Performance Testing:

Throughout the Application Life-Cycle

Created for:

By:

Scott Barber

Chief Technologist

PerfTestPlus, Inc.

Performance Testing:

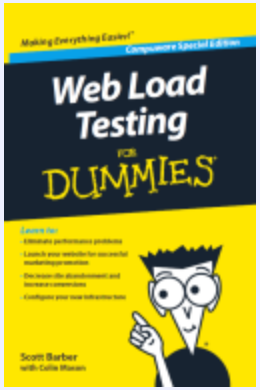
Throughout the Application Life-Cycle

Scott Barber

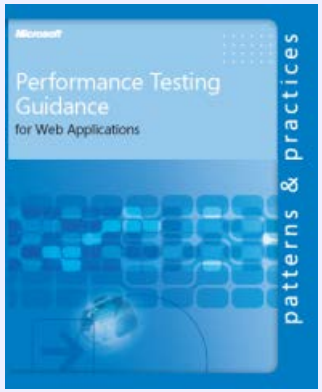
Chief Technologist, PerfTestPlus, Inc.
sbarber@perftestplus.com
www.perftestplus.com
@sbarber

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

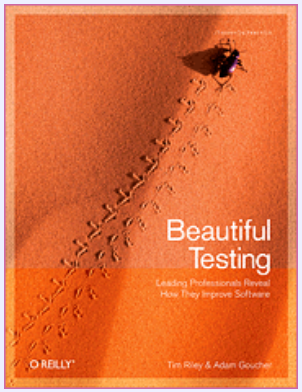
Author:

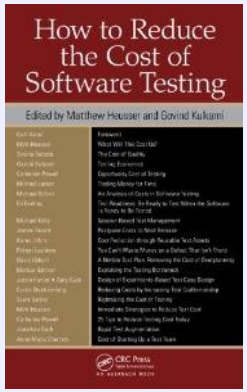


Co-Author:



Contributing Author:





Books: www.perftestplus.com/pubs

About me: about.me/scott.barber

Notes

- This keynote 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 keynote is based on personal experience in *some* context (but not yours)
- Adapt concepts to your situation/context
- Do network with others who are here (I bet they have great ideas too)
- Follow–up with me (what works for you… or doesn't)
- I *like* questions!

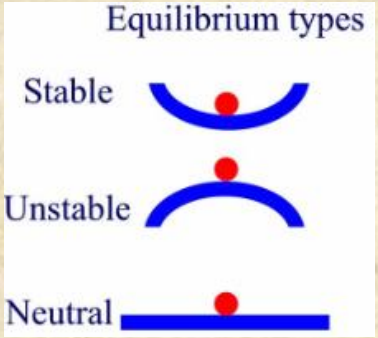
<Background>

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

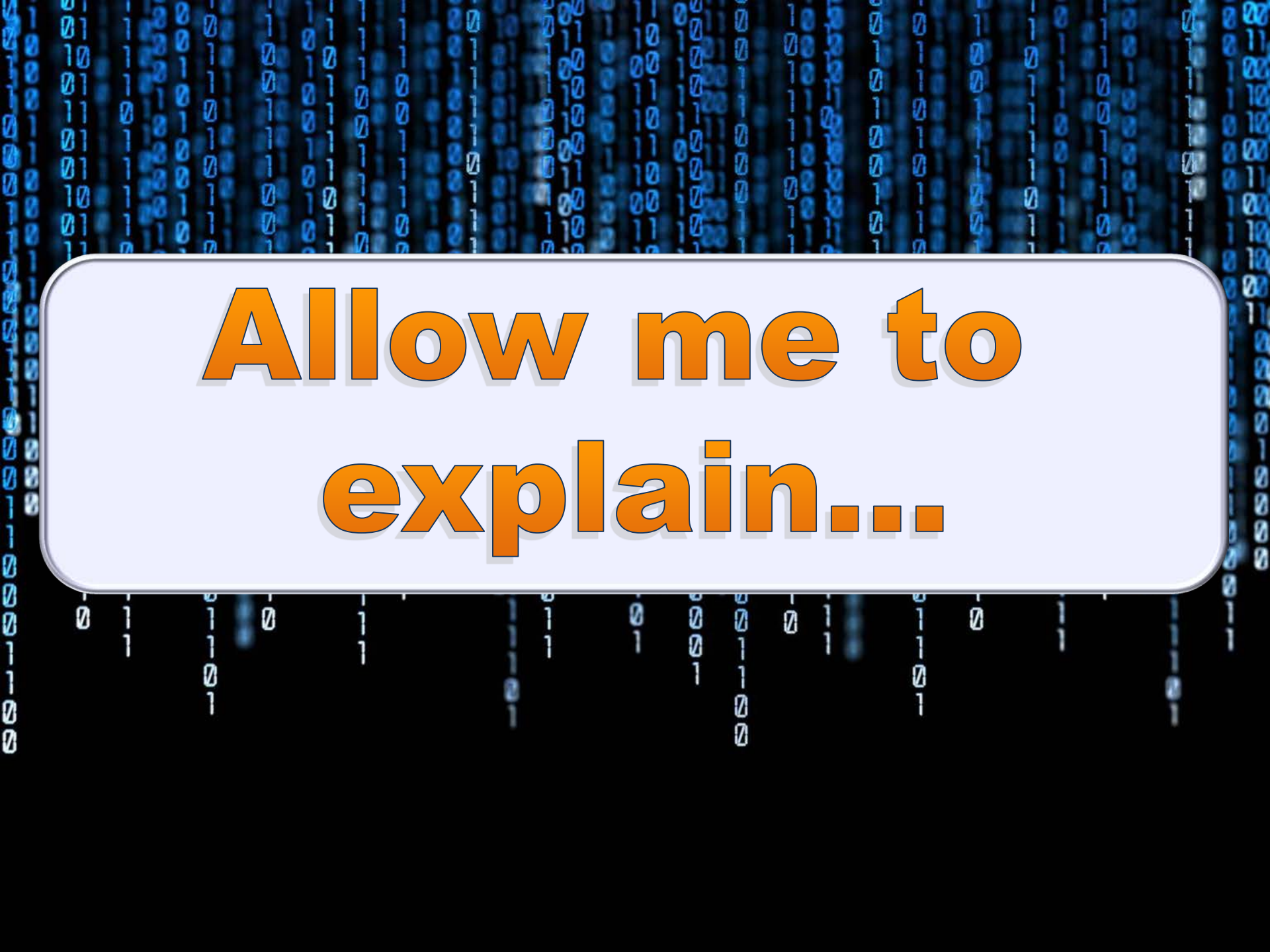
</Background>

Who is Responsible?

Everyone

Where Does it Fit?

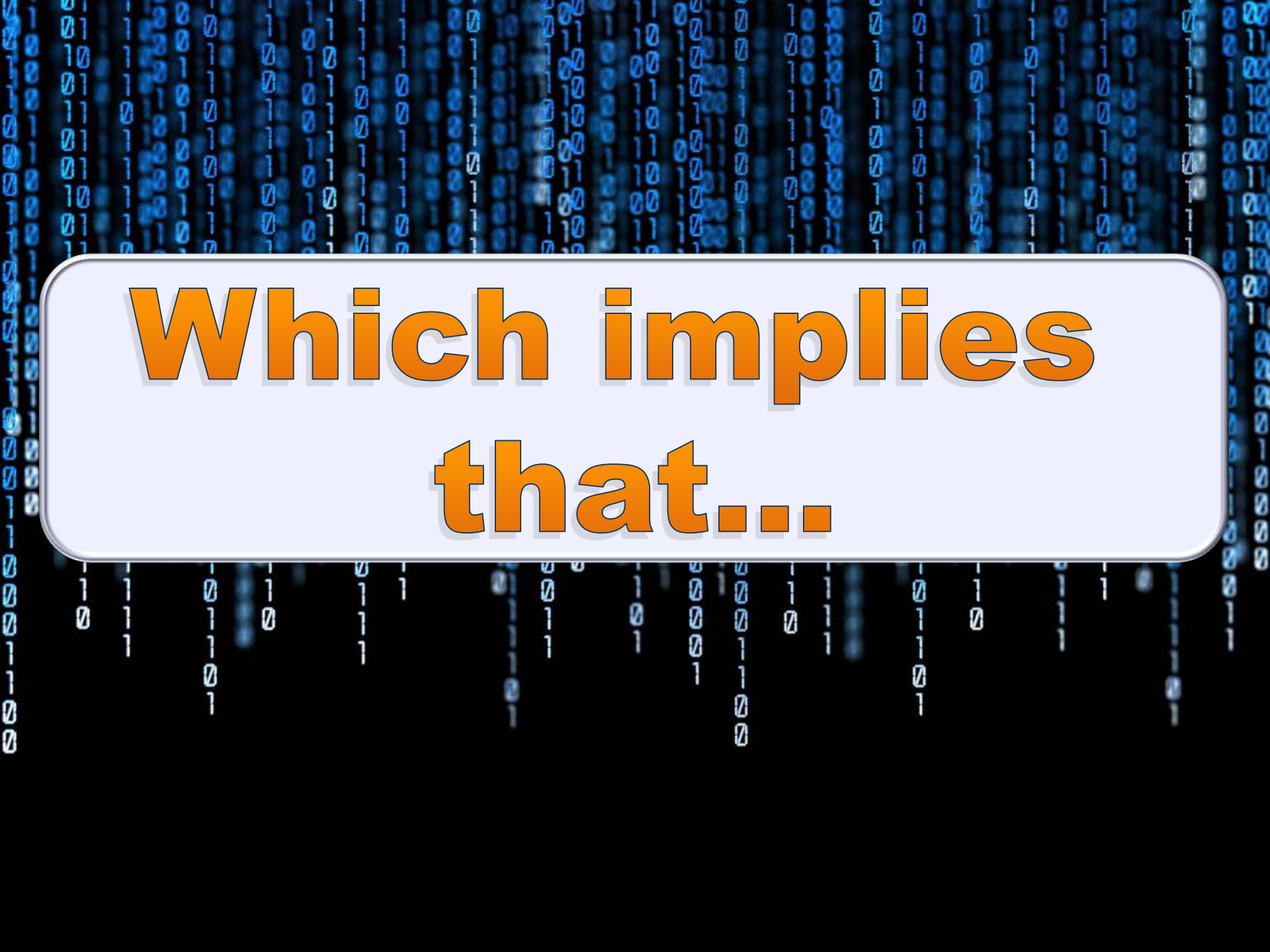
Everywhere


Allow me to
explain...

To Prevent Poor Performance…

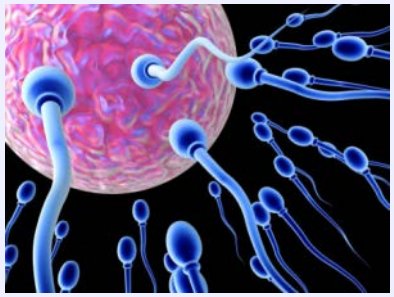
… don’t just react when it happens.

An Ounce of Prevention…


Which implies
that...

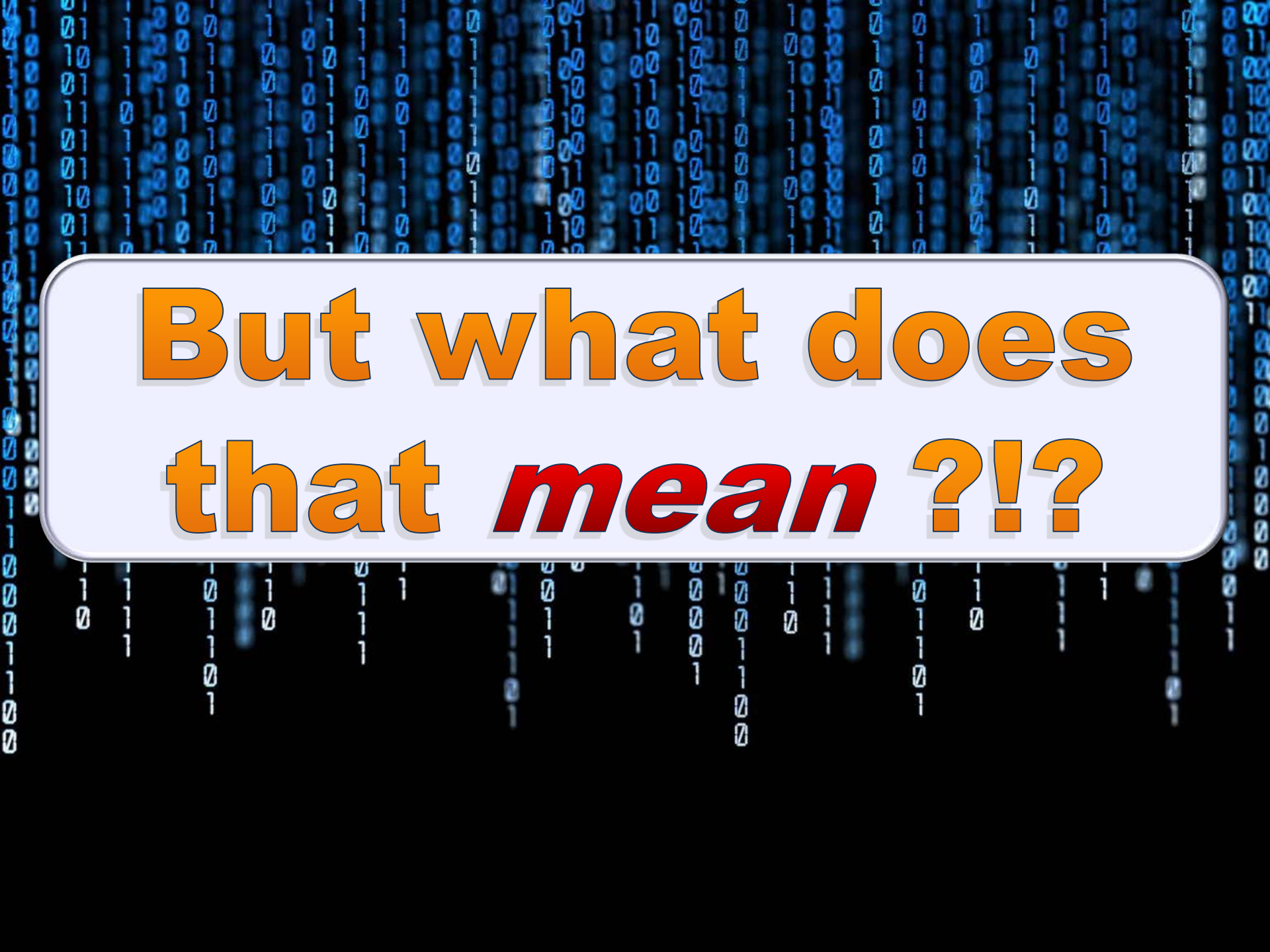
The Performance Lifecycle is:

Conception to Headstone

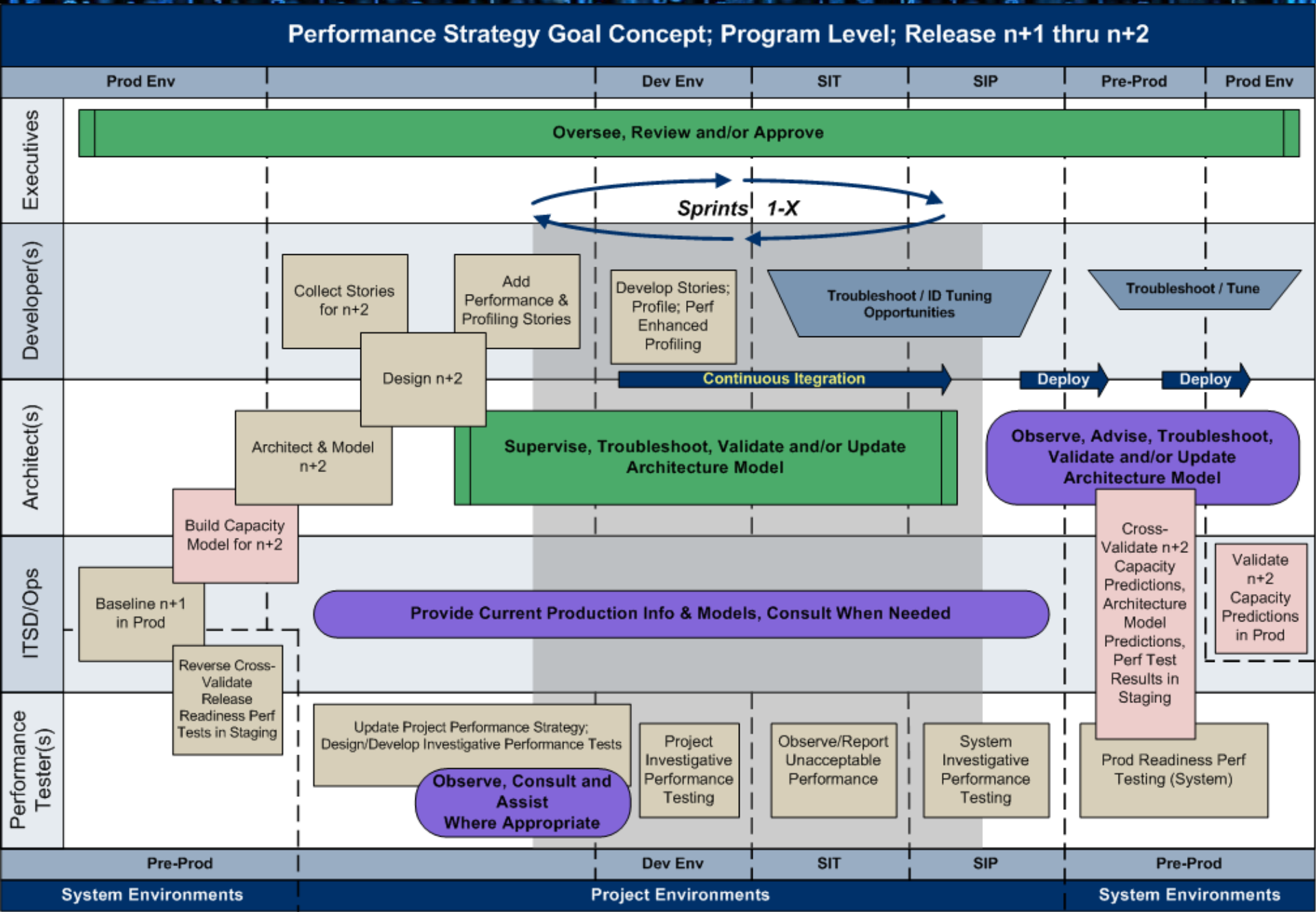


Not

Cradle to Grave


But what does
that mean ?!?

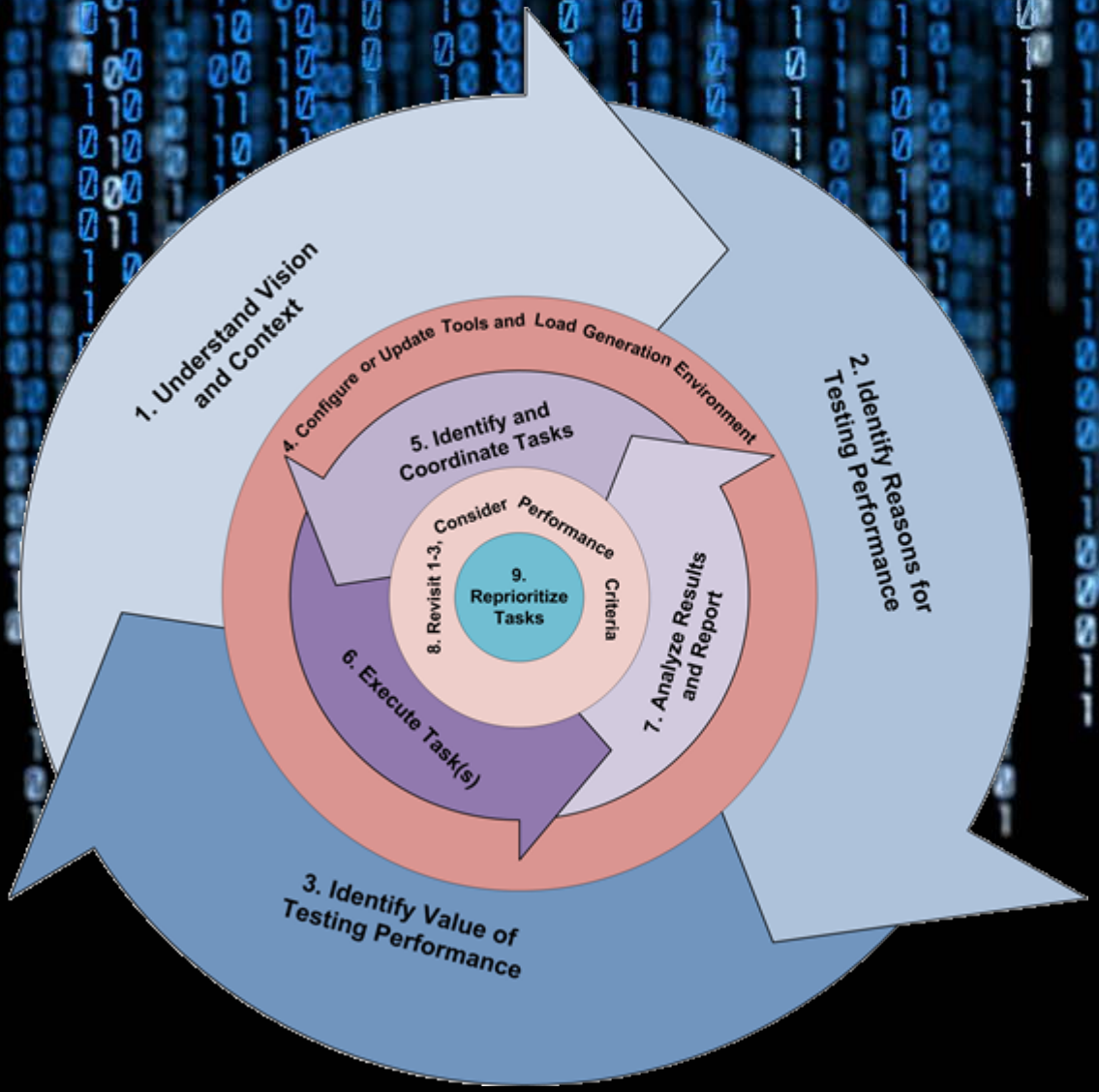
…Do you like Swim Lanes?



...Or maybe Circles?

Agile Performance Testing Activities

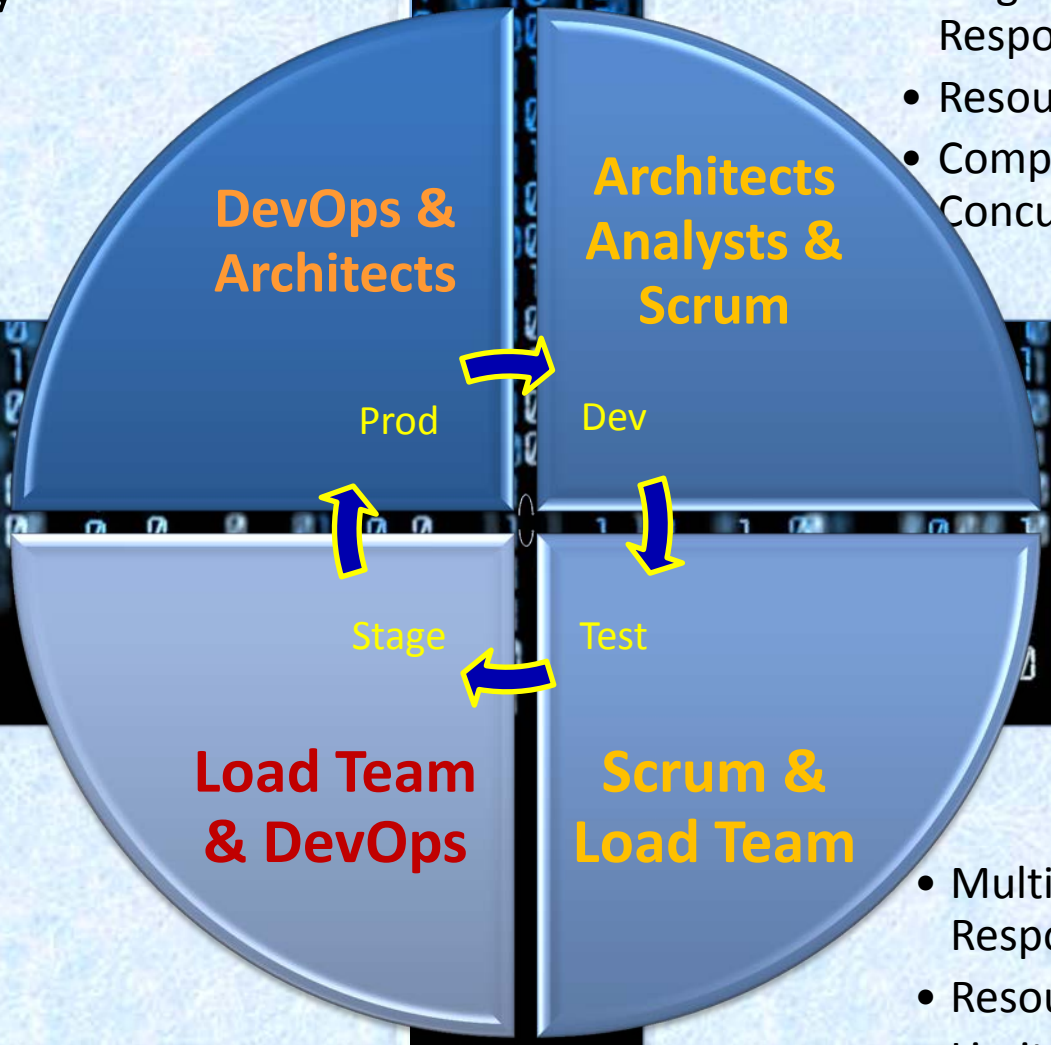
1. Understand Project Vision and Context
2. Identify Reasons for Testing Performance
3. Identify Value of Testing Performance
4. Configure or Update Tools and Load Generation Environment
5. Identify and Coordinate Tasks
6. Execute Task(s)
7. Analyze Results and Report
8. Revisit 1-3, Consider Performance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



...Or Annotated Pie Charts?

- Current Capacity
- Projections
- Scalability Plan
- Reliability

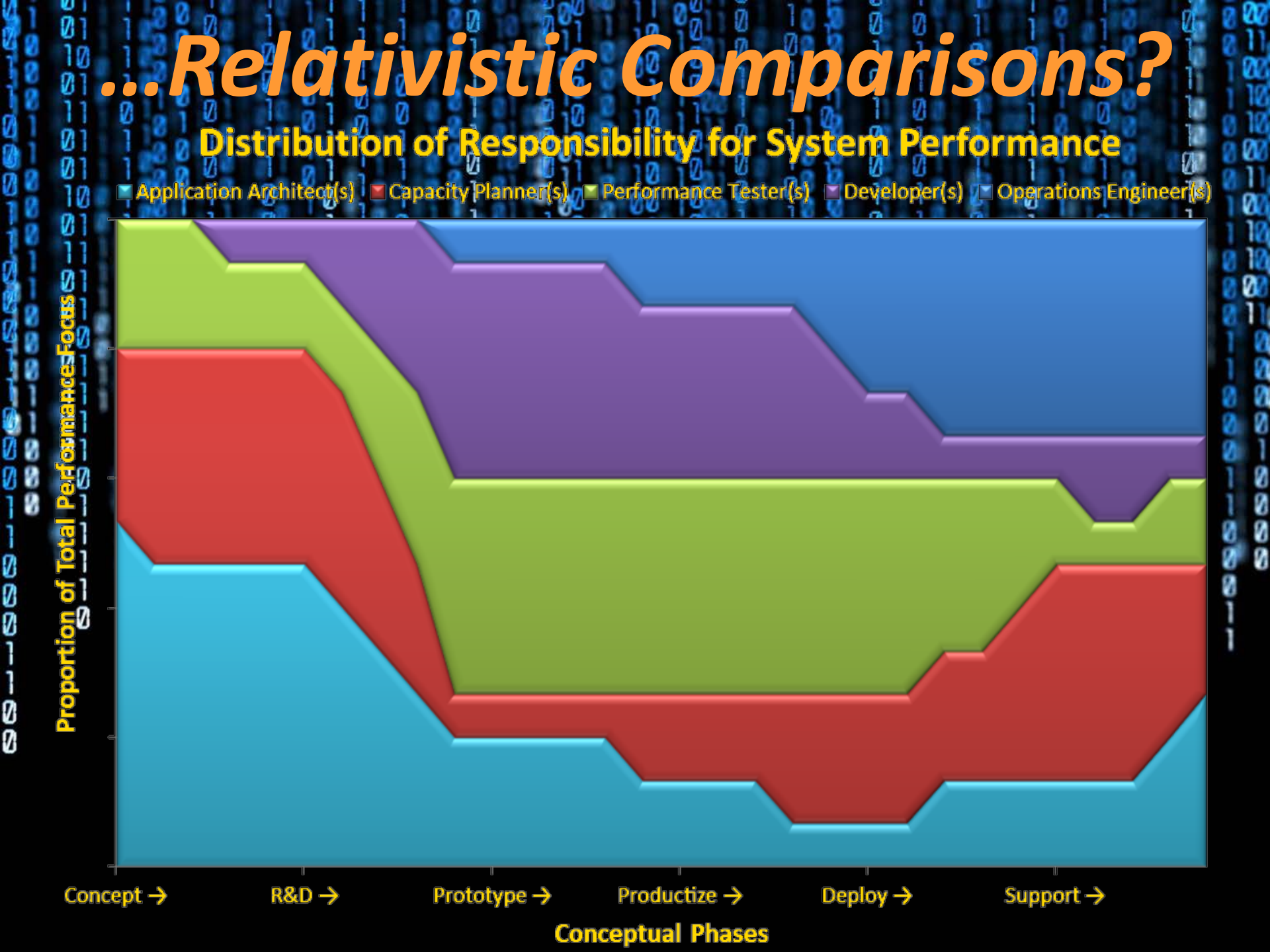
- Single User Responsiveness
- Resource Management
- Component-Level Concurrency



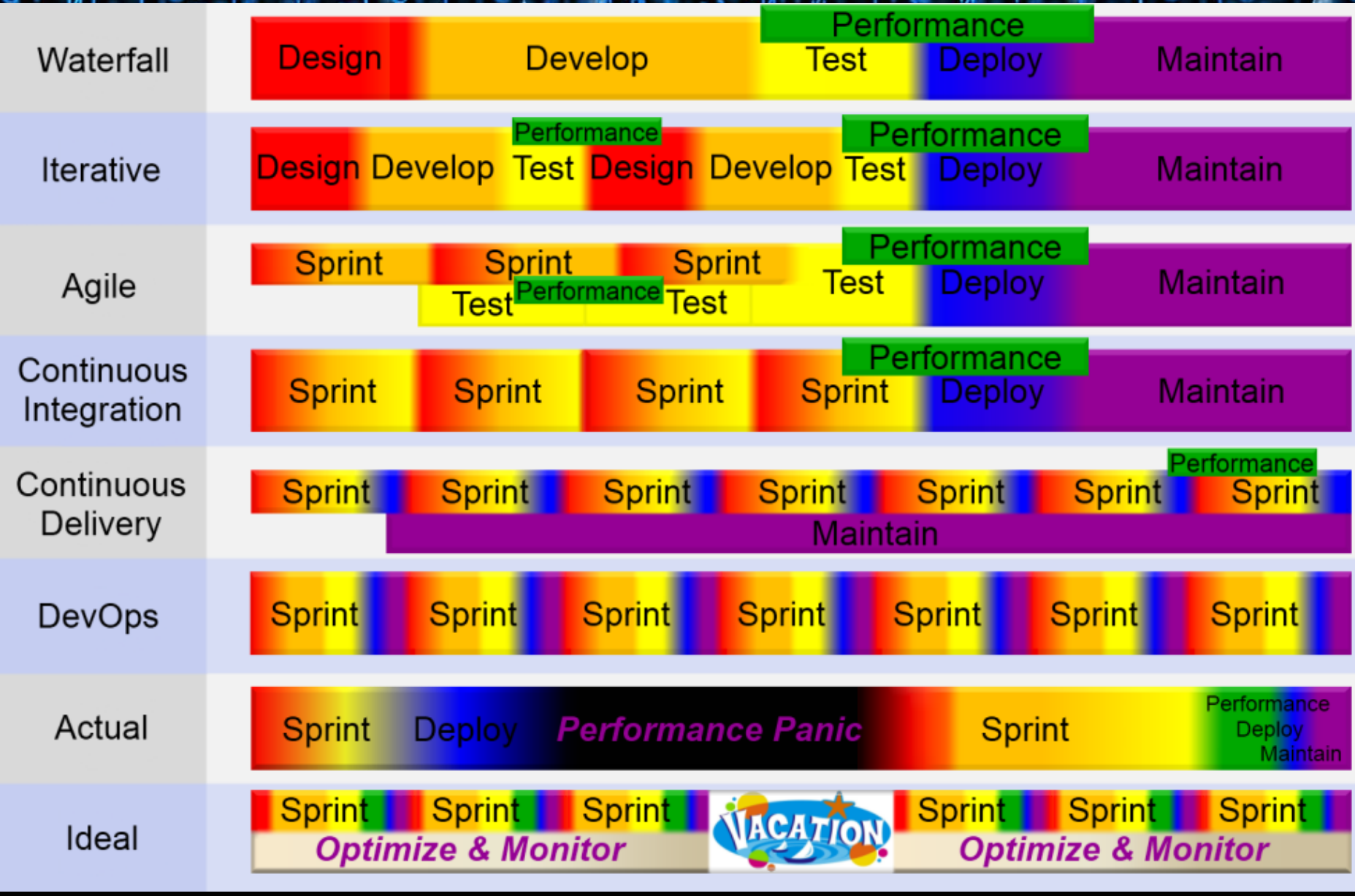
- Prepare for Prod
- Tune/Optimize
- Predict
- Early Warning Identification

- Multi-User Responsiveness
- Resource Trending
- Limit Identification

...Relativistic Comparisons?



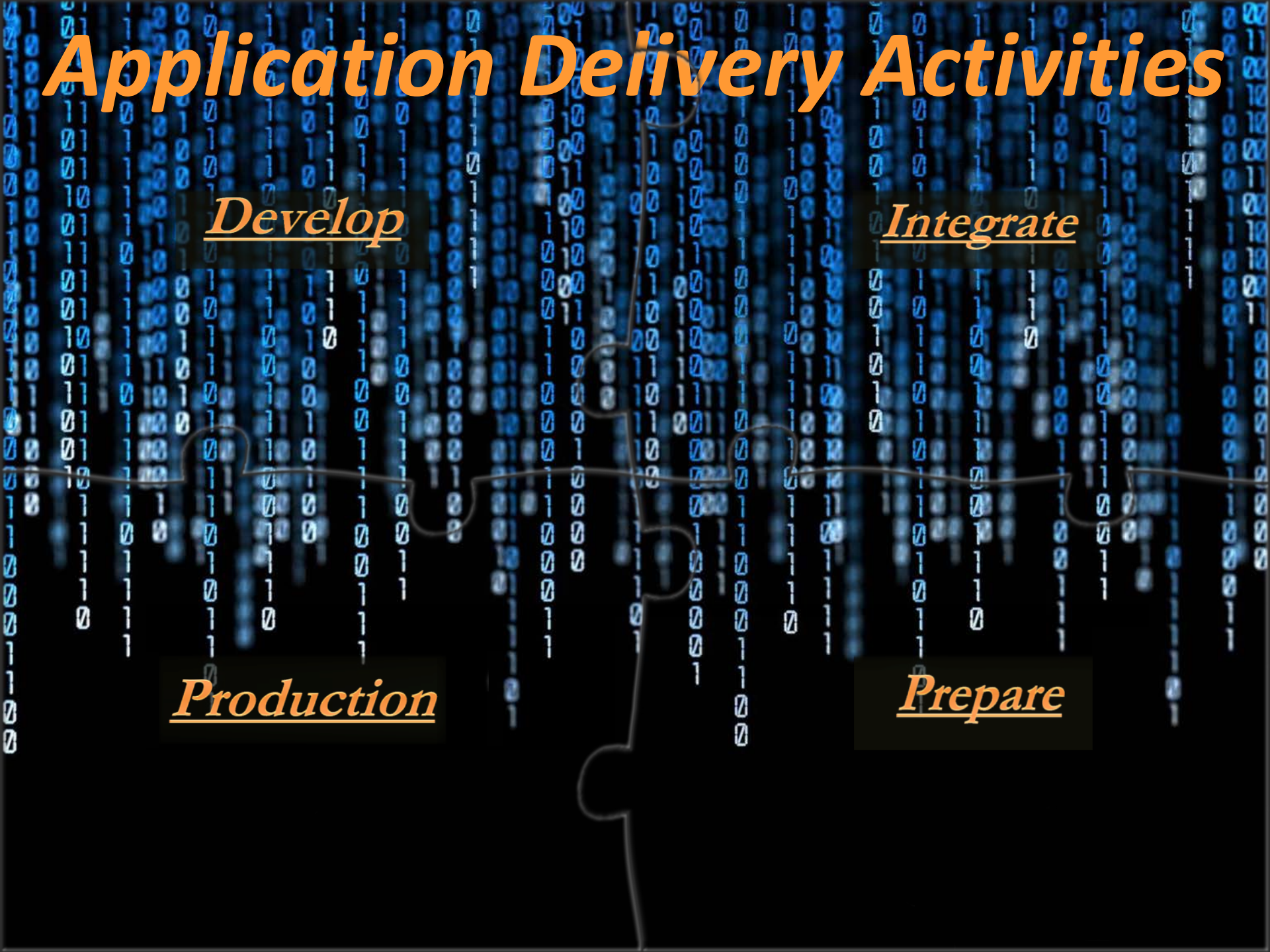
...How about Colors?

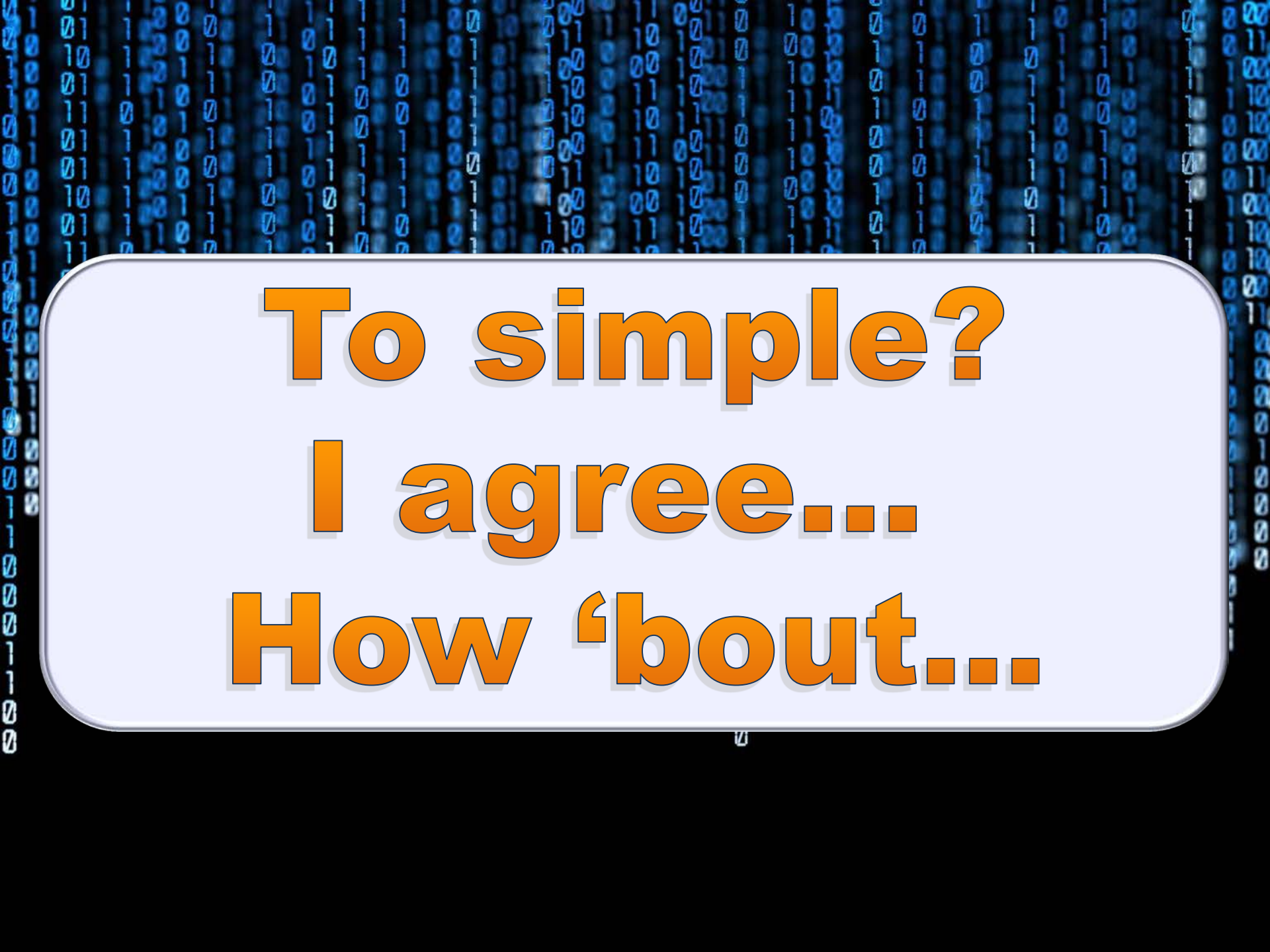


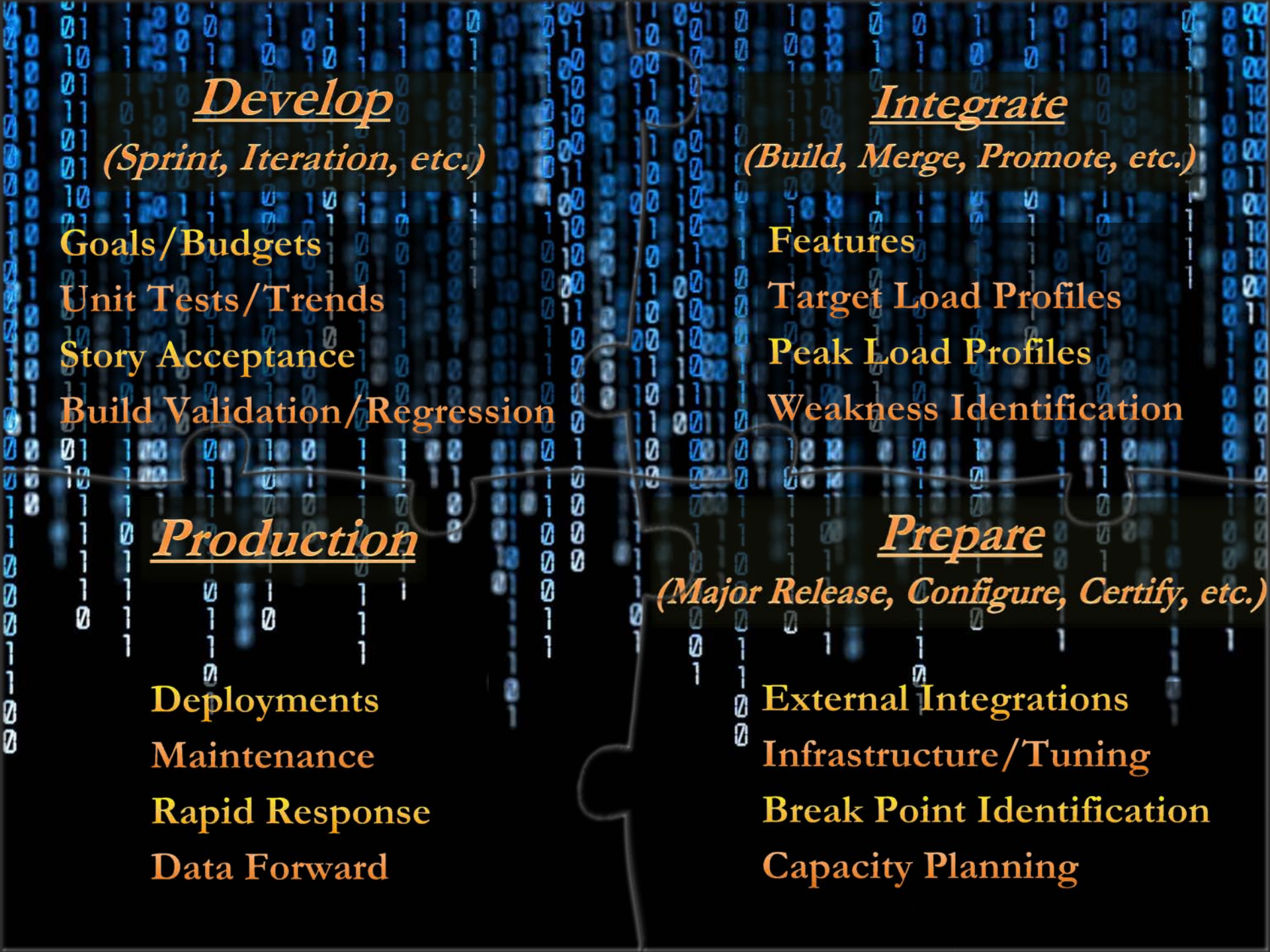

I’ve tried all
of those…

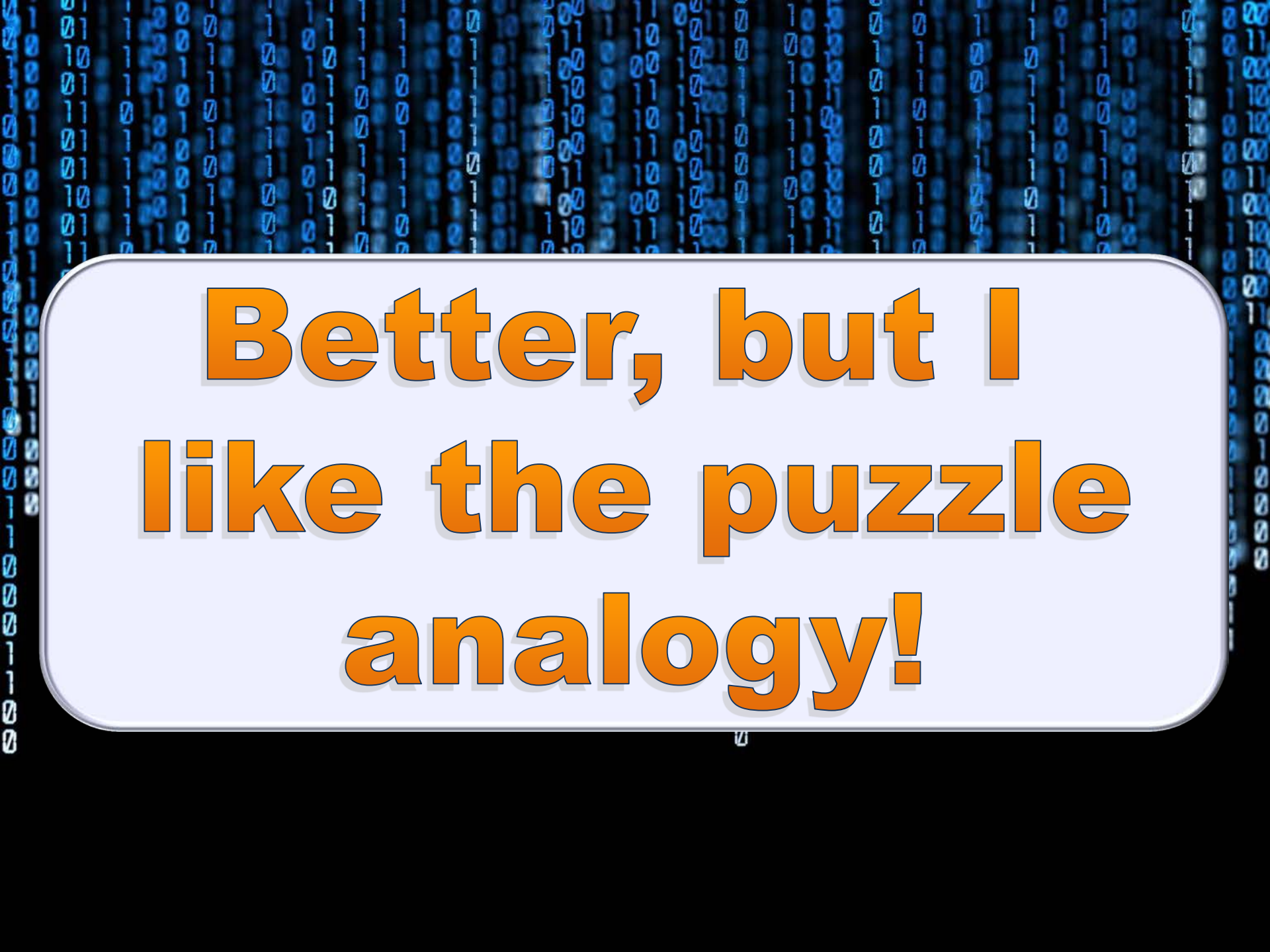

... But now
I prefer...

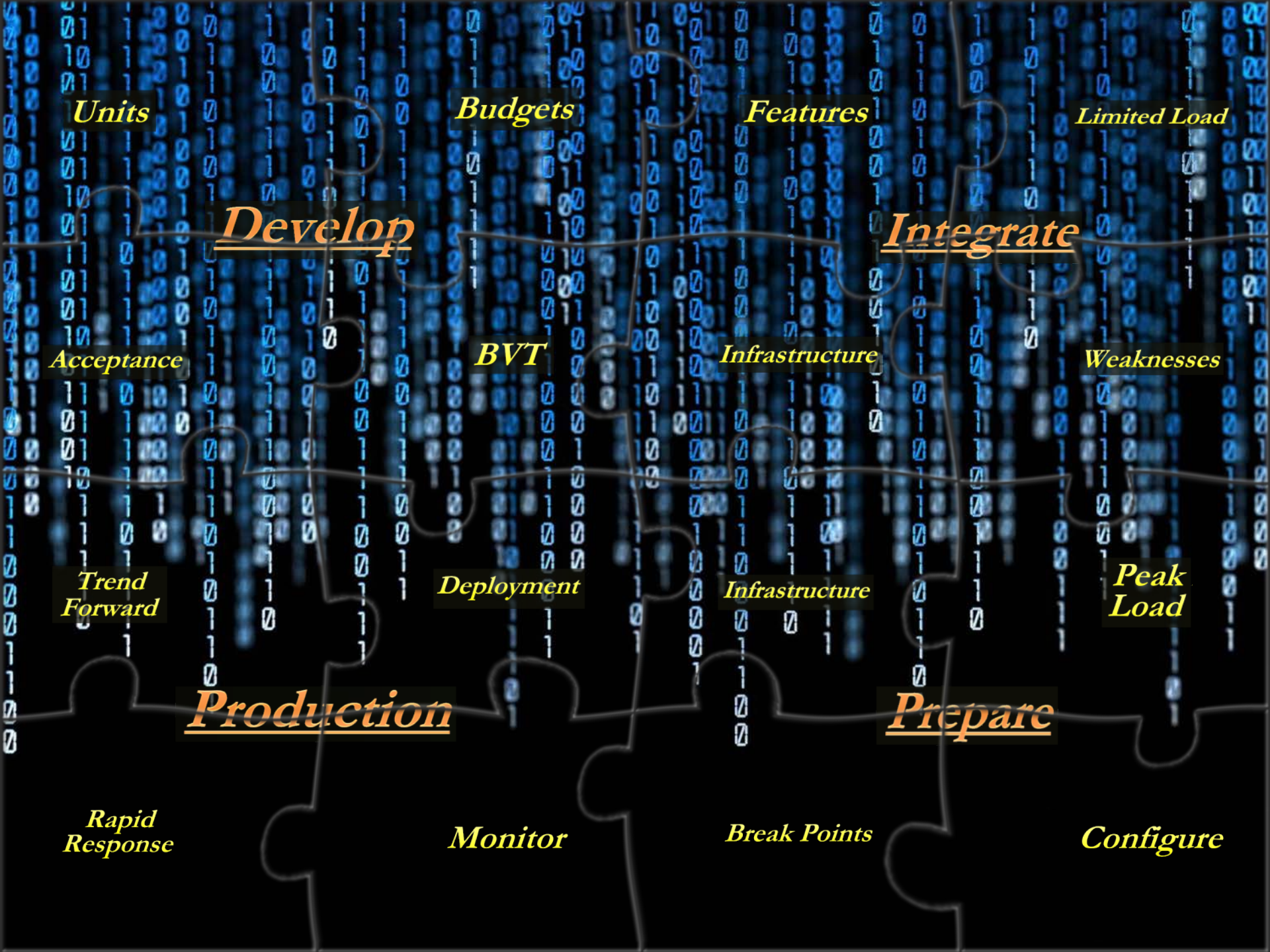

...the
Performance
Puzzle.


Application Delivery Activities
Develop
Integrate
Production
Prepare

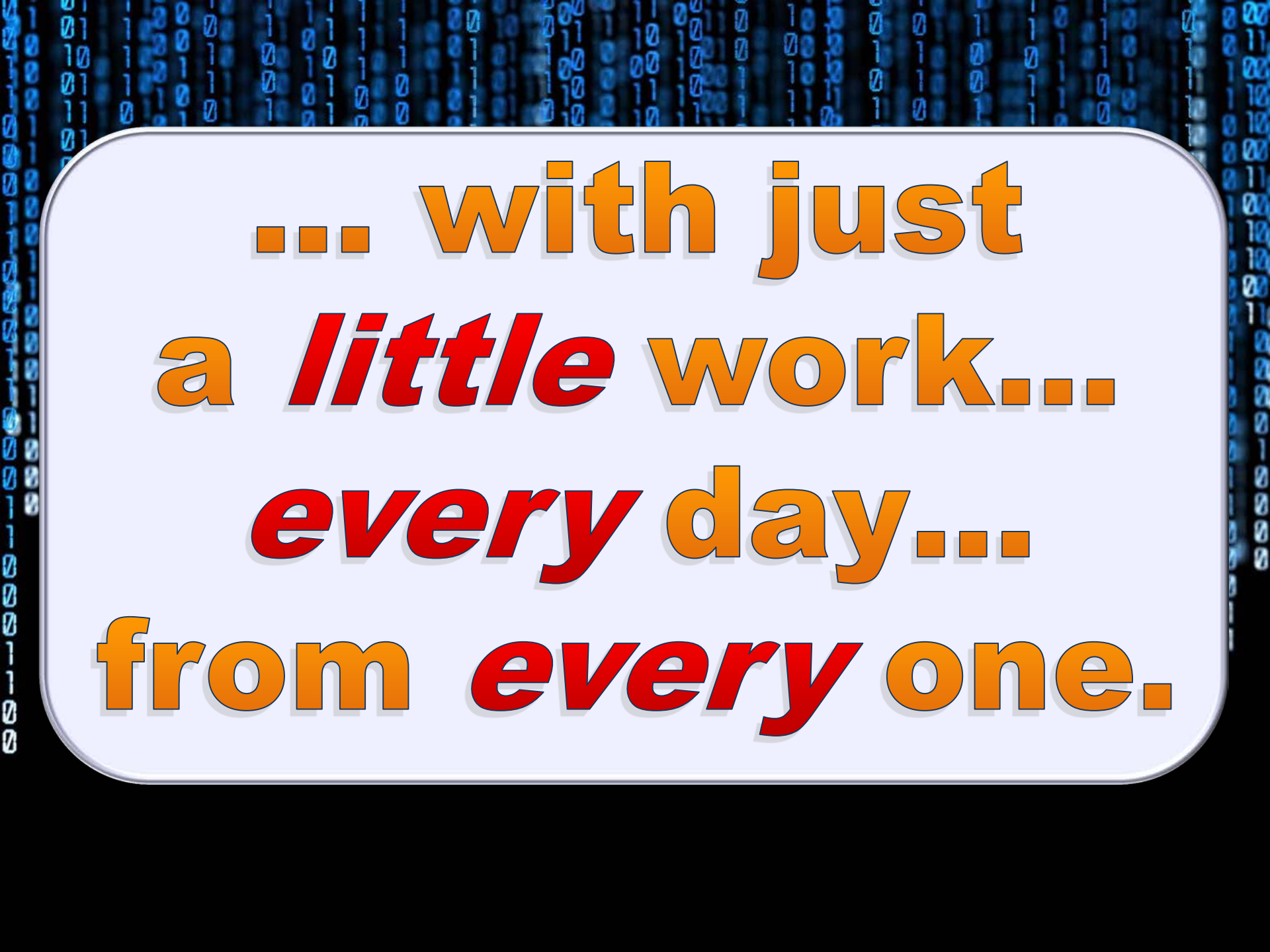

To simple?
I agree…
How ‘bout…


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!


Units
Budgets
Features
Limited Load
Develop
Integrate
Acceptance
BVT
Infrastructure
Weaknesses
Trend Forward
Deployment
Infrastructure
Peak Load
Production
Prepare
Rapid Response
Monitor
Break Points
Configure

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


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

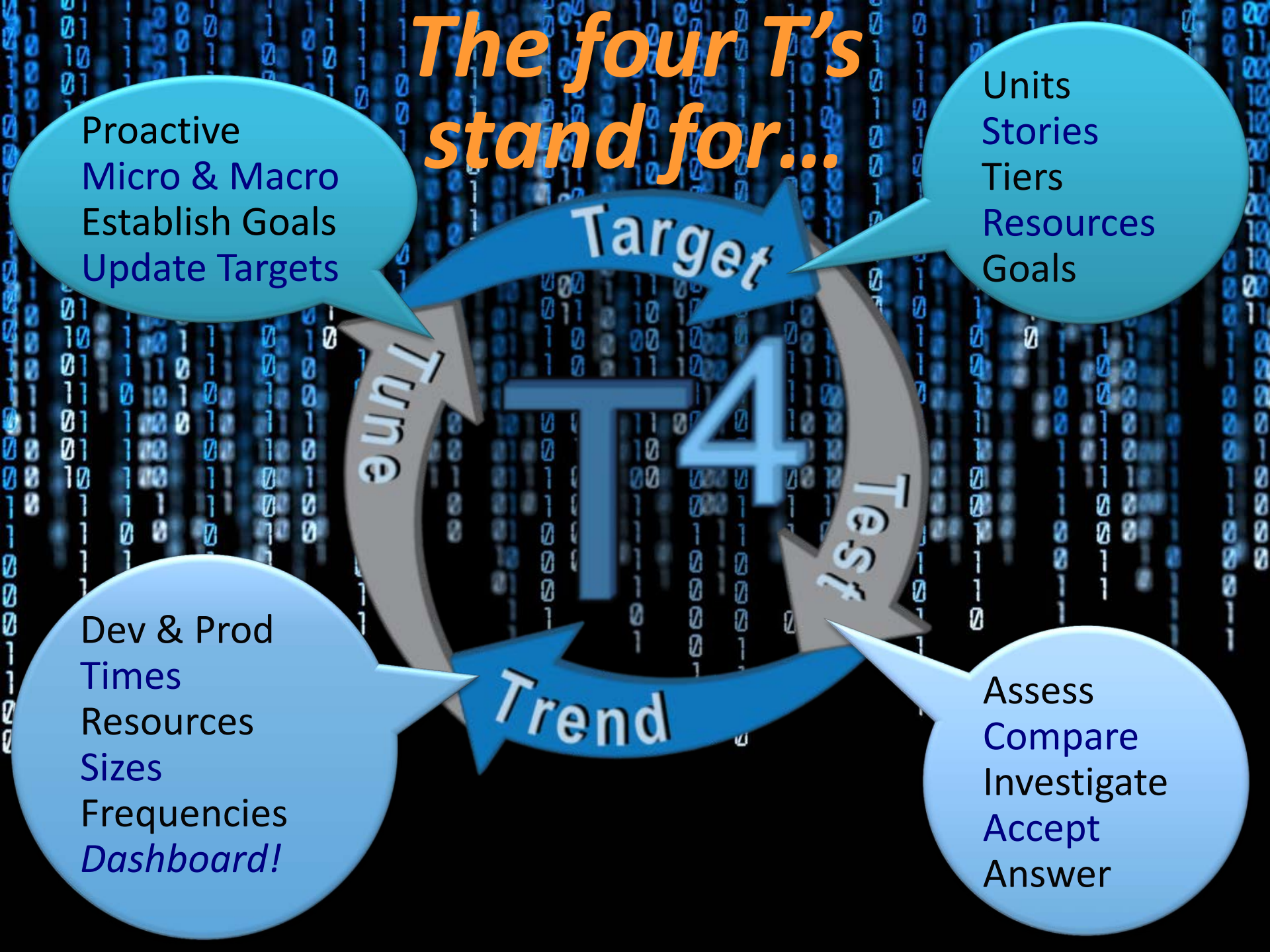

Want to
know how?

With:

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

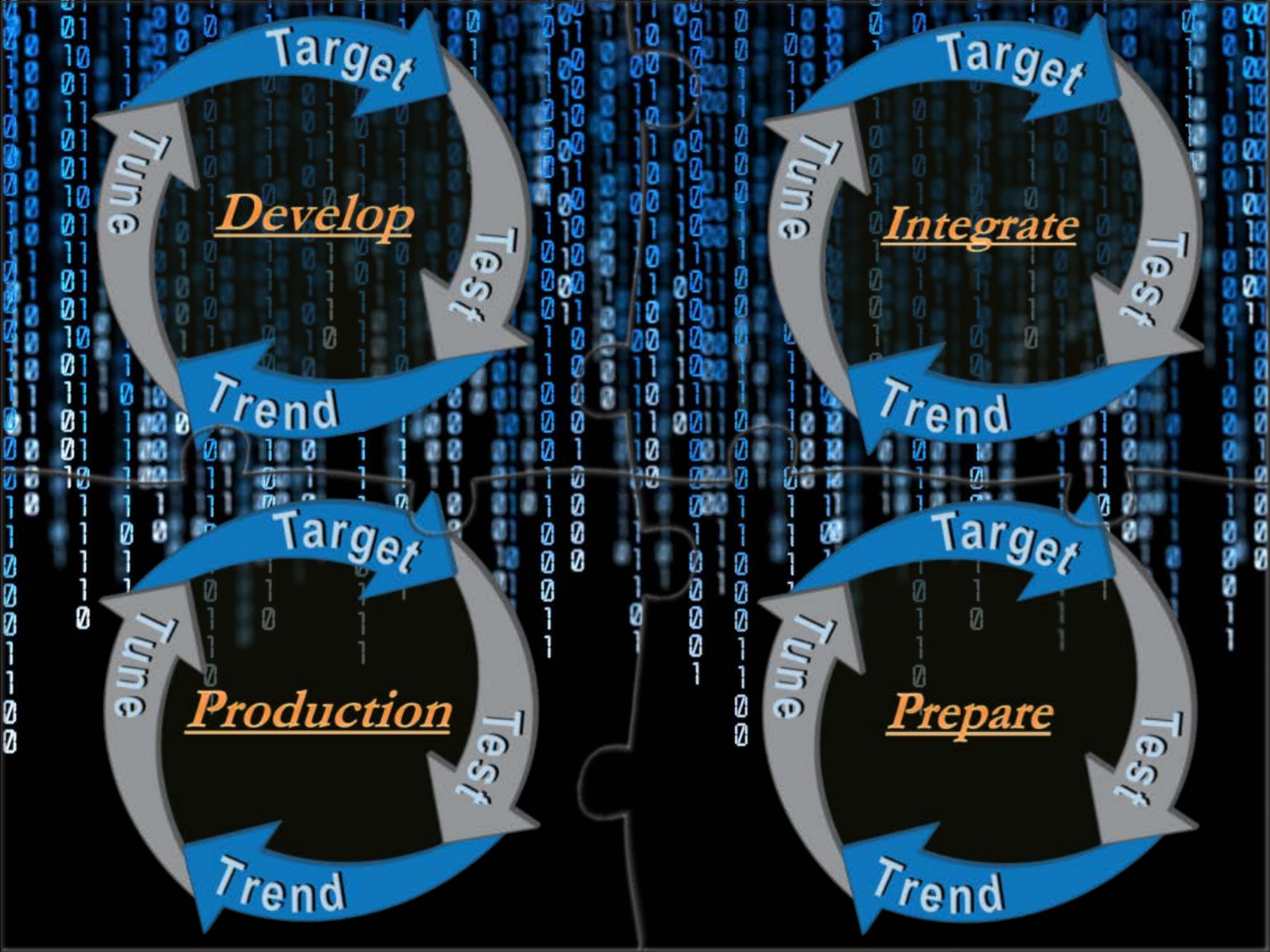
Dev & Prod
Times
Resources
Sizes
Frequencies
Dashboard!

Assess
Compare
Investigate
Accept
Answer

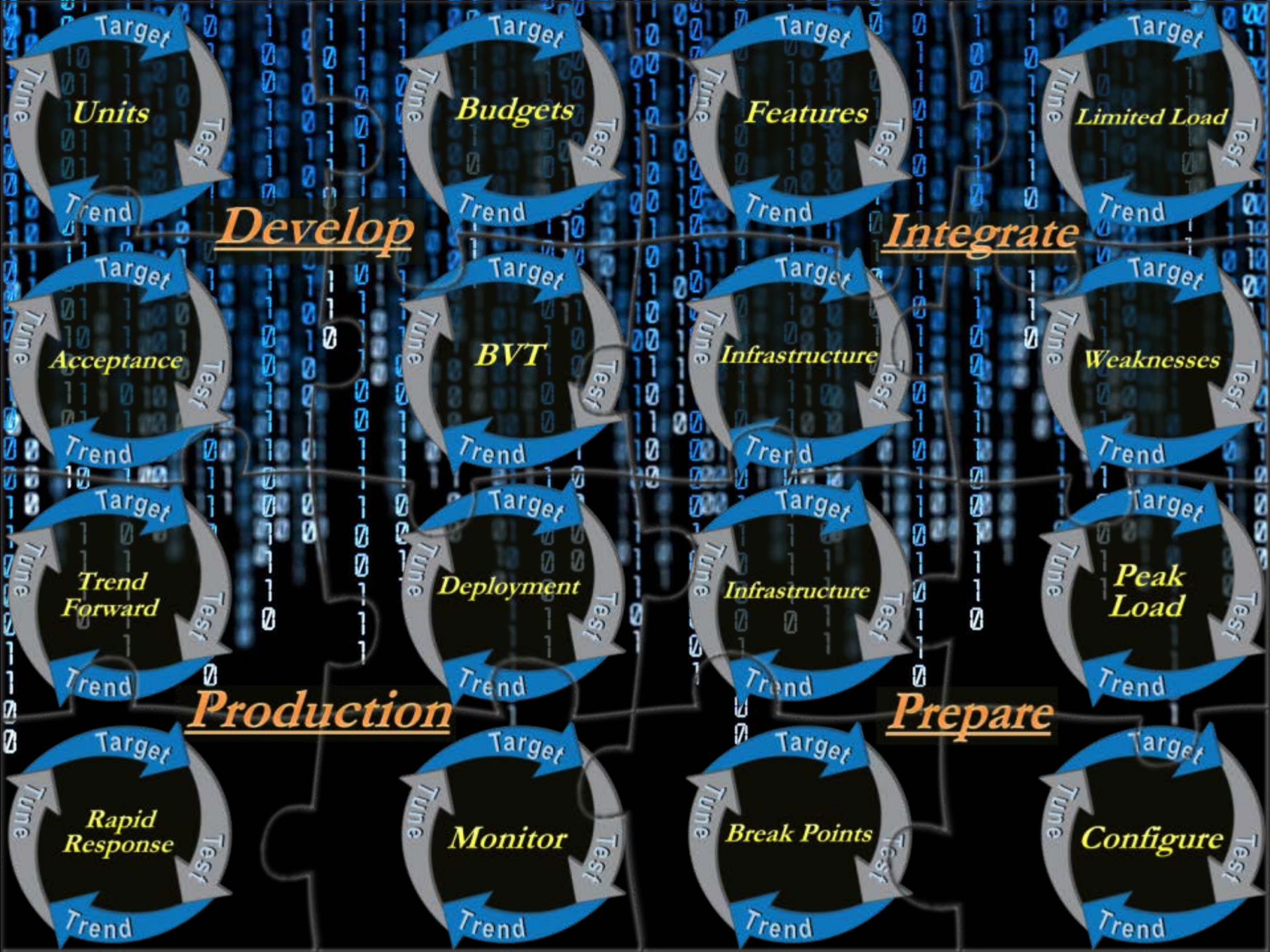

... and you can
apply them
like...


Target
Develop
Integrate
Tune
Test
T4
Production
Prepare
Trend

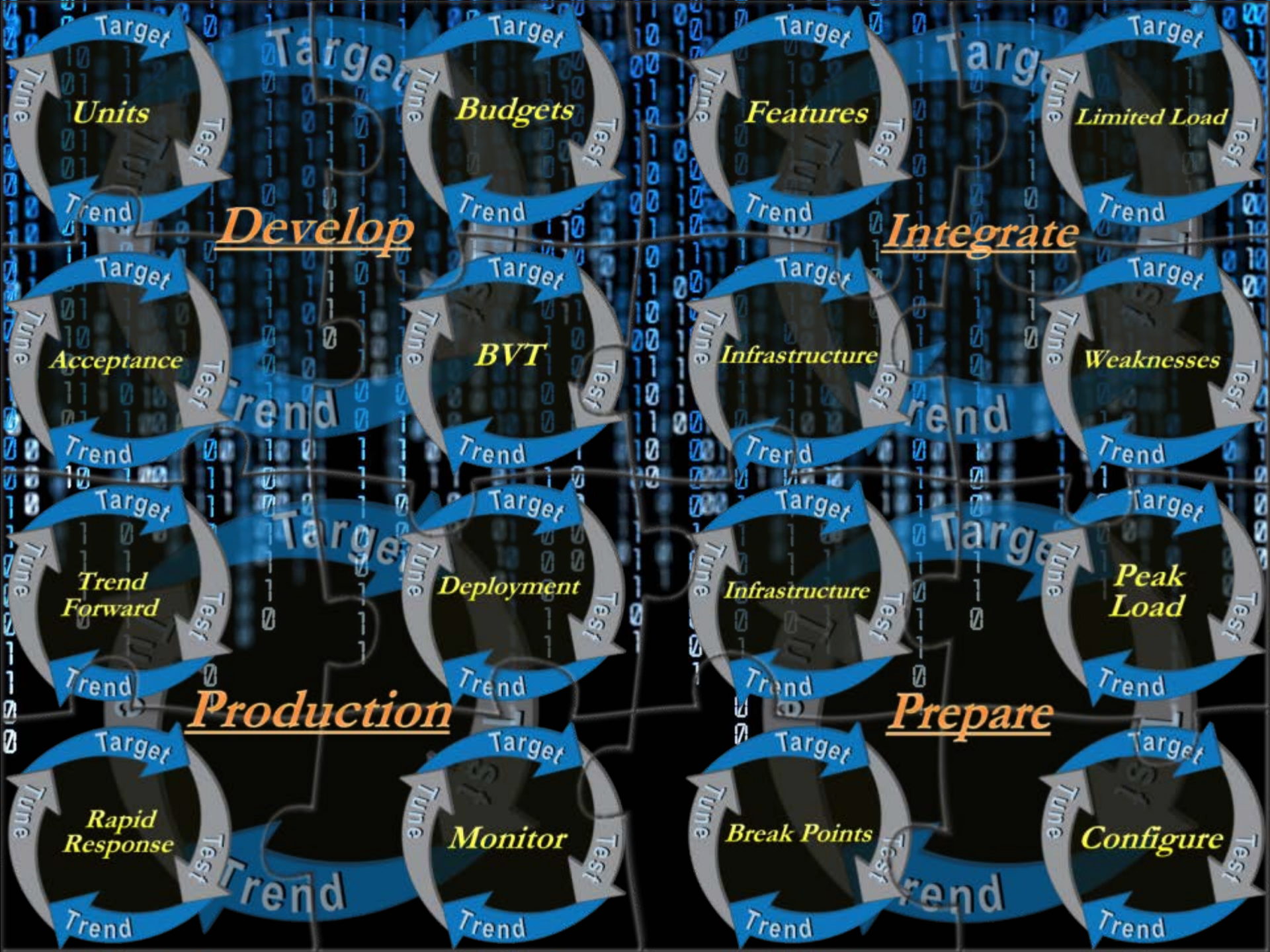

... or ...


Target
Test
Trend
Tune
Develop
Target
Test
Trend
Tune
Integrate
Target
Test
Trend
Tune
Production
Target
Test
Trend
Tune
Prepare

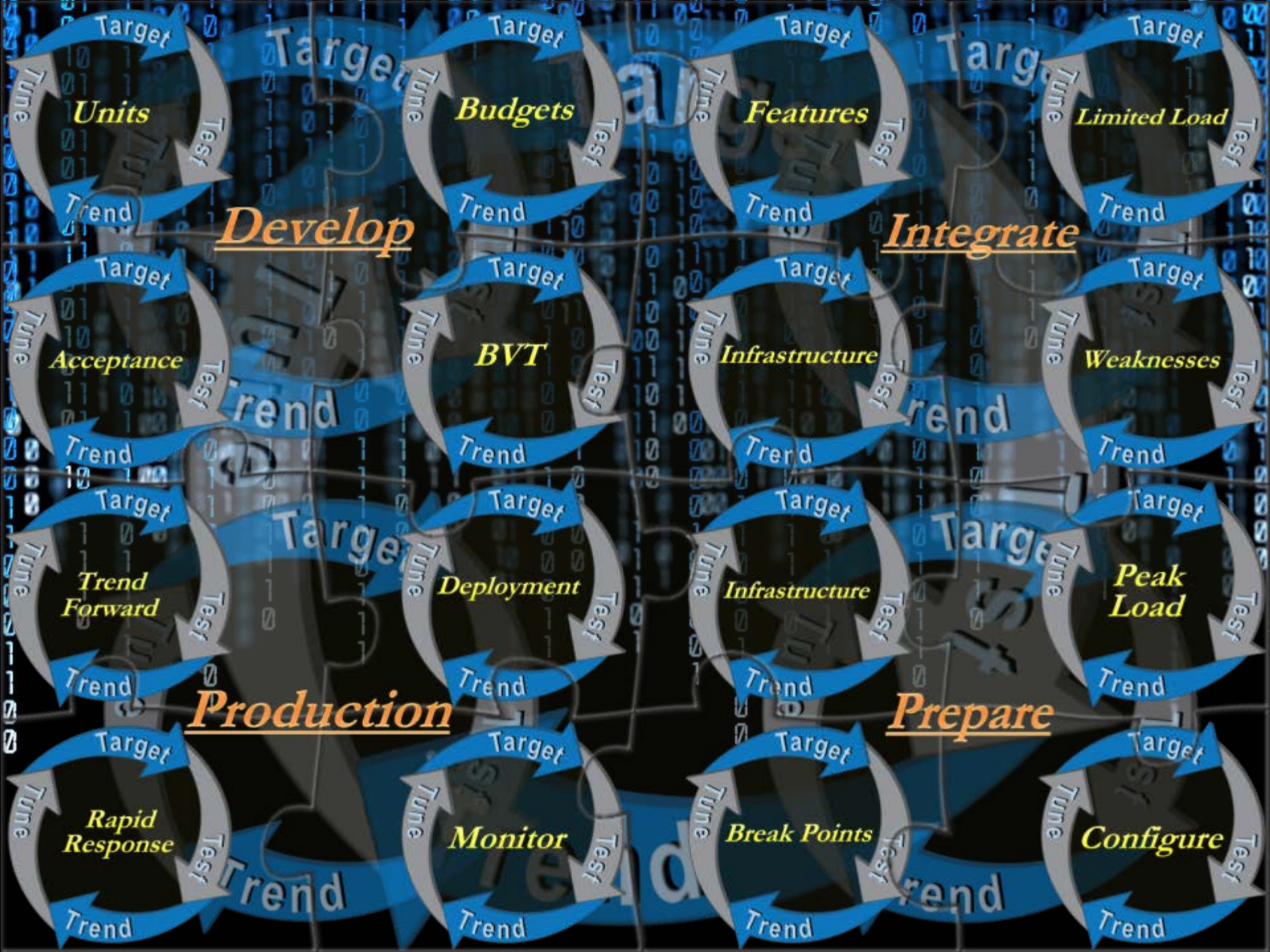

... or ...


Develop
Units
Budgets
Features
Limited Load
Integrate
Acceptance
BVT
Infrastructure
Weaknesses
Trend Forward
Deployment
Infrastructure
Peak Load
Production
Prepare
Rapid Response
Monitor
Break Points
Configure
Target
Tune
Test
Trend


... but and
is better ...

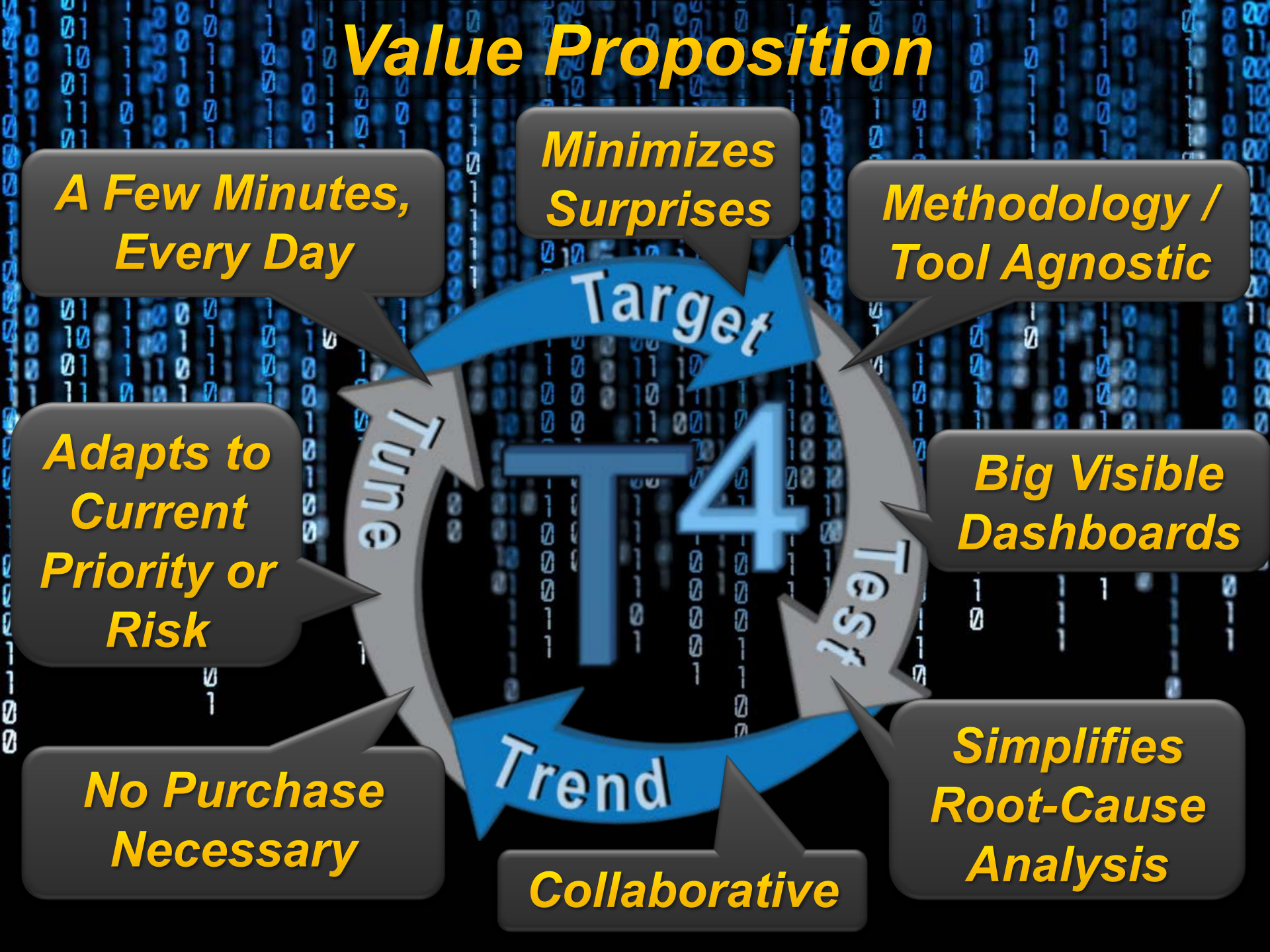

Develop
Units
Budgets
Acceptance
BVT
Trend Forward
Deployment
Rapid Response
Monitor
Integrate
Features
Limited Load
Infrastructure
Weaknesses
Infrastructure
Peak Load
Break Points
Configure
Production
Prepare
Target
Tune
Test
Trend


... and more
ands are
even better!


Develop
Units
Budgets
Integrate
Features
Limited Load
Acceptance
BVT
Infrastructure
Weaknesses
Production
Trend Forward
Deployment
Infrastructure
Peak Load
Prepare
Rapid Response
Monitor
Break Points
Configure
Target
Tune
Test
Trend


Let me help
you convince
the others...

Value Proposition



Challenges

Some Education Required

Demands Discipline

Culture Change is Hard

Several Cycles to Full Value

Footnote 1:

For best results I recommend bundling with...

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

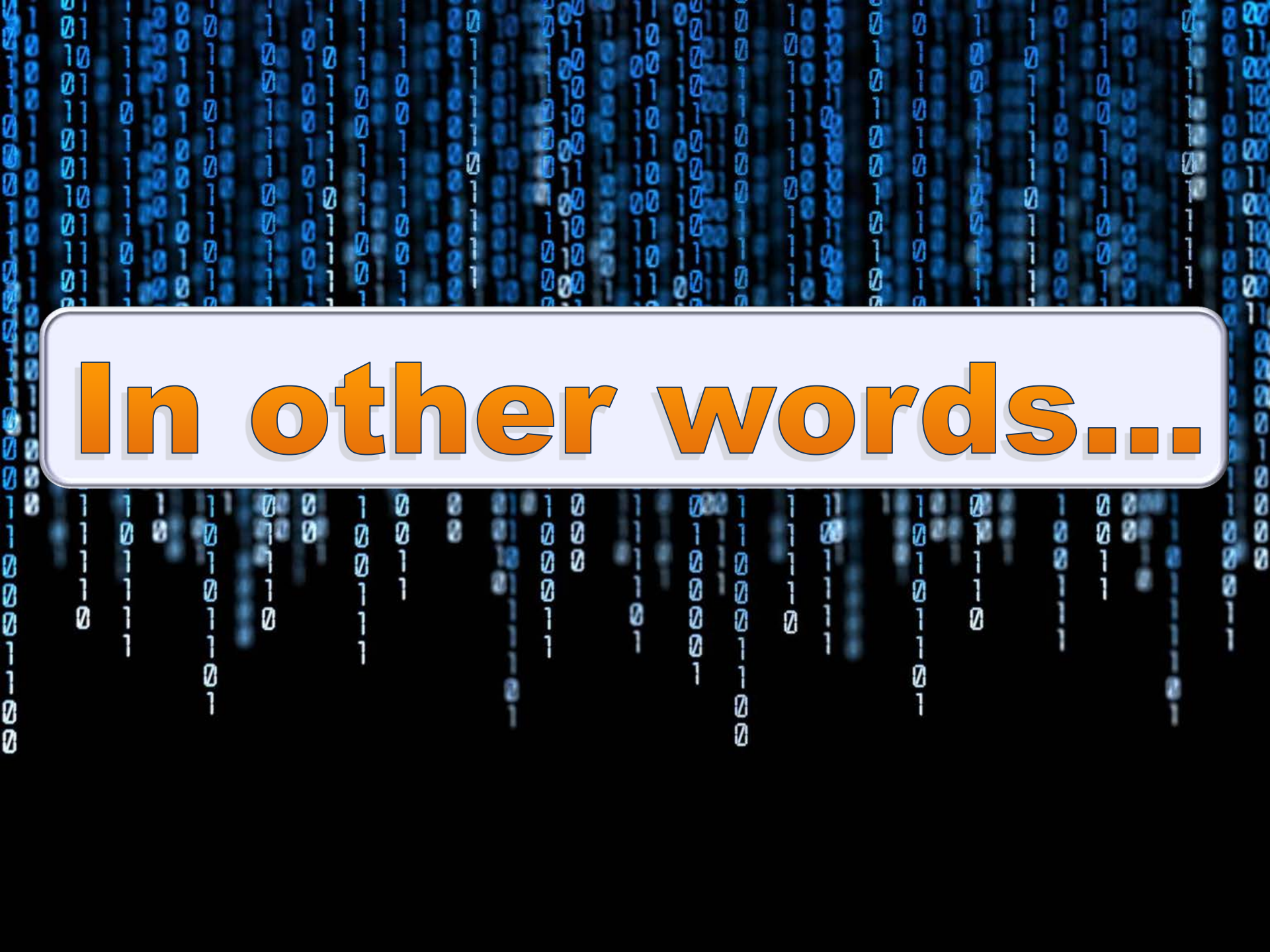

... and ...

Throughout the Life-Cycle

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.

Throughout the Life-Cycle


In other words...

APM™ + RPT + 

=>

A Practical & Holistic Approach for Achieving Fast & Scalable Systems


Footnote 2:

Unit-Level Testing Tools

(‘cause folks always ask)

FireBenchmarks; Performance testing addin for NUnit

JUnitPerf; a collection of JUnit test decorators for performance

Firefox Performance Tester's Pack

HTTPerf

Footnote 3:

These slides have been given to iqnite to post/distribute & they will go on my site shortly.

Questions?

APM™

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

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?