

A photograph of two dogs, one black and white and one brown, playing together in a grassy park with fallen leaves.

What is DevOps?

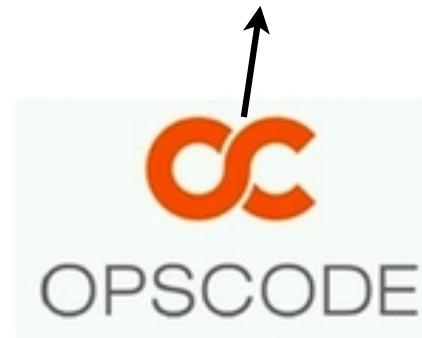
John Willis

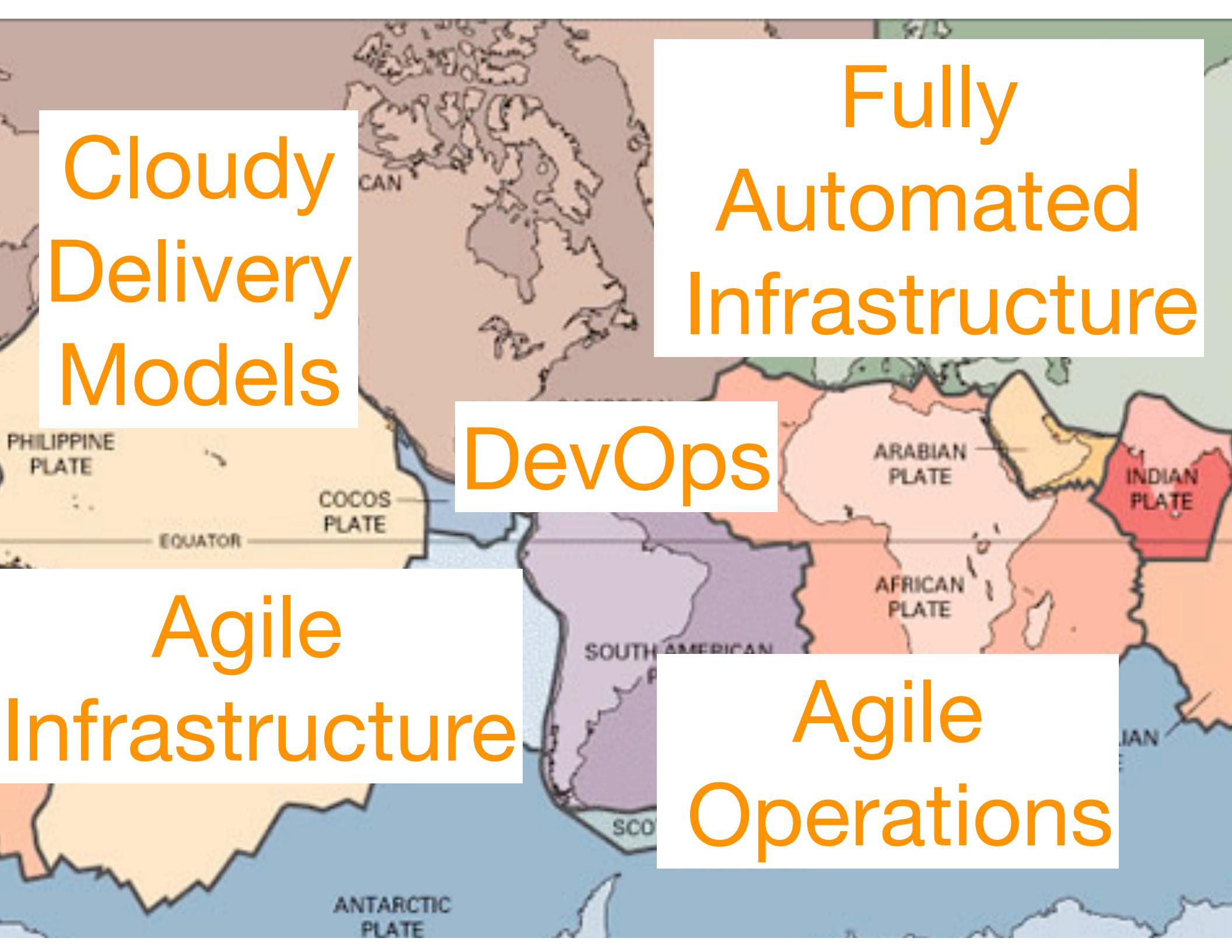


Damon Edwards

Dev Ops *Cafe*

With
John Willis
Damon Edwards





Cloudy
Delivery
Models

Fully
Automated
Infrastructure

DevOps

Agile
Infrastructure

Agile
Operations

“We are in a period of combinatorial innovation”



- Abstract and fault tolerant components
- Integrated network accessible services
- Unlimited infrastructure

Hal Varian

Chief Economist, Google

Hierarchy Theory

***The rate of evolution of
any system is dependent
upon the organization of
its subsystems***



Herbert Simon

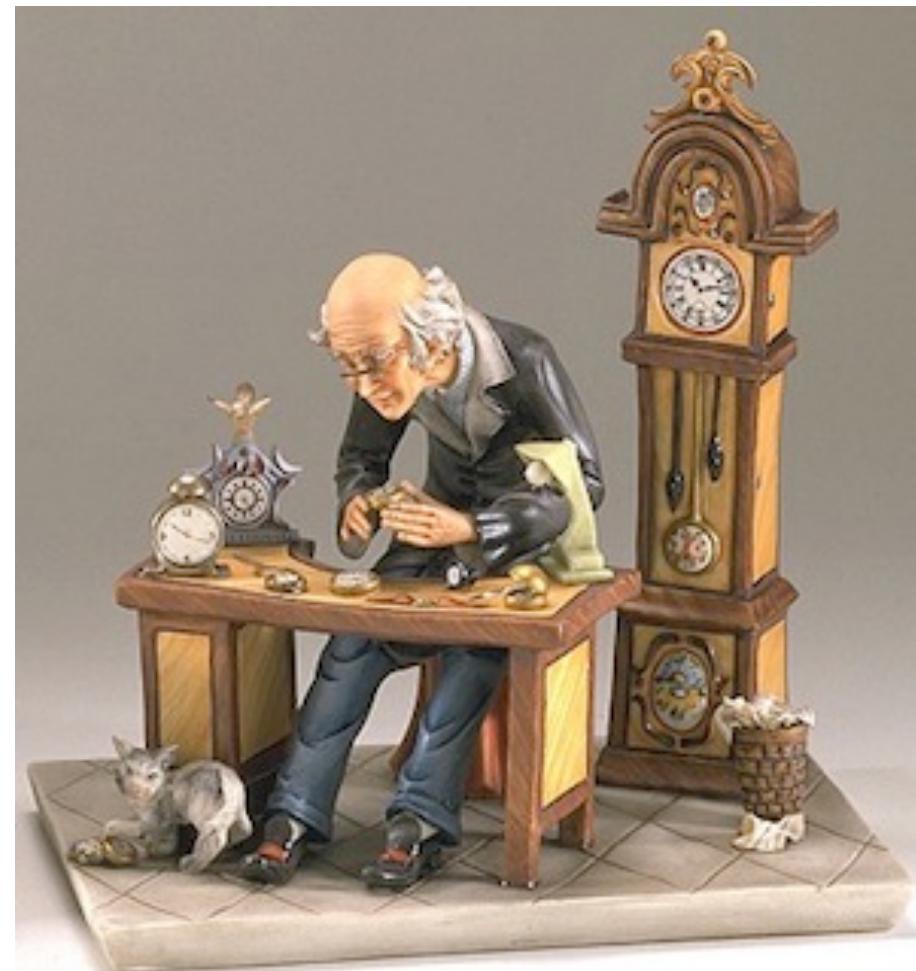
Theory of Hierarchy and
Componentisation

A Parable of Two Watch Makers

**Technological evolution
(and revolution)**

**Introductions of new
products**

**Strong and unpredictable
demand fluctuations**



Operations: The Elephant in the Room

“Once we move to software as a service, everything we thought we knew about competitive advantage has to be rethought.”

Tim O'Reilly
Operations: The New Secret Sauce

Operations: The Strategic Weapon

“This advantage is the ability to consistently create and deploy reliable software to an unreliable platform that scales horizontally.”

Jesse Robbins

Operations is a competitive advantage...
(Secret Sauce for Startups!)

How can we compete today?

1. Scale

2. Velocity of Innovation

What is DevOps?

~~What is DevOps?~~

~~What is DevOps?~~

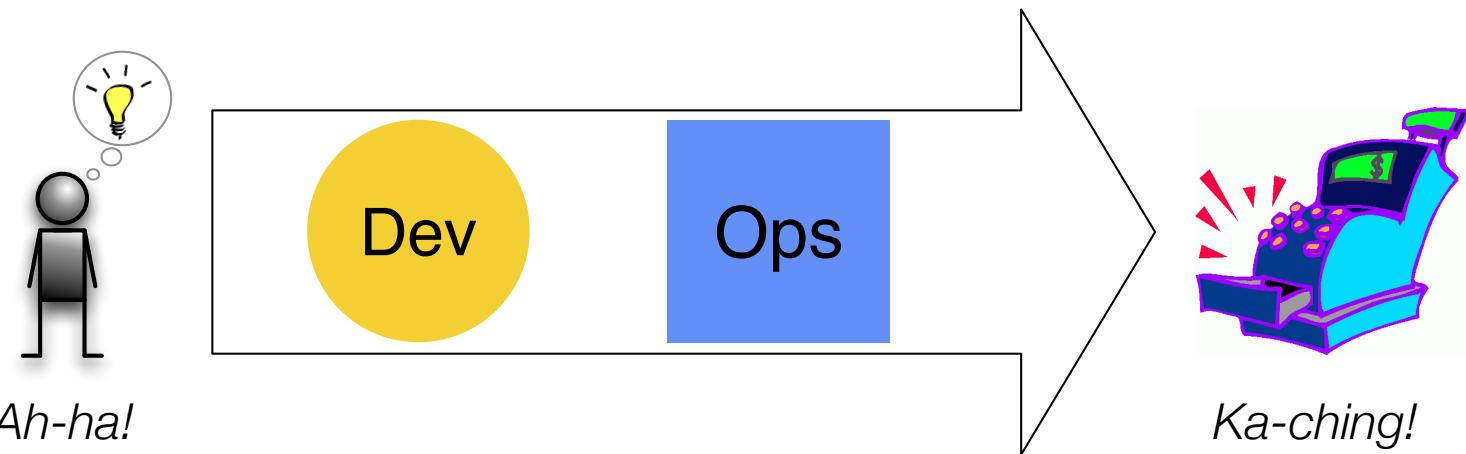
What are DevOps problems?

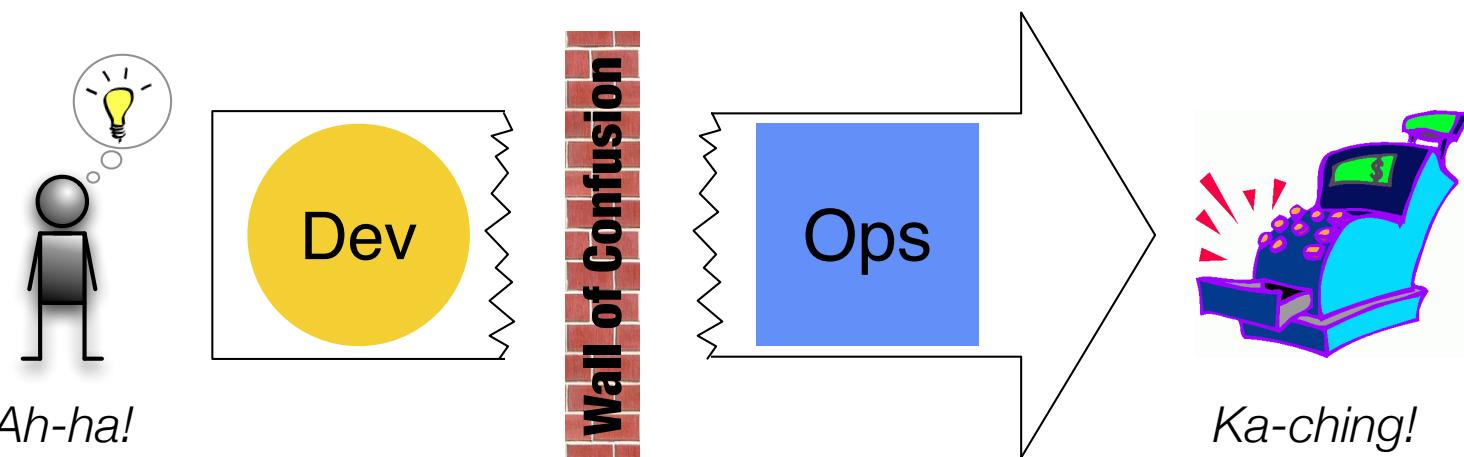


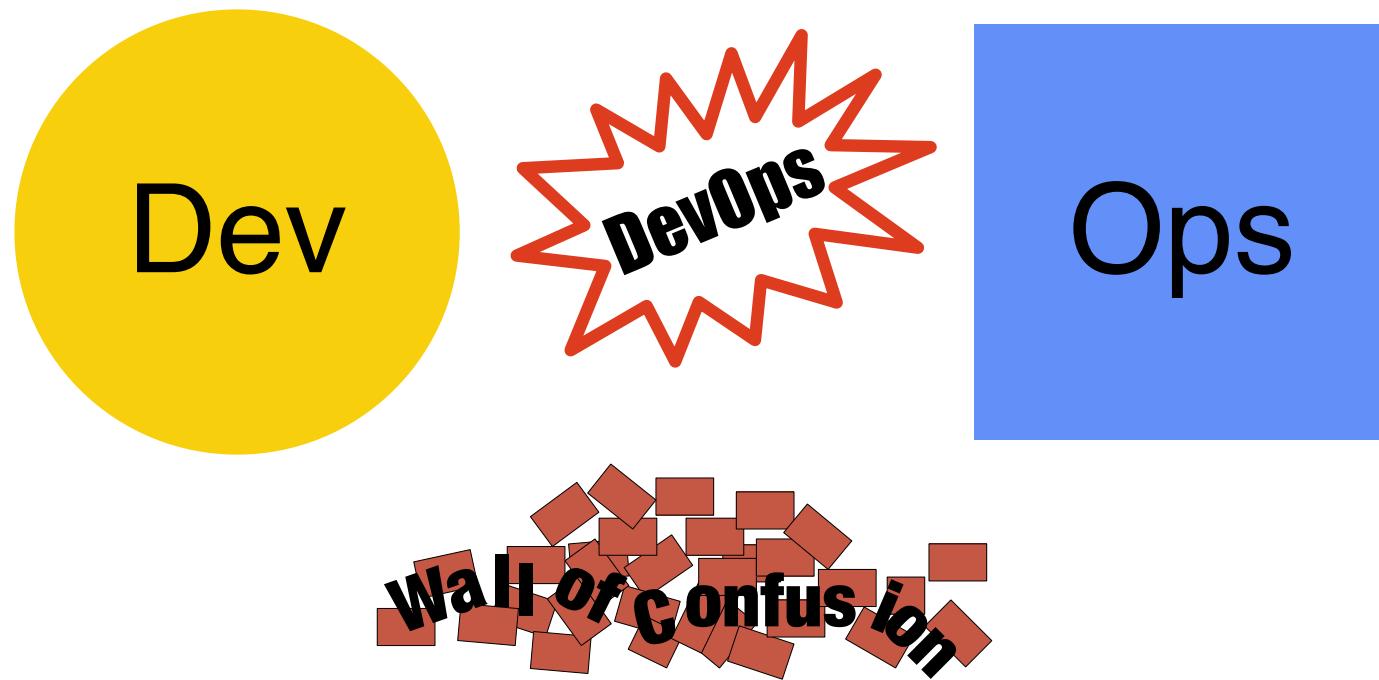
Ah-ha!

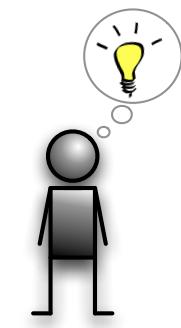


Ka-ching!

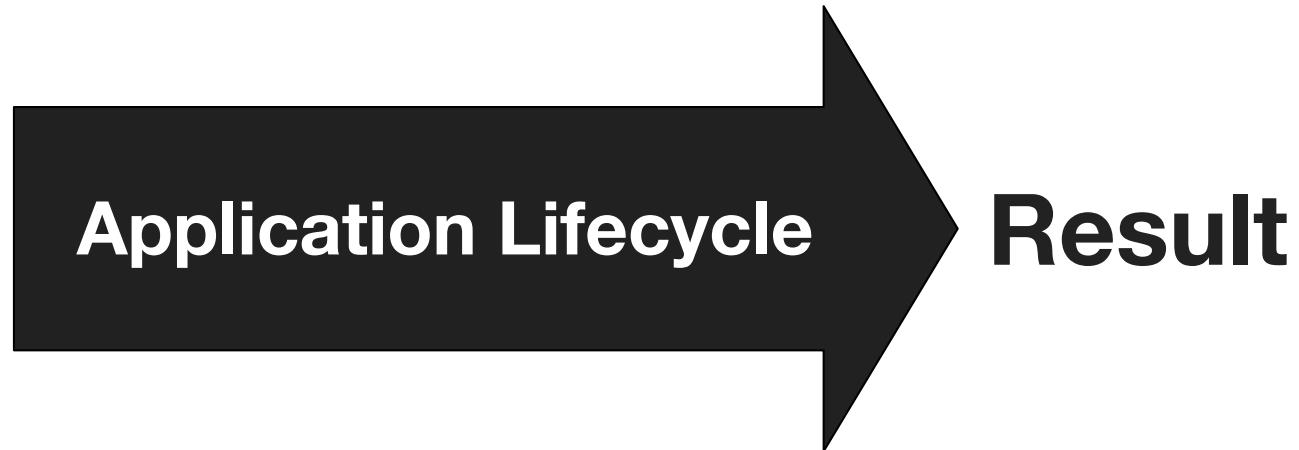








Ah-ha!





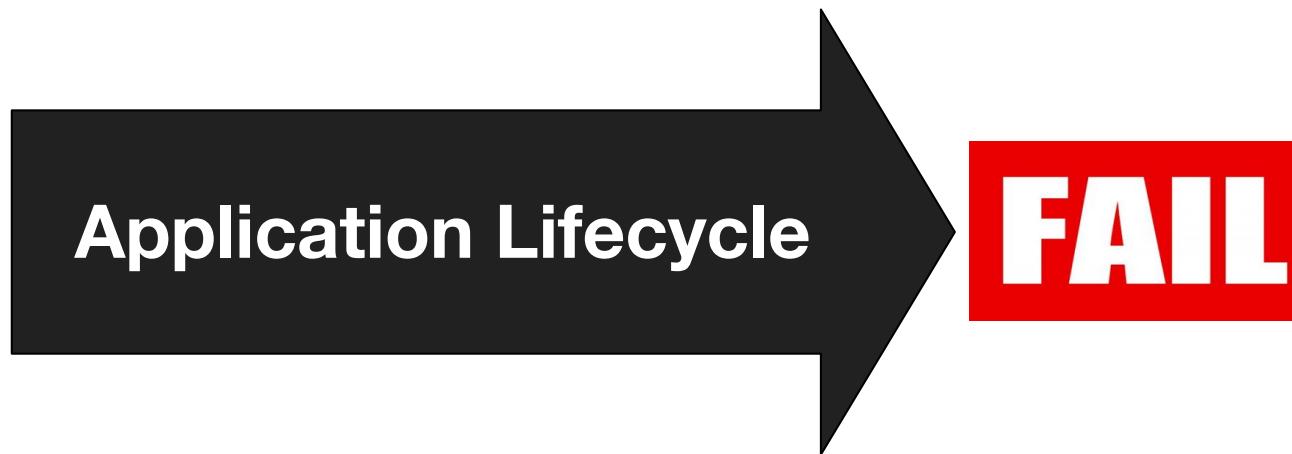
Ah-ha!

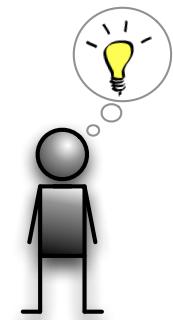


Ka-ching!



Ah-ha!



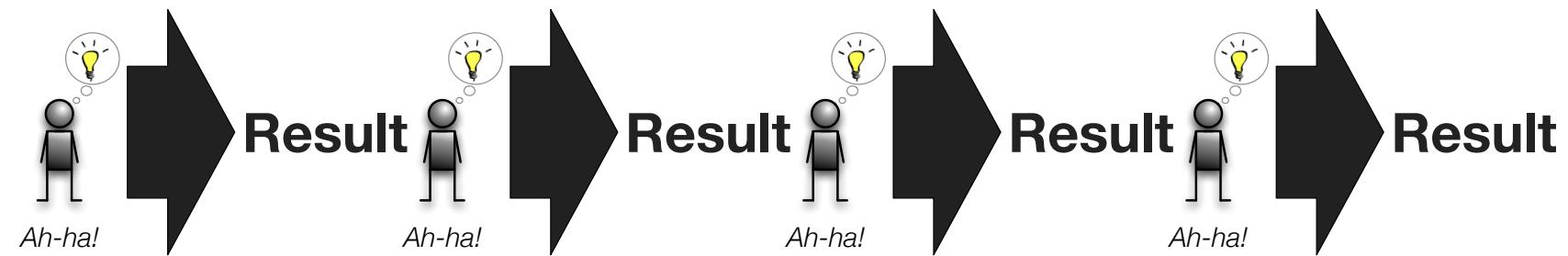


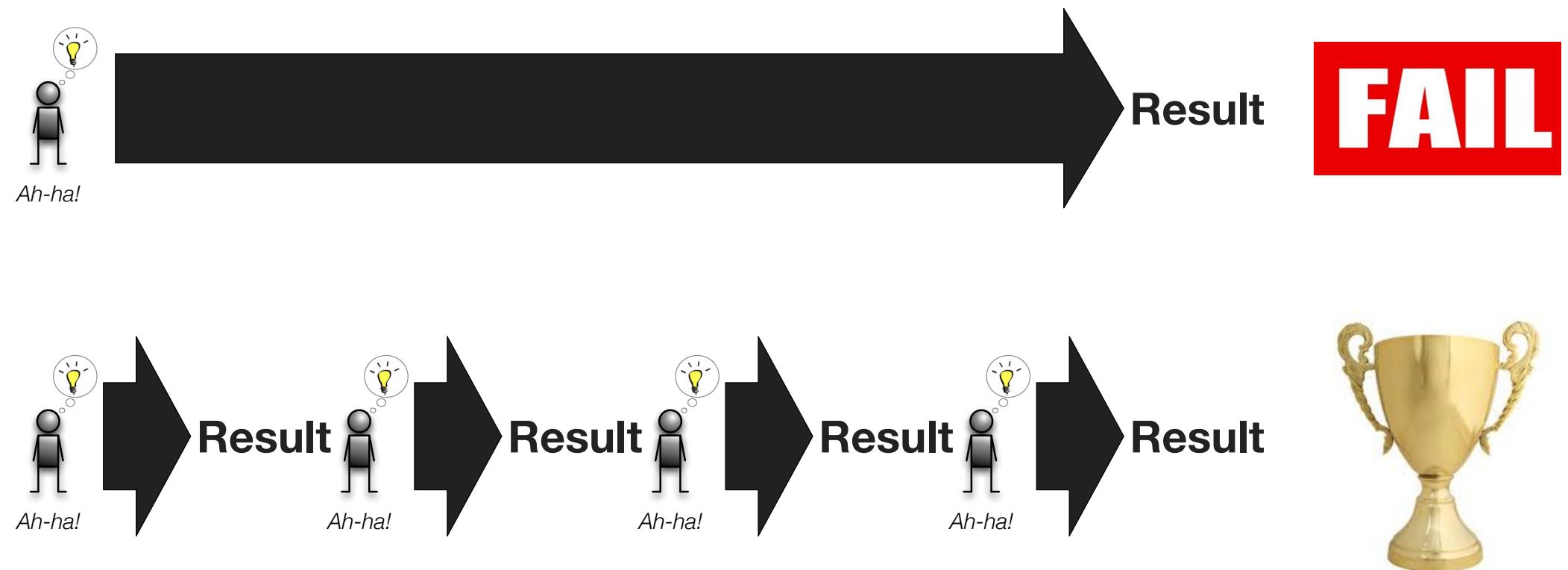
Ah-ha!



Ka-ching!

| -----> |
Day 0 **Day n**





**Sounds good...
But what can I do?**

C

A

M

S

Culture

A

M

S



Little bit weird
Sits closer to the boss
Thinks too hard



Pulls levers & turns knobs
Easily excited
Yells a lot in emergencies

John Allspaw and Paul Hammond
Velocity 2009

People
over
Process
over
Tools

**“Make the problem the
enemy”**

Jody Mulkey
CIO, Shopzilla

“You can’t directly change culture. But you can change behavior, and behavior becomes culture”

Lloyd Taylor
VP Infrastructure, ngmoco:)

Culture

Automation

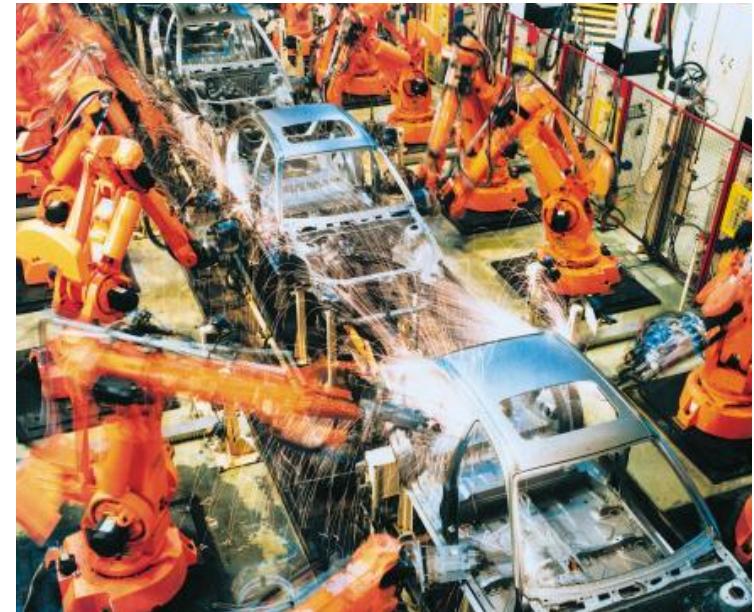
M

S

Get fully automated. Period.



Web Operations 1.0



Web Operations 2.0

Infrastructure as Code

- ▶ Version control
- ▶ Process control
- ▶ Application control

Infrastructure as code...

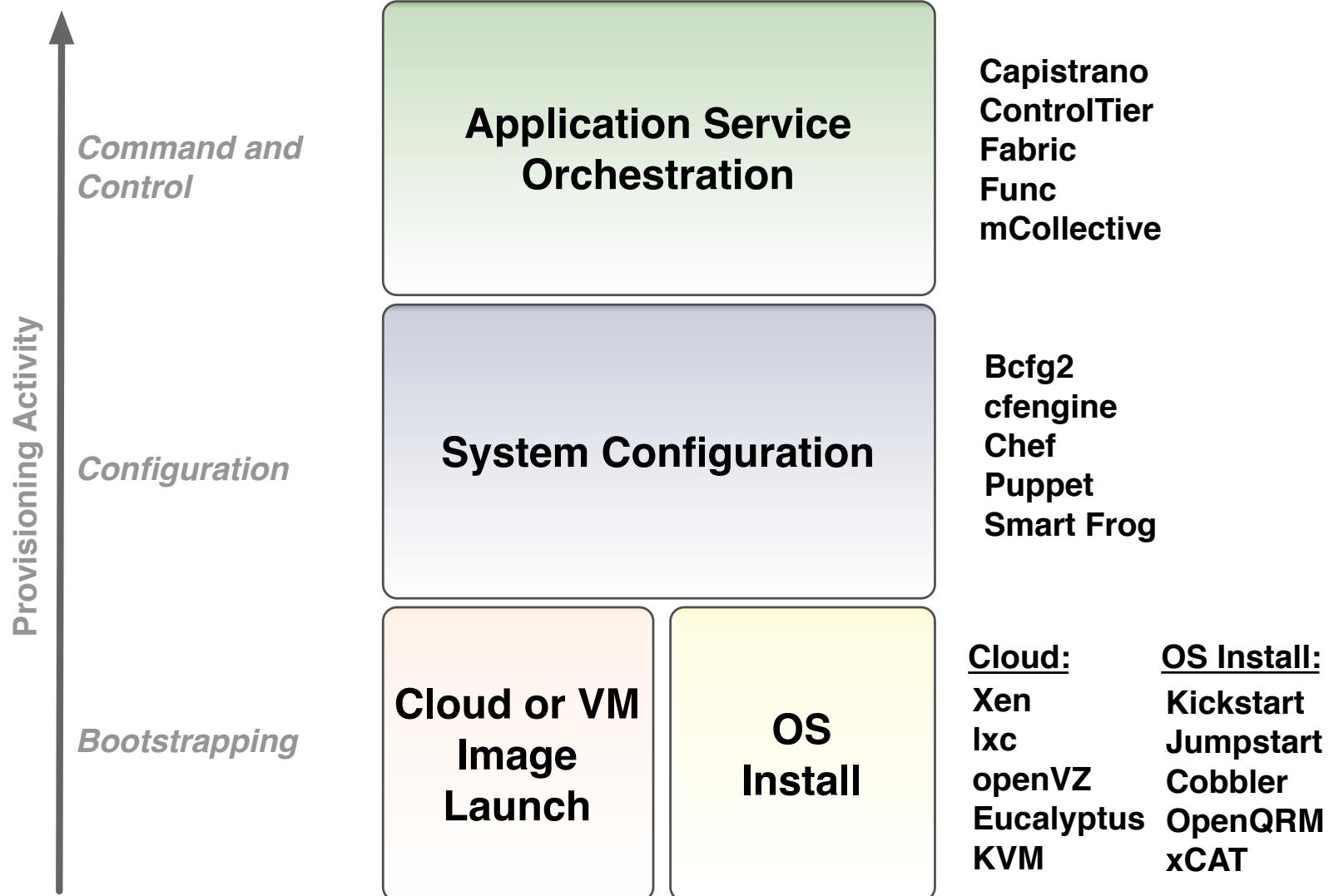
“10th floor test”

Can you automatically restore your services if you:

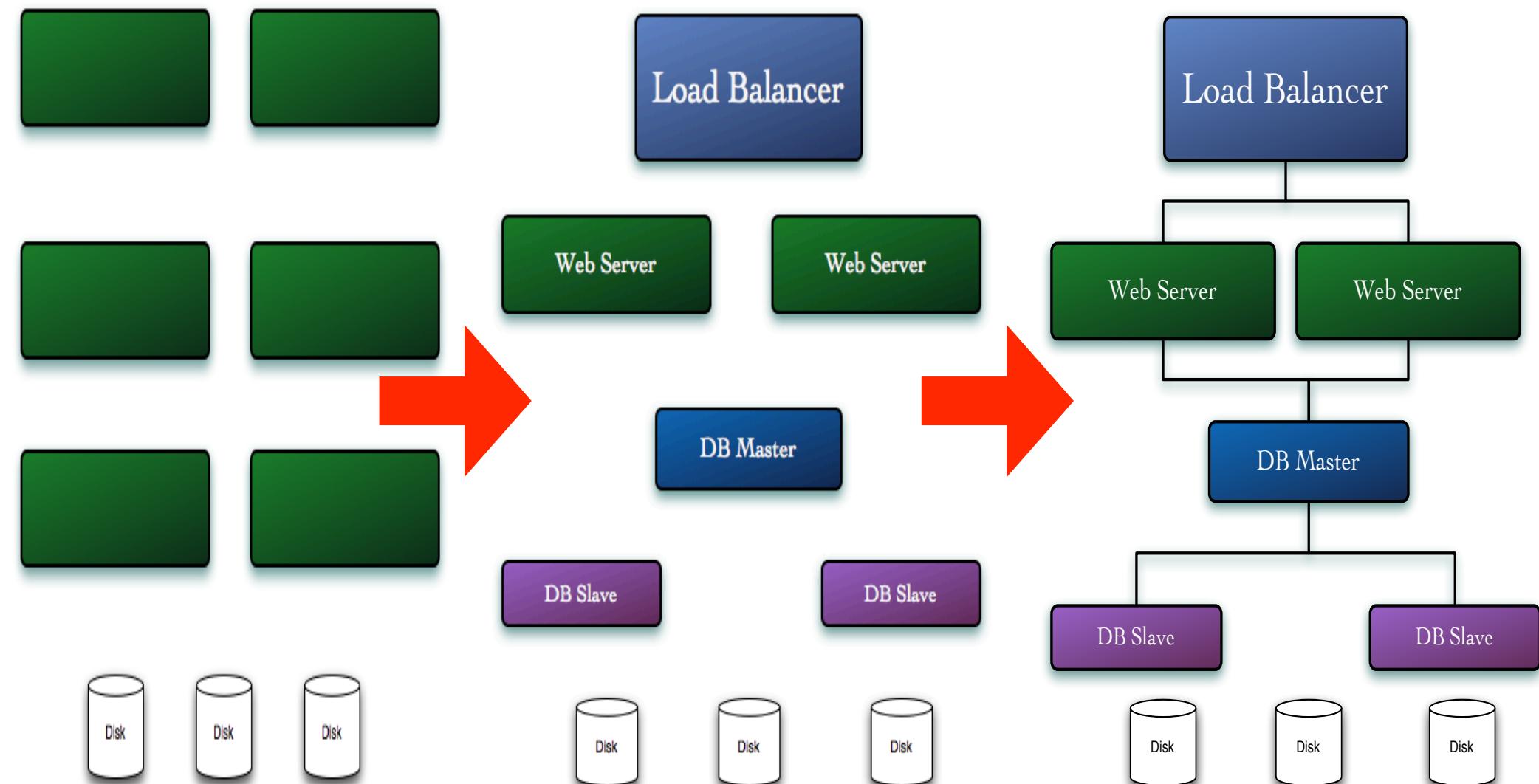
- 1. Toss a random server out the window**

- 2. Toss a random engineer out the window**

Use toolchains...



Provisioning Configuration Integration



Culture

Automation

Measurement

S

Measure Everything

Performance Metrics

Process Data

People Data

Heuristics

Am I actually measuring my...

processes?

people?

(hint: we are here to make those two things better)

Culture
Automation
Measurement
Sharing

1. ~~Request~~ Demand sharing!

2. DevOps Days

3. Meetups

4. Mailing Lists

DevOps Toolchain (Google Group)

Agile Systems Administration (Google Group)

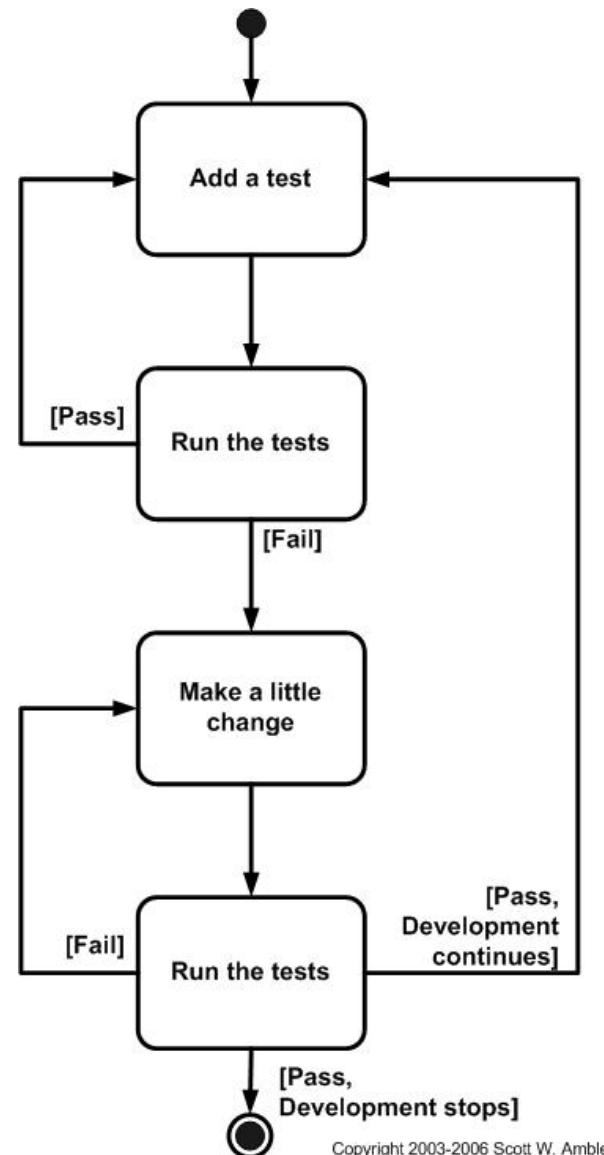
DevOps (Google Group)

5. #DevOps ... Tweet, Tweet, Tweet

Culture
Automation
Measurement
Sharing

Other things to think about...

Become Test Driven...



Initial realizations...

- 1. Quality is everyone's responsibility**
- 2. Traditional “QA” doesn't scale**

What does that mean for QA?

QA is not a team that “does”

QA is a cross-cutting concern

QA must leverage automation

Think Test Driven...



Do it in isolation

- Verify components in isolation (“unit”)
- Verify integrated components or subsystems (“functional”)
- End-to-end system tests (“system”)

on demand, low minutes

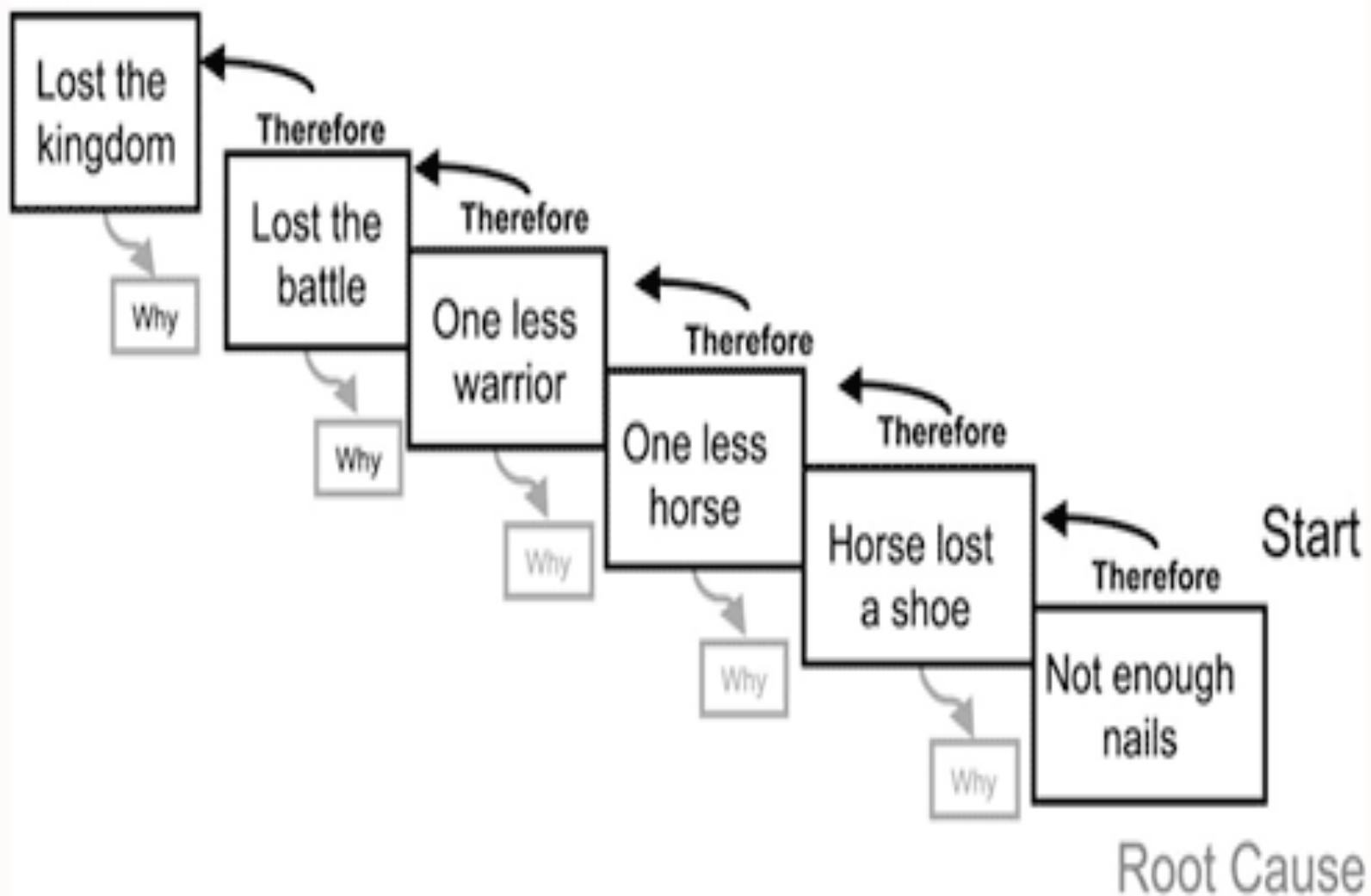


Do it with a crowd

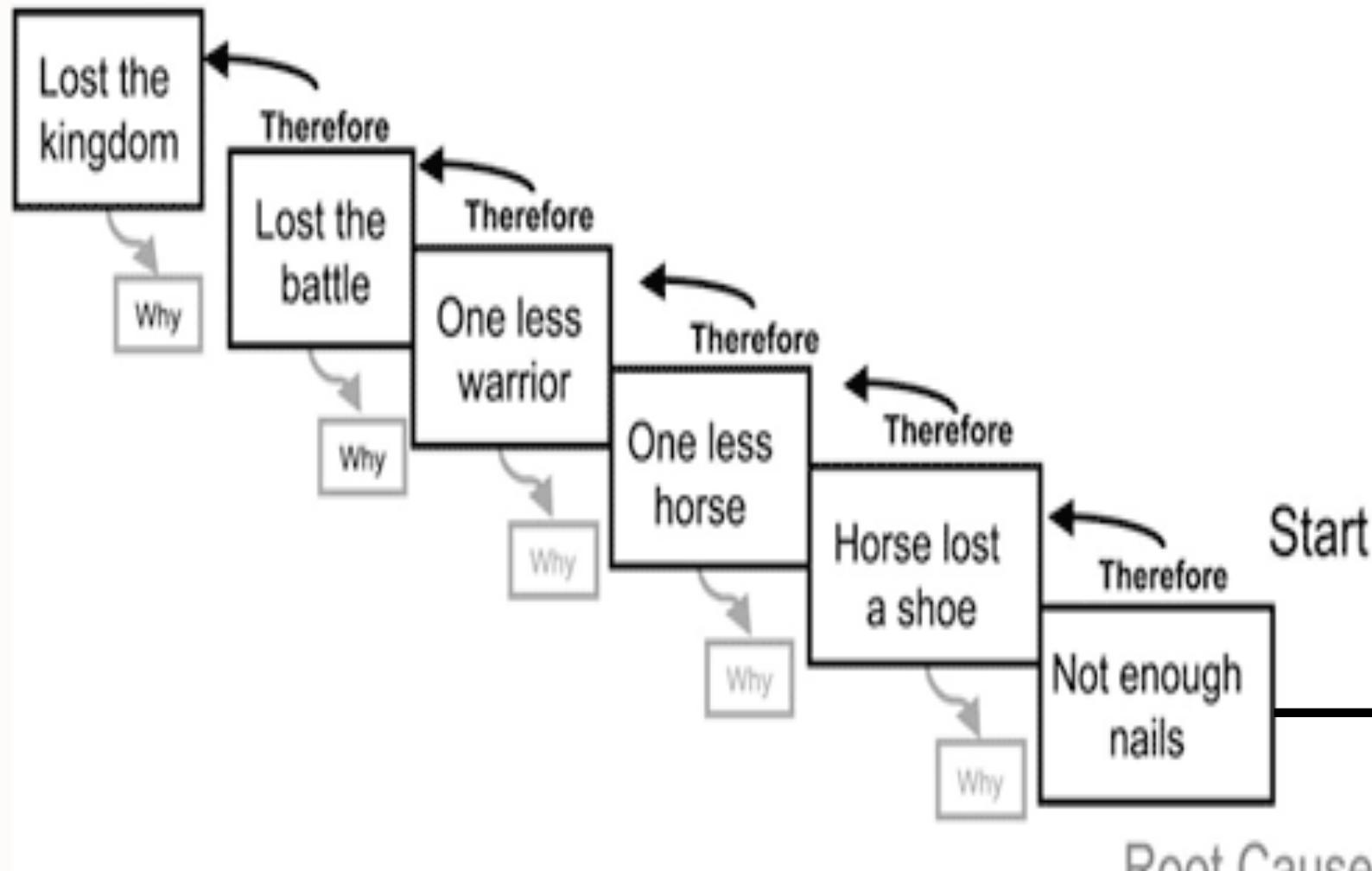
- Performance
- User interaction analysis
- Traffic analysis
- A/B Testing / Feature Flags
- Operational unit tests

continuous, real-time

Result



Result



Start

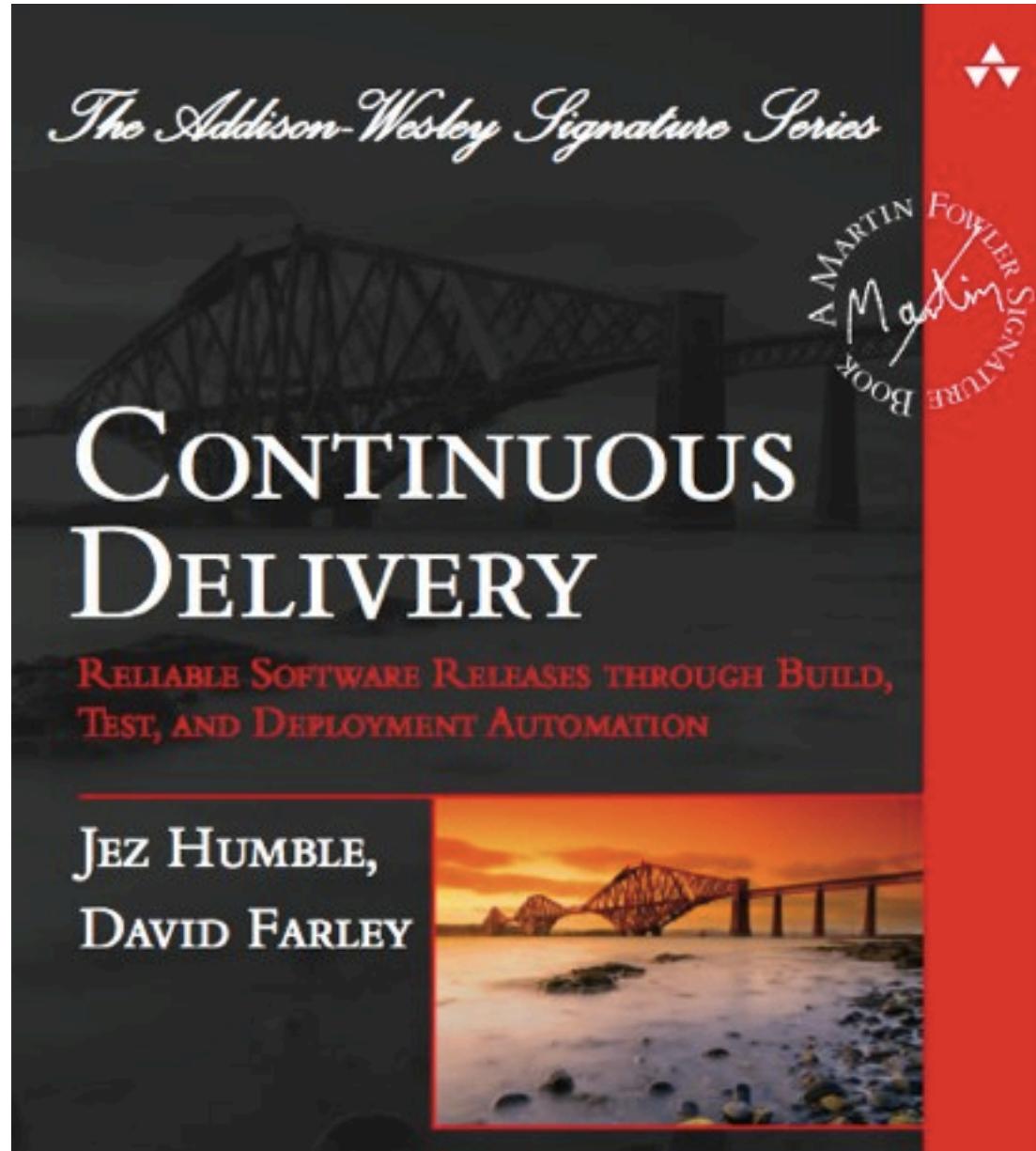
Root Cause

**Write a
test!**

**“Application code is a
business *liability*. Tests are a
business asset”**

Lee Thompson
CTO, Consumer Travel, HP

Continuous Delivery



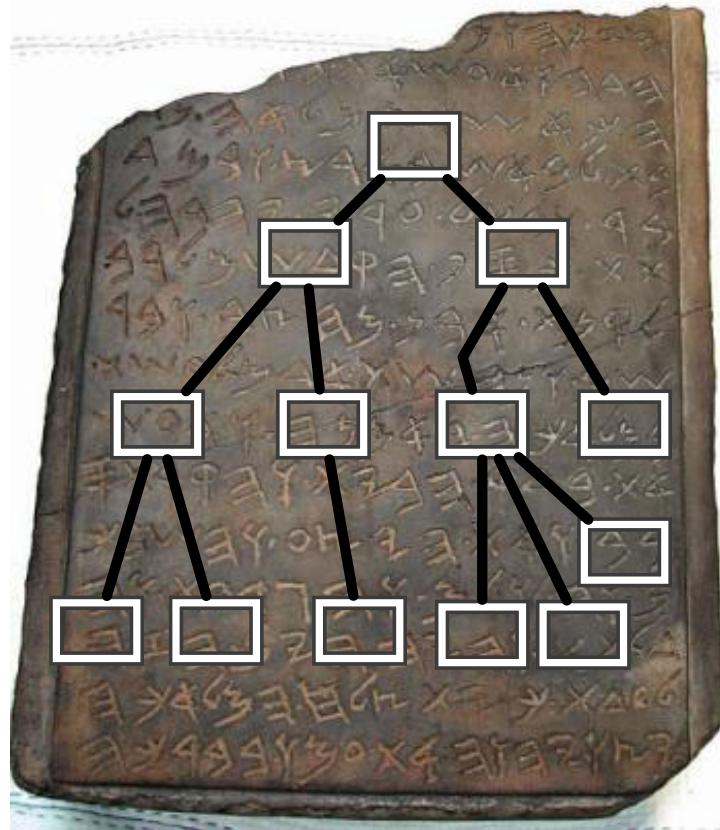
Principles of Continuous Delivery

- Repeatable, Reliable Release Process
- Automate Almost Everything
- Keep Everything in Version Control
- Bring the Pain Forward
- Build Quality In
- Done Means Released
- Everyone is Responsible
- Continuous Improvement

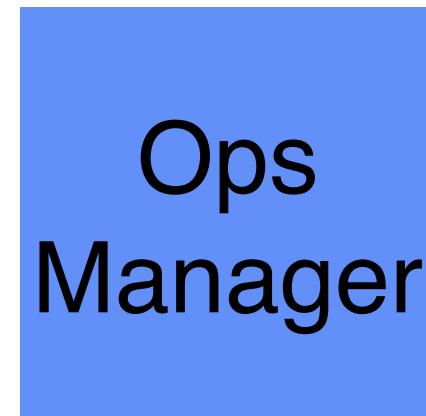
Continuous Delivery in a Nutshell

- Source Code Management
- Test Driven Development
- Continuous Integration
- Continuous Deployment
- Infrastructure as Code
- Fully Automated Infrastructure
- Continuous Improvement

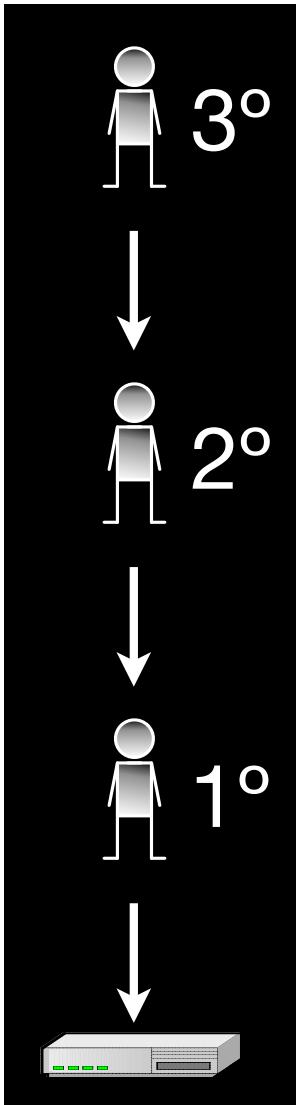
Org structures are not sacred!



Anti-Pattern #1:



Anti-Pattern #2:



If responsibility for full lifecycle of a service is *more than 3 degrees* away from the actual process

What Developers want...

Self-service

**Infrastructure is the app
(and vice versa)**

Not systems administrators

Faster feedback



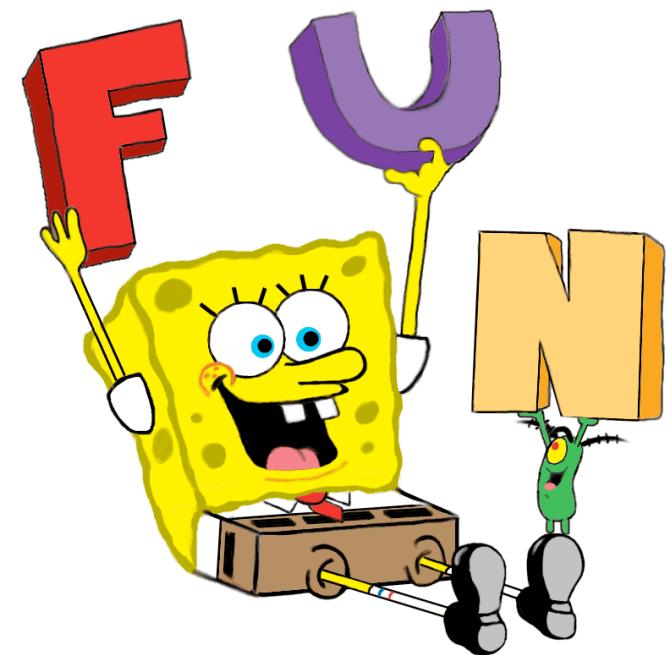
What Operations wants...

Get out of the muck

Spend time adding value

To get to say “Yes” more

Be “agile”



And one little pet peeve...

DevOps is *not* a job title!



John Willis
john@opscode.com
twitter.com/botchagalupe



Damon Edwards
damon@dtosolutions.com
twitter.com/damonedwards