructure

Weaknesses

Production

Trend Forward

Deployment

Rapid Response

Monitor

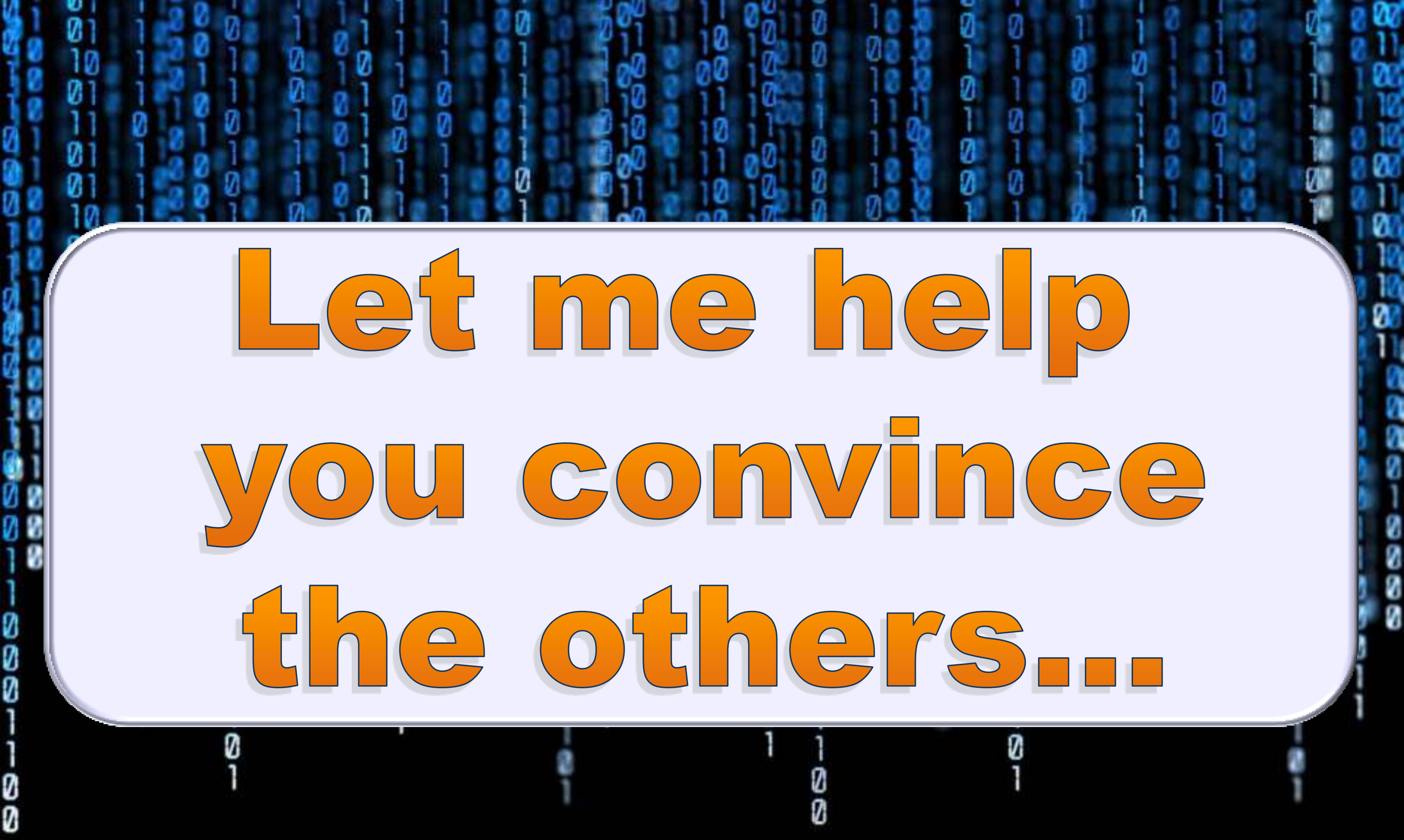
Prepare

Infrastructure

Peak Load

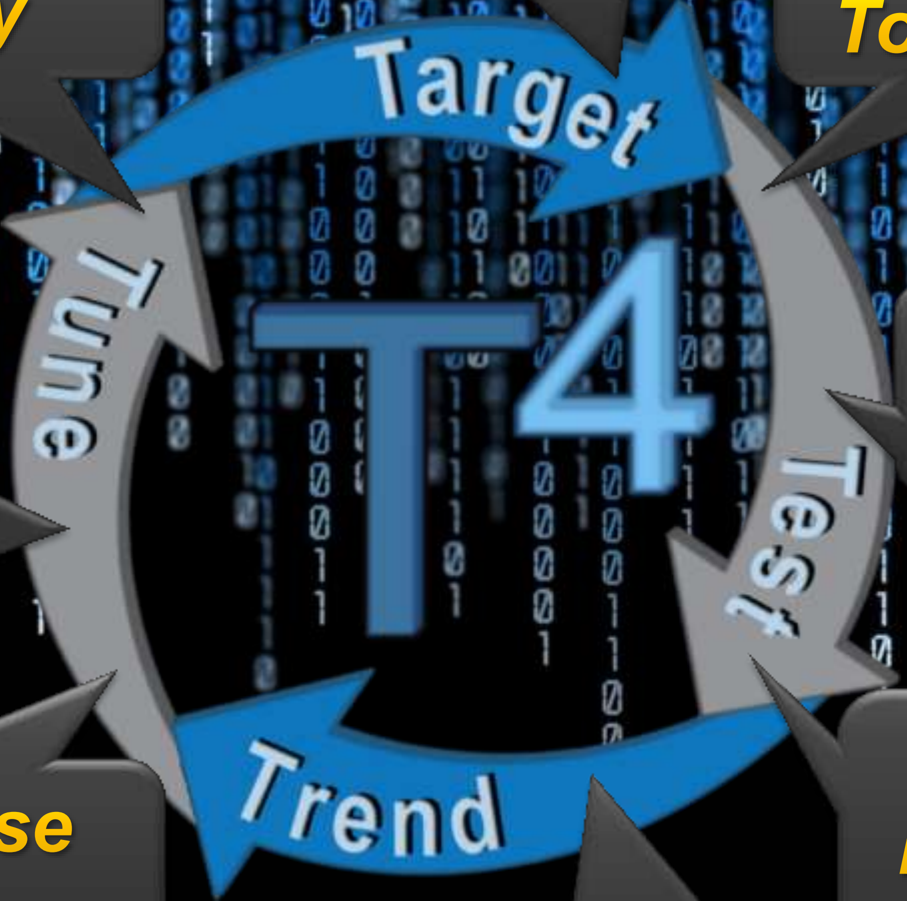
Break Points

Configure



Let me help you convince the others...

Value Proposition



A Few Minutes, Every Day

Minimizes Surprises

Methodology / Tool Agnostic

Adapts to Current Priority or Risk

Big Visible Dashboards

No Purchase Necessary

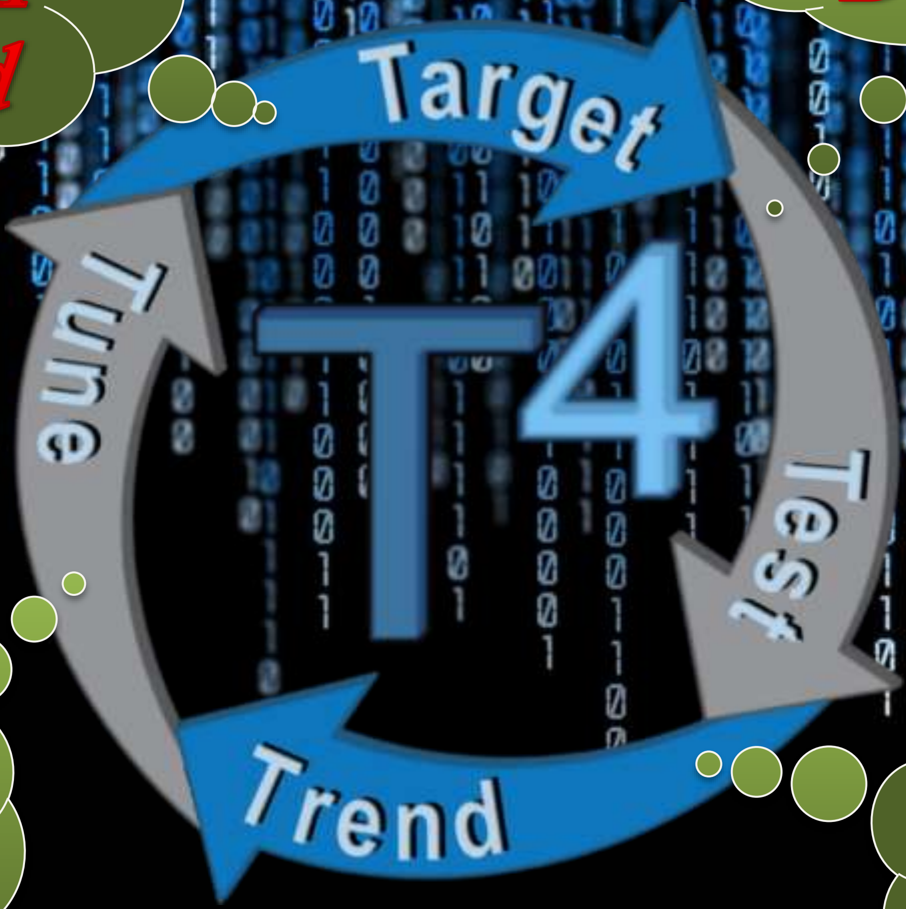
Collaborative

Simplifies Root-Cause Analysis

Challenges

Some Education Required

Demands Discipline



Culture Change is Hard

Several Cycles to Full Value

For best results I recommend bundling with…

Rapid Performance Testing

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

RPT 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?"*

RPT 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*

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*
   - *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*

Conceptual Approach

1. *Execute Strategy* *(~2.5 hrs)*
   - *Snapshots are your friends*
   - *Anecdotal is sufficient*
2. *Consolidate/Analyze Data* *(~30 min)*
   - *Identify patterns*
   - *Confirm patterns (time permitting)*
3. *Report Results* *(~20 min)*
   - *Answer(s)*
   - *Time/Effort to answer(s)*
   - *Follow-on questions of interest*

... and with...

Heuristic Load Testing

Load Testing Principles

Context Project context is central to successful 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.

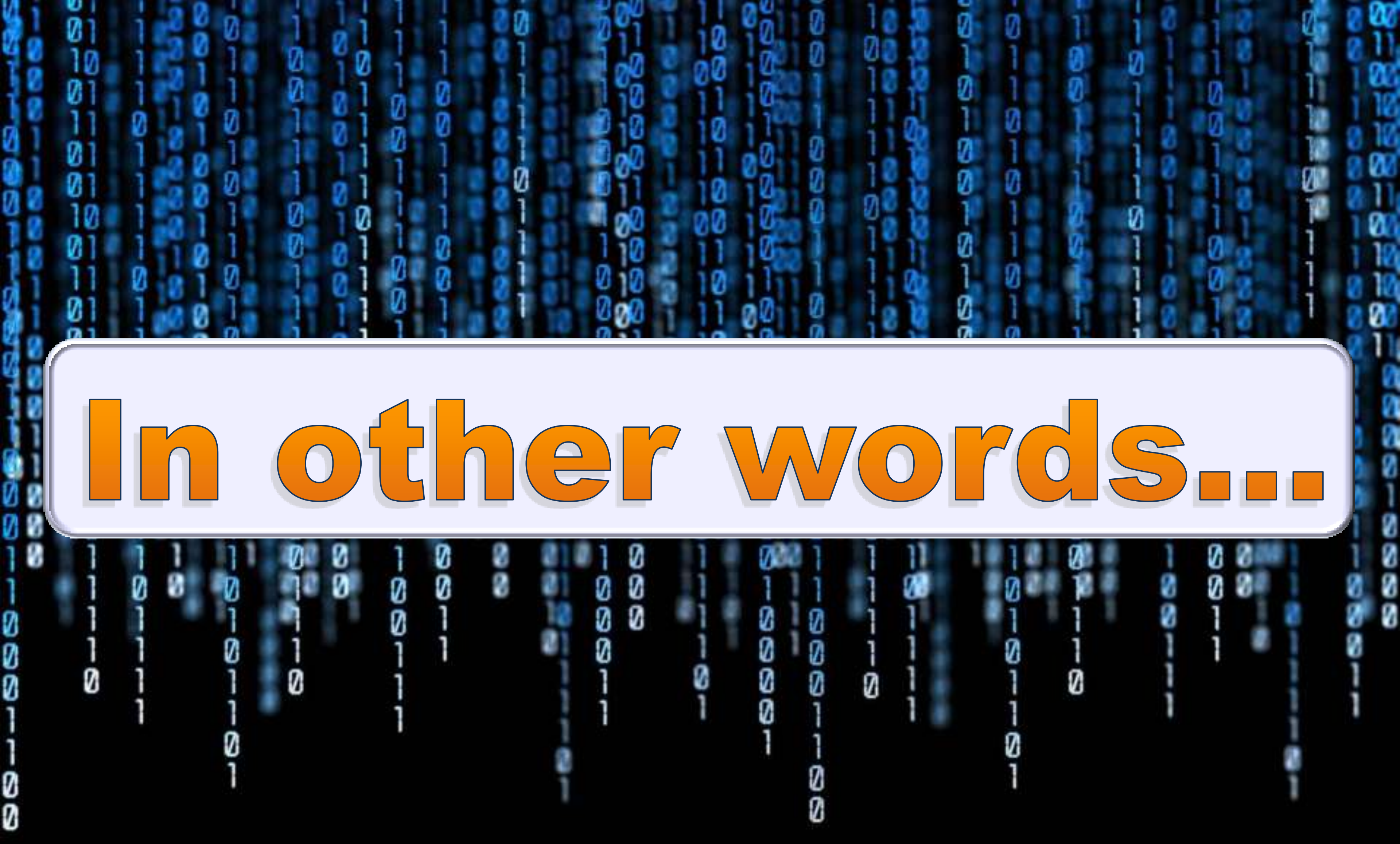
Script Script the tests as designed.

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.



In other words…

T^{4}APM™ + RPT + HLT =>

A Practical & Holistic Approach for Achieving Fast & Scalable Systems

Footnote:

These slides have been posted at.

Questions?

Target

Test

Trend

Tune

T4

APM™

Preventing Poor Performance with

a *little* work…

***every* day…**

from *every* one.

http://wiki.perftestplus.com/dokuwiki/doku.php?id=atd:atd2012

Contact Info

Scott Barber

Chief Technologist

PerfTestPlus, Inc

E-mail:

sbarber@perftestplus.com

Web Site:

www.PerfTestPlus.com

Blog:

scott-barber.blogspot.com

Twitter:

@sbarber

Review & Questions

Did we learn anything?

Please evaluate my presentation by using the evaluation booklets which you can find in your conference bag.

Thank you!