



Transforming Government Agencies into Cloud-Native Organization Or, beyond “survival is not mandatory”

@cote

May 2017

Hello!



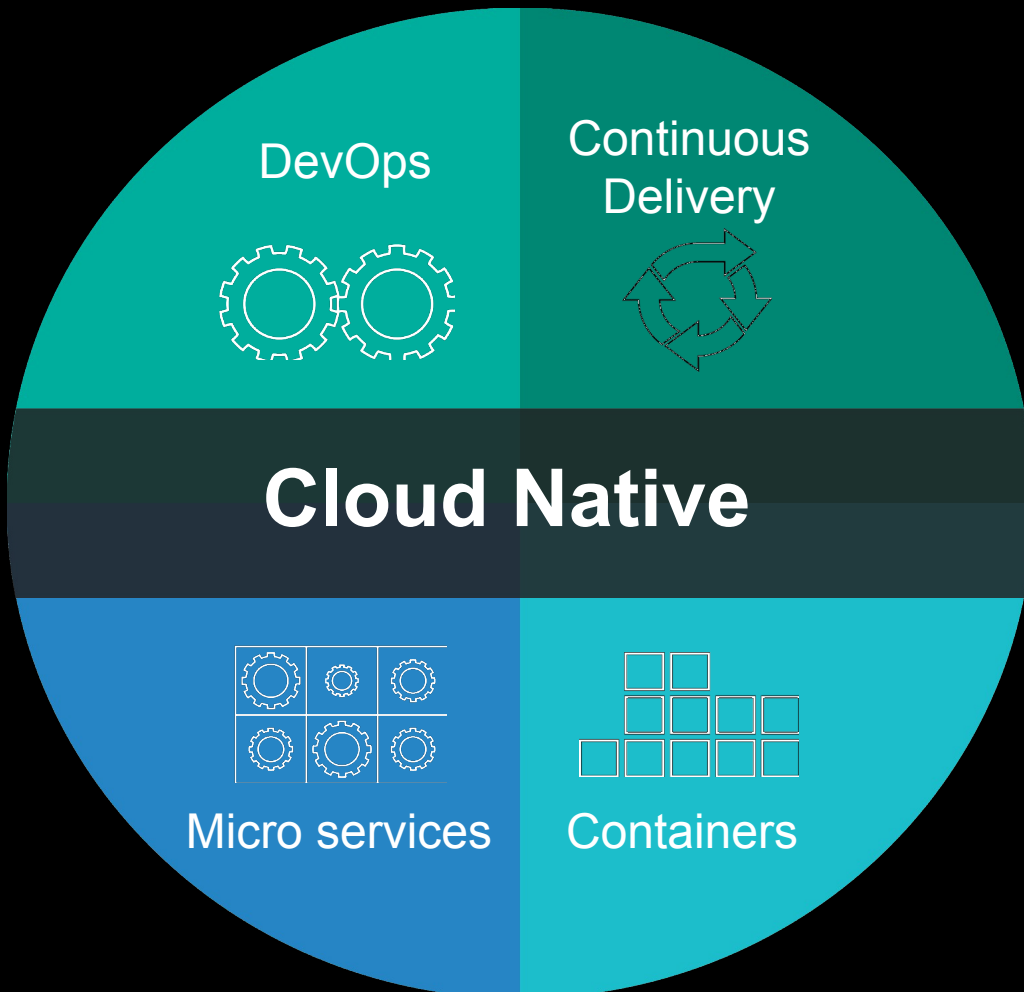
- [@cote](#) – Director, Marketing for Pivotal Cloud Foundry
- Former industry analyst at 451 Research and RedMonk
- Corporate strategy & M&A at Dell
- Former software developer
- Register [DevOps, agile, etc. column](#)
- Podcasts: [cote.io/podcasts](#)
- More: <http://cote.io> or cote@pivotal.io

Government IT concerns

- **Costs/removing waste** – 60-75% productivity boost
- **New capabilities** – 95% are “typical” or “trailing”
- **Dependability & longevity** – can’t go bankrupt
- **Open, portability, multi-cloud** - options
- **Compliance** - 14 months to 3 days w/ more controls
- **Security** - #2 concern in 5 year planning
- **Working with legacy** - ~60% of portfolio is pre-2010



#OMGWTF CloudNative



Cloud-native applications are purpose built for the cloud model. These applications - built and deployed in a rapid cadence by small, dedicated feature teams to a platform that offers easy scale-out and hardware decoupling - offer organizations greater agility, resilience, and portability across clouds.



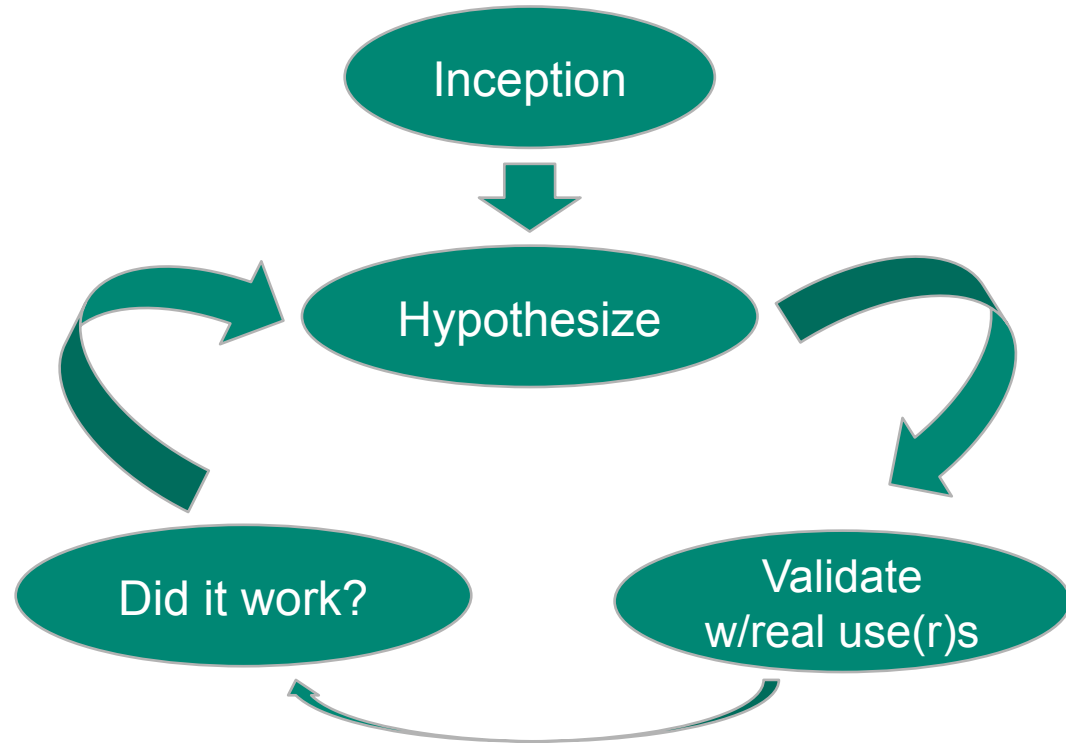
“In order to grow Citi, we first have to grow our own perspective, skills and capabilities... Our curiosity, our openness to learning and trying new things, our ability to adjust and adapt quickly and our willingness to fail fast and fail small are the essence of a culture that innovates and exposes new value to our clients in real time.”

Stephen Bird, CEO Citi Global Consumer Group



Goal: deliver value, weekly reliably, with small batches

“We did an analysis of hundreds of projects over a multiyear period. **The ones that delivered in less than a quarter succeeded about 80% of the time, while the ones that lasted more than a year failed at about the same rate.** We’re simply not very good at large efforts.”



Large financial institution

Sources: [“Good Software is a Series of Little Failures,”](#) Coté, April 2016; [The Lean Startup](#), Eric Ries, 2011. [The Lean Enterprise](#), Barry O’Reilly, Jez Humble, and Joanne Molesky. See [also overview of this approach at the IRS from Dec 2015](#). [“Application Modernization, Ser vice By Microservice,”](#) Kurt Bittner and Randy Heffner, Forrester, Dec 2015; [“Best Practices For Agile-Plus- Architecture,”](#) Randy Heffner, Forrester February, 2015.

Small batches & taxes

Before

Overview by Year			
YEAR	STATUS	AMOUNT	
2014	Balance Due	\$644	>
2013	Taxes Paid	\$685	>
2012	Refund/Applied	\$100	>
2011	No Information Available	--	

After

Overview by Year	
YEAR	BALANCE DUE
2014	\$644.00 >
2013	\$0.00
2012	\$0.00
2011	No Return on File >

It's actually working in mainstream, large shops



40%+ productivity/cost, rebooted member facing app



From 30% time coding to 90%



3+ week to 3 days, 50% reduction in incidents



Vitality drove engagement from 3% to 30%+




100% visibility into app portfolio



Delivered 3x features year/year





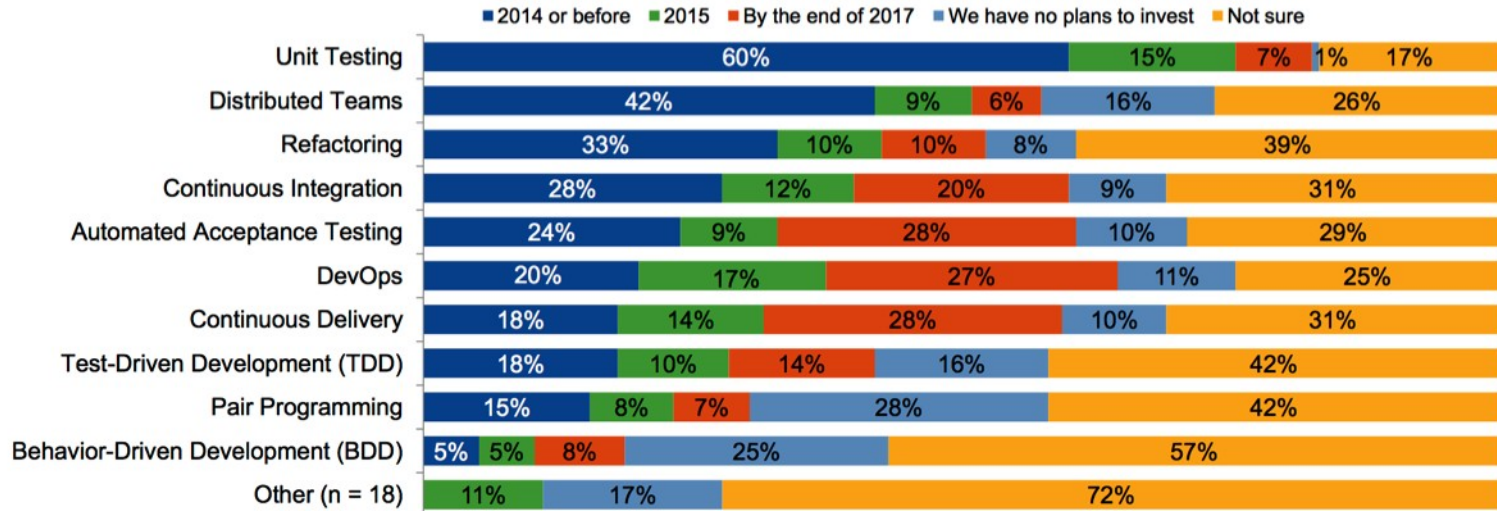
Skills & practices

Or, removing all the walls

“We’ve been doing AGILE since vacuum tubes.”

When did or will your organization begin to use each of the following agile enterprise practices?

n = 155; Base: Organization uses agile or is in pilot implementation phase



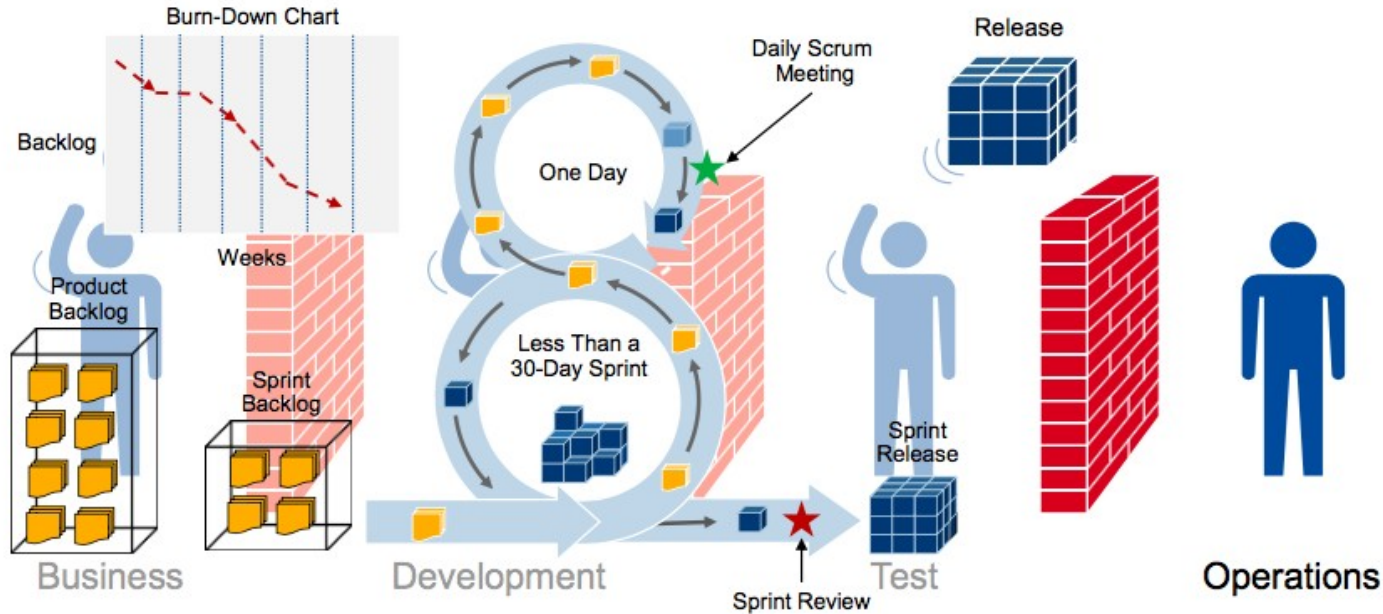
16 © 2015 Gartner, Inc. and/or its affiliates. All rights reserved.

Gartner

Source: “Town Hall: Agile in the Enterprise,” Mike West, Nathan Wilson, Thomas Murphy, Dec 2015, Gartner AADI US conference; [“Pair programming – you’ll never guess what happens next.”](#) Cote, Oct 2016.

Pivotal

Agile Adoption Still Leaves One Remaining Barrier



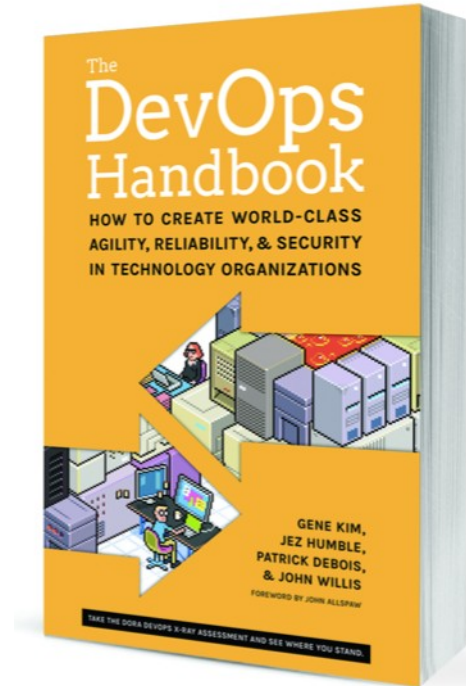
DevOps, the word defines itself

(1.)

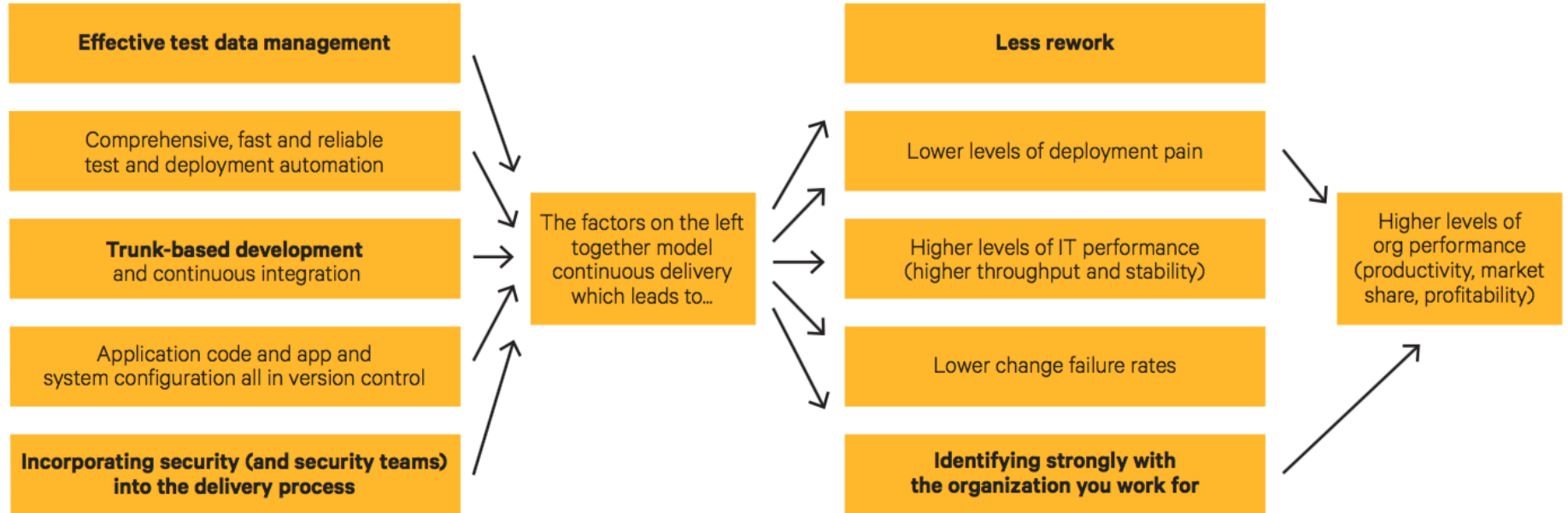
DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.

(2.)

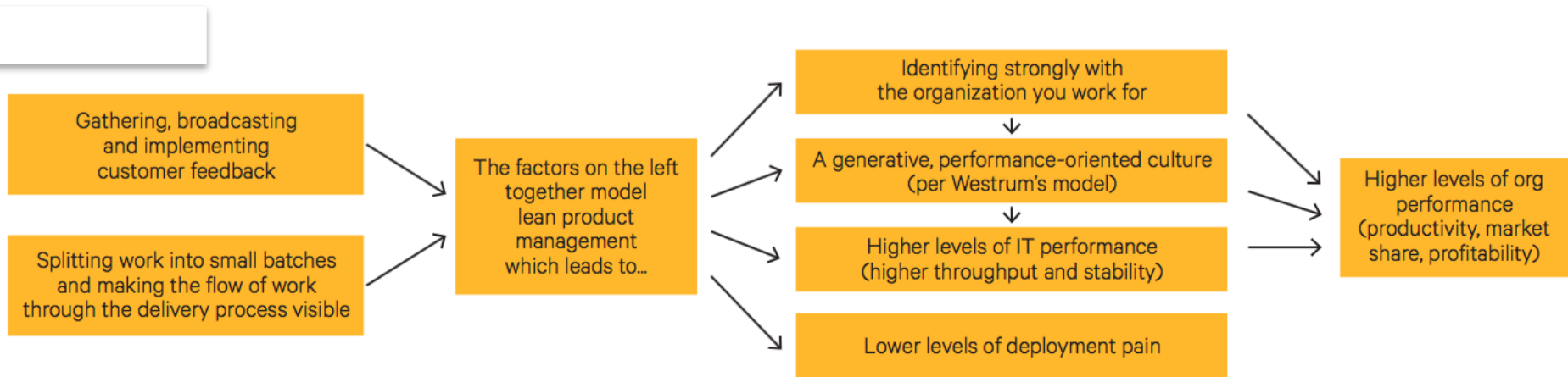
DevOps is also characterized by operations staff making use of many of the same techniques as developers for their systems work.



Goals, practices, and process are more durable



Goals, practices, and process are more durable





Scaling the teams

Balanced teams led by a product owner drive the apps

“You’re not spending money on huge IT contracts or huge teams of people, so a team of 12 might be replacing a team of 100. And you’re not building features that no one wants and no one uses and you’re not wasting time duplicating.”

Ben Terrett, former head of design at the UK GDS



Source: “[Why Britain banned mobile apps; Interview with Ben Terrett, former design chief at the GDS](#),” GovInsider, June 2016. See also “[From Aristotle to Ringelmann](#),” June 2016. Picture from [Pivotal SafeMeds scrapbook](#); “[Roles and Responsibilities for DevOps and Agile Teams](#),” Coté, May 2016; Comcast’s [Christopher Tretina at SP1 2016](#).

“But what about my team?”

- Enterprise architects
- PMO
- QA
- Operations
- Design
- Business analysts

- How do they deliver value to production?
- Slim matrixing to *enable* teams
- Maybe you're not helping
- ...perhaps join the team!

Reverse-quarantine: from 20% to 90% productivity




Allstate

You're in good hands.

CompoZed Labs

Sources: ["Allstate / CompoZed Labs, One Year Later,"](#) Doug Safford, Sep. 2016; ["Building a Brand Around a Technology and Cultural Transformation,"](#) Matt Curry, Sep. 2016; ["Cull your rules and regulations, or be frozen by them,"](#) Coté, Feb 2017.



One weird trick!

(Pairing)

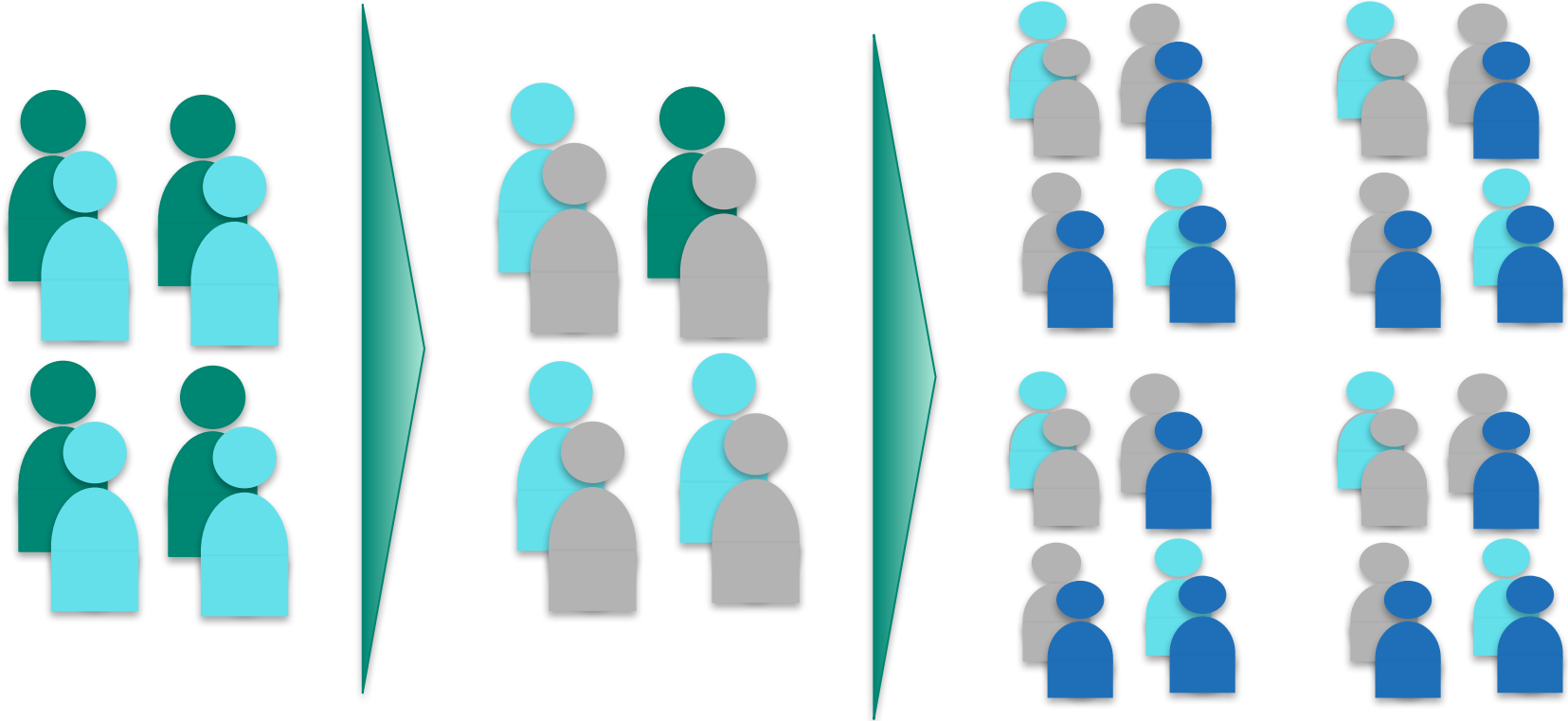
Pairing: across roles, geographies, rotating



Pairing: it stinks, but it works

- Rotating pairs diffuses knowledge, lowers risk
- Removes time used to “sync”
- Shrinks design and QA review time
- Beyond developers: QA, design, product owners

Pairing to spread skills & cloud native culture



Year 0

Year 2

A dimly lit, modern office space with many employees working at computers. The word "Management" is overlaid in large white text. The office has a high ceiling with exposed pipes and lights. There are several desks with multiple monitors, and people are seen working in the background. A staircase is visible in the distance.

Management



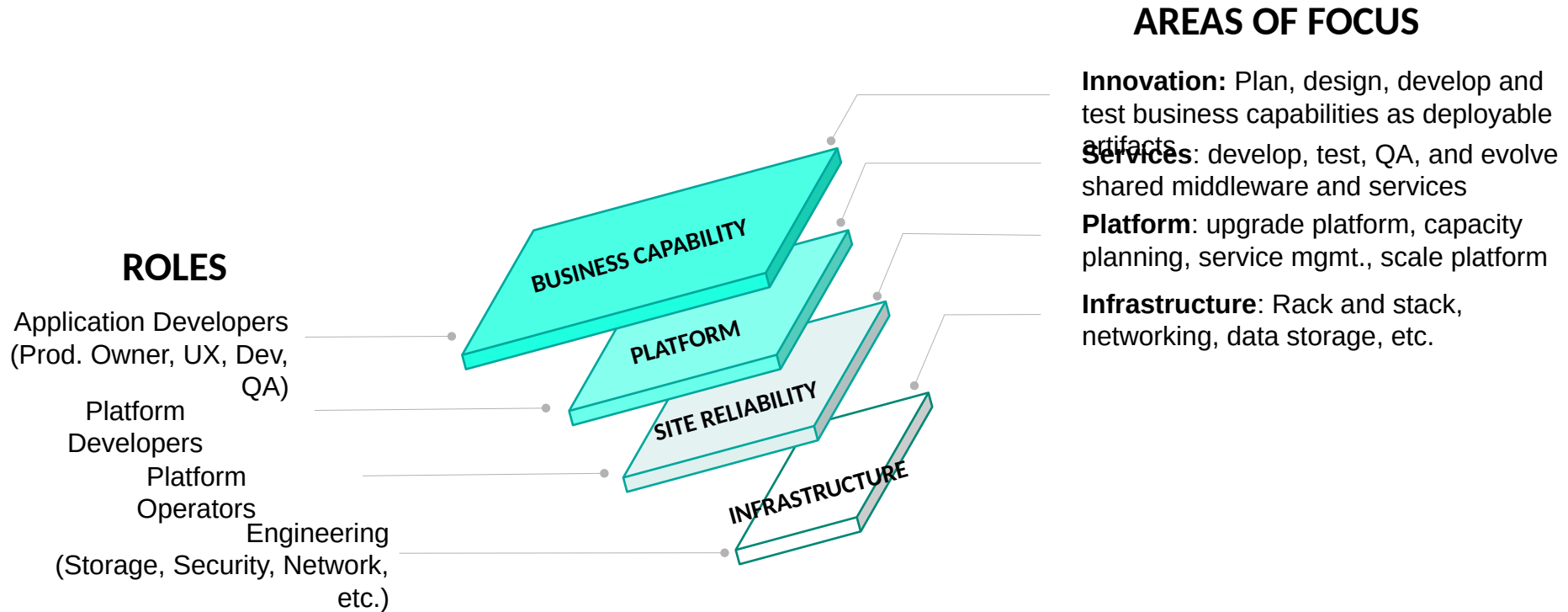
Management will be the first to fail



“I can’t tell you what having a leader stand-up in front of an organization with a hoodie and t-shirt does to cultural change. It all the sudden makes it OK for everyone within that organization to participate in change.”

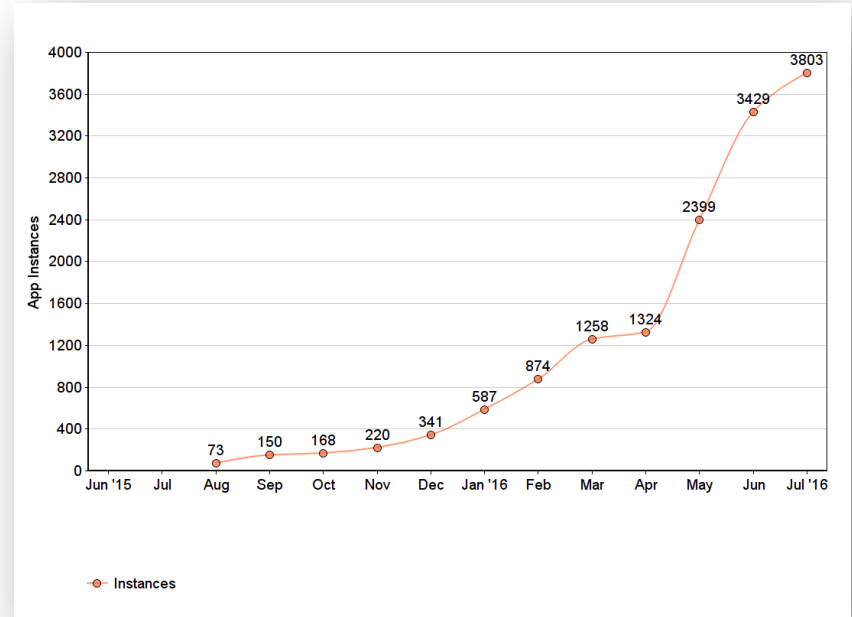
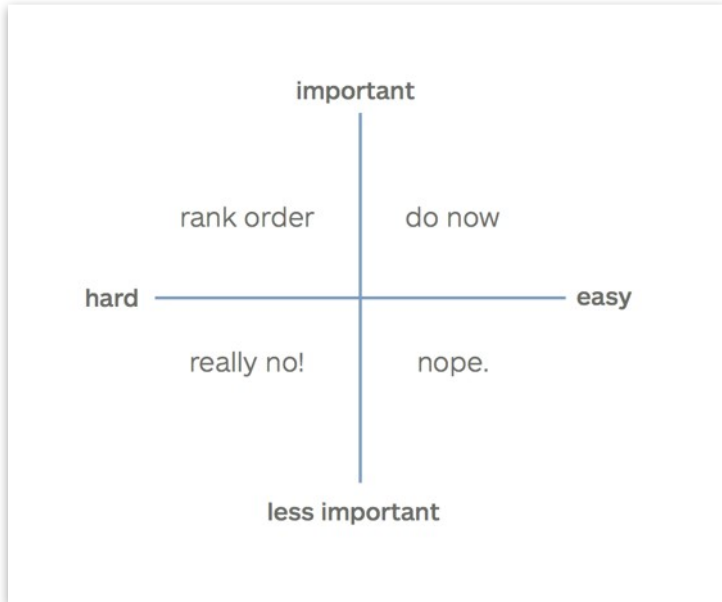
-[Matt Curry, Allstate](#)

The emerging cloud native organization shape



Ramping up: “Pilot low-risk apps, and ramp-up”

Picking your initial apps, according to Comcast, Humana, Ford, & Home Depot



Sources: [Home Depot meetup, Oct 2015](#); [Humana at CF Summit 2015](#); EU payday loan company; Pivotal Labs on large auto company; [“Getting started.”](#) Coté, Oct 2016; [Comcast’s Christopher Tretina at SP1 2016](#); [“Cloud-Native at Home Depot, With Tony McCulley.”](#) Number of AI’s equates to ~130 apps composed on ~900 services. See also a general pattern [applied to analytics from BCG, July 2016](#).

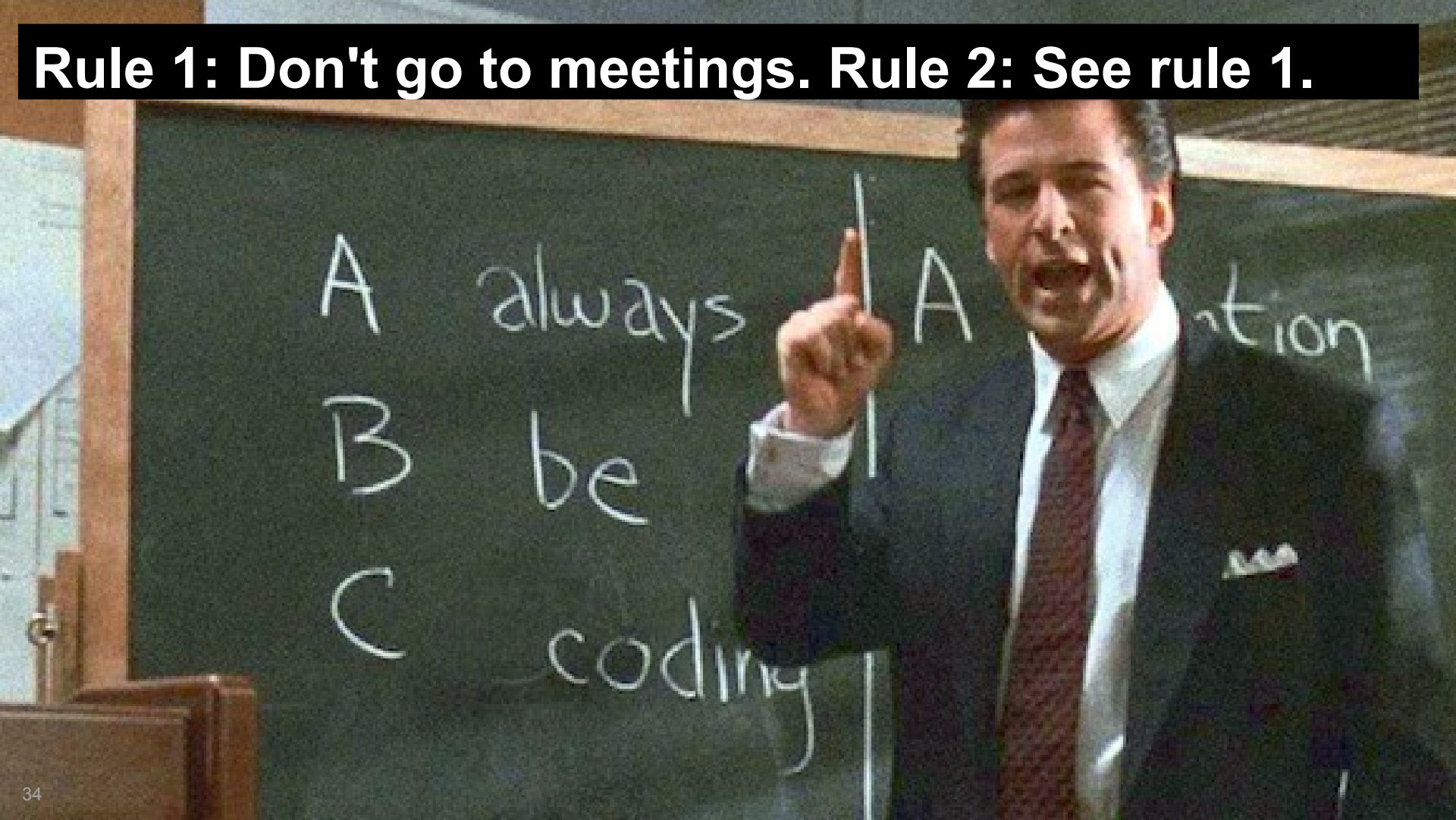
Managing the change: pace yourself

- Liberty Mutual 10 (simple) apps in 10 weeks
- Allstate 16 apps in a year
- THD ~130 apps in a year
- Auto manufacture ~115 after two years
- BUT! If you don't start, you'll suffer analysis paralysis.

Internal marketing: everything, frequently



Rule 1: Don't go to meetings. Rule 2: See rule 1.

A man in a dark suit and tie stands in front of a chalkboard, pointing his right index finger upwards. The chalkboard has handwritten text in white chalk. The text is arranged in three rows: 'A always', 'B be', and 'C coding'. To the right of the man, the word 'A' is written vertically, and the word 'ation' is written horizontally, partially obscured by the man's arm.

A always
B be
C coding

A
ation

Rule 1: Don't go to meetings. Rule 2: See rule 1.

- Meetings to do: stand-up, start, end
- “Don't go to meetings” meeting invites
- Pairing, end-to-end involvement reduce “syncing”
- Allstate: from 20% coding to 90%

Dealing with legacy: the stuff that makes all the money

Quarantine

- Low impact applications
- Lacks good testing, risky to change
- Change resistant priorities



Migration Strategies

- Virtualize & re-platform to control costs and maximize management
- API gateways combined with the strangler pattern
- Portfolio management (cf. The Three Horizons)

Sources: ["DevOps isn't just about the new: It's about cleaning up the old, too,"](#) Coté, April 2016; [The Cloud Native Journey](#), Coté, 2016; [Es cape Velocity](#), Geoffrey Moore, Sep 2011 and ["To Succeed in the Long Term, Focus on the Middle Term,"](#) Geoffrey Moore, August 2007.

A dark, semi-transparent image of a modern office with people working at computers. The office is filled with desks, monitors, and people. The text 'Controls: Compliance & Security' is overlaid in white.

Controls: Compliance & Security

Over time, tools that helped us scale are now obstacles



A pair of silver-rimmed glasses is resting on a stack of papers and a laptop keyboard. The background is a soft-focus office setting with a yellow sticky note visible at the top. A semi-transparent grey box is overlaid on the center of the image, containing text.

“You can type anything you want in a Word document!” [Mark Ardito, HCSC](#)

“You can type anything you want in a Word document!”

- You can likely automate the 3 ring binder
- Compliance is often a feature, treat it as such
- Include auditors on the team
- Give auditors access to the raw systems, “go see”
- Ruthlessly cull your governance & auditors
- It will *always* be a battle – staff up

Small Batches Reduce Risk & Increase Controls

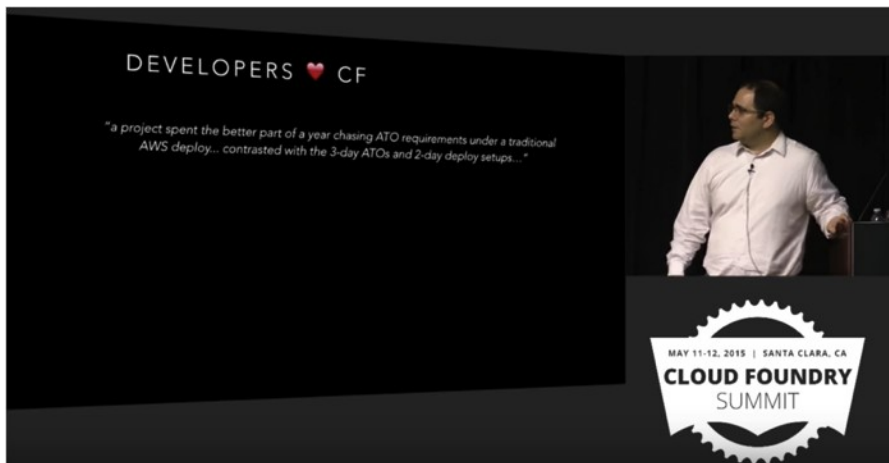
Five risks that small batches reduce

1. **Bug swarms** – less software leads to faster fixing.
2. **Useless software** – don't wait 6+ months to find out, find out every 24 hours.
3. **Stymied Innovation** – daily opportunities to learn and innovate.
4. **Budget overruns** – check in daily on ROI, projections. Shut it down early and re-try if needed.
5. **Schedule elongation** – if you ship every day, guaranteed, how can you ever be late?

"There have obviously been culture shocks. What is more interesting though is that the teams that tend to have the worst culture shock are not those typical teams that you might think of, audit or compliance. In fact, **if you're able to successfully communicate to them what you're doing, DevOps and all of the associated practices seem like common sense. [Auditors] say, 'Why weren't we doing this before?'**"

*["DevOps Enterprise Adoption at E*TRADE,"](#)
InfoQ/DOES, Jan 2016.*

Baked-in: cloud-native has an abundant controls



18F/cloud.gov
ATO reduction from
9-14 months to 2-3 days

Schemas

Example Project Organizations

```
data
├── certifications
│   ├── FedRAMP-low.yaml
│   ├── FedRAMP-med.yaml
│   └── LATO.yaml
├── components
│   └── 18F
│       ├── system.yaml
│       ├── AC_Policy
│       └── component.yaml
└── standards
    └── NIST-800-53.yaml
```

Standards Documentation

```
# nist-800-53.yaml
standards:
  C-2:
    name: User Access
    description: There is an affordance for managing access by...

# PCI.yaml
standards:
  Regulation-6:
    name: User Access PCI
    description: There is an affordance
```

Certifications

```
# Fisma.yaml
standards:
  NIST-800-53:
    C-2:
    C-3:
  PCI:
    6:
```

 OpenControl

Source: [“Deployments We Can Believe In,”](#) Diego Lapiduz, 18F, June 2015; see also [“Barriers to DevOps in Government,”](#) Coté, Oct 2015; [“OpenControl Overview,”](#) Joshua McKenty, March 2016. [“Running Cloud Foundry in a Compliance and Security Focused Environment,”](#) Diego Lapiduz and Bret A. Mogilefsky, 18F, May 2016; [baking in controls](#), from Cornelia Davis, Aug. 2016.

Security – reduce risk by going faster & automating, & acting

- Authentication/UAA
- RBAC
- BOSH enforcement
- Repair
- Repave
- Rotate
- Encryption, data, network
- Isolation segments
- Zero-trust model
- Trusted auditing a-plenty
- Distributed tracing & microservices ops

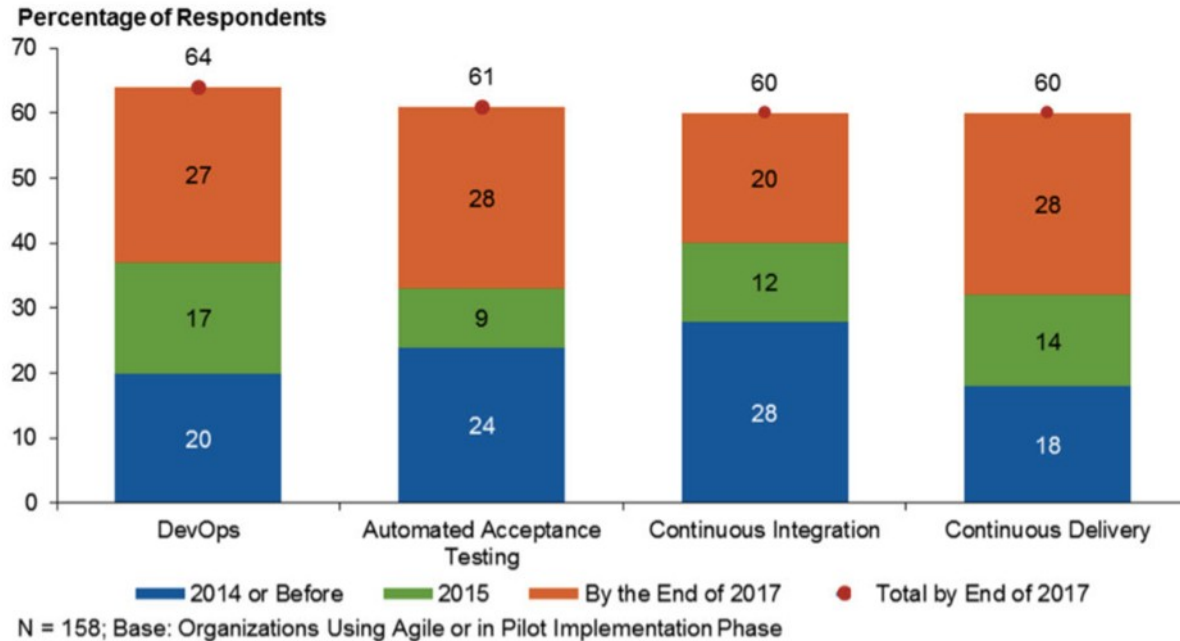
Source: [“The Three Rs of Enterprise Security: Rotate, Repave, and Repair.”](#) Justin Smith, April 2016; [“Understanding Cloud Foundry Security.”](#) Pivotal docs;



Tools

(They do matter)

60% of organizations still don't have CI in place, only ~15-25% have full CD



“I think that’s one of the biggest things that needs to be built up over time when you’re talking about deep ingrained cultures around managing risk by slowing everything down. To speed things up you need to have trust and complete transparency through that process which you get through enforcing continuous delivery pipelines.”

[Dave Ehringer, Liberty Mutual](#)

Source: ["A Guidance Framework for Continuous Integration: The Continuous Delivery 'Heartbeat,'"](#) William R Holz, Gartner, Jan 2016. See also ["CI/CD from a donkey perspective"](#) for a more depressing perspective, and [Gary Gruver's books for the importance of CI](#).

Pipeline Risks & Pitfalls



Brittle Scripts & Slow, Broken Builds

CI Making Things Worse Before They Get Better

Team Members Who Work Around Governance

Stakeholders Who Lose Confidence in Testing Automation

New Platforms That Don't Support Automated Testing

Sources: "[A Guidance Framework for Continuous Integration: The Continuous Delivery 'Heartbeat,'](#)" William R Holz, Gartner, Jan 2016; "[Unwinding Platform Complexity with Concourse,](#)" Matt Curry & Alan Moran (Allstate), Aug 2016; "[Pipelines,](#)" Côté & Matt Curry, *Côté Show*



concourse

- pending
- started
- succeeded
- failed
- errored
- aborted
- paused

© Matt Curry & Alan Moran, 2016

Sources: [“Unwinding Platform Complexity with Concourse,”](#) Matt Curry & Alan Moran (Allstate), Aug 2016; [“Pipelines,”](#) Coté & Matt Curry, Coté Show Variety Podcast #34.





You know, your basic, highly scalable, fault tolerant, distributed enterprise application management, integrated middleware, fully operational battle platform.

No big deal, or anything.

The infrastructure promises you need

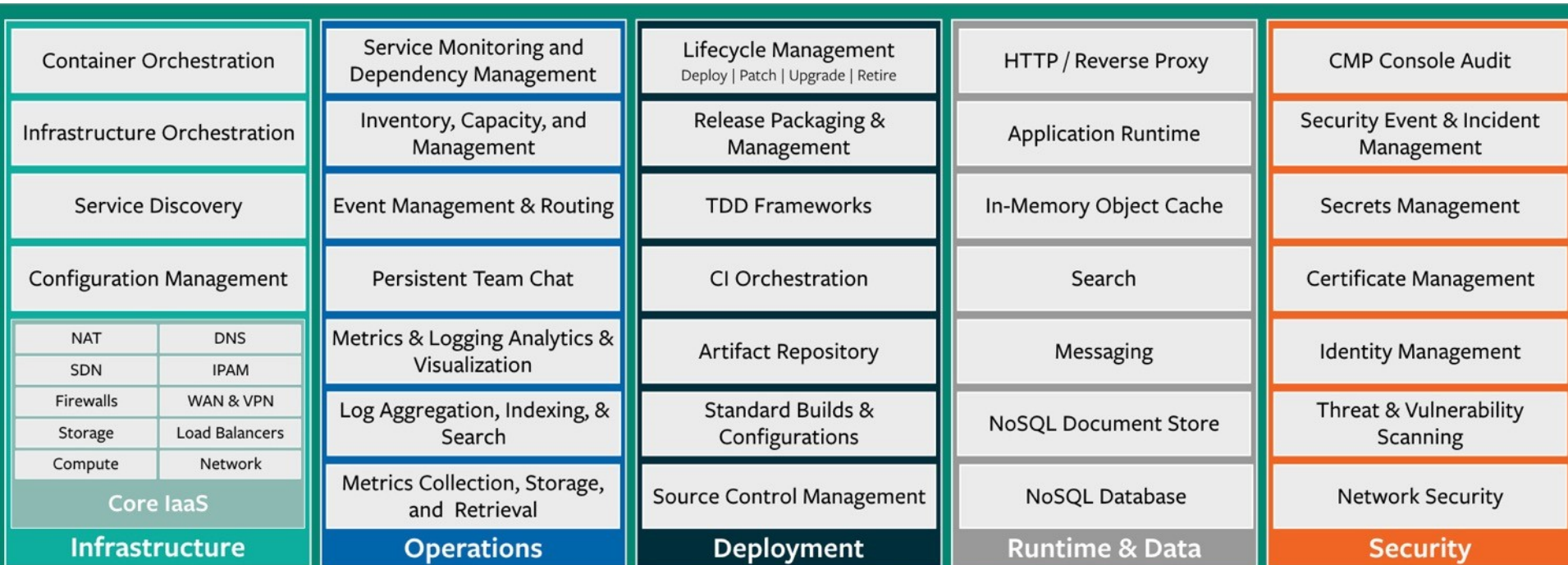
- Canary deployment
- Configuration changes
- Rolling upgrades
- Health checks and remediation
- Scale out/scale in
- Across multiple IaaS



The runtime platform promises you need

- Role based access to resources
- Run code on demand
- Coordinate cross service configurations
- Route public requests
- Read and write persistent data
- Add and remove resources
- Record internal and external events
- Isolate resources and failures
- Measure performance/health
- Detect and determine failure (plan & provoke failure)
- Recover failures
- Work tomorrow

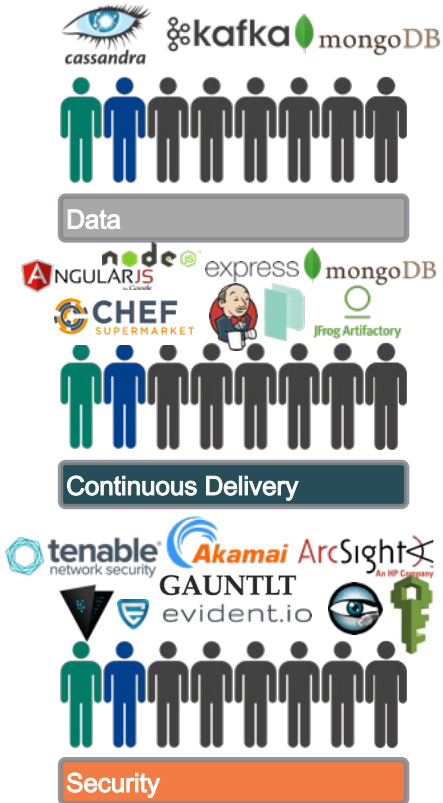
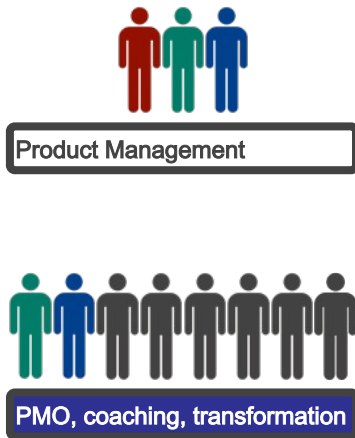
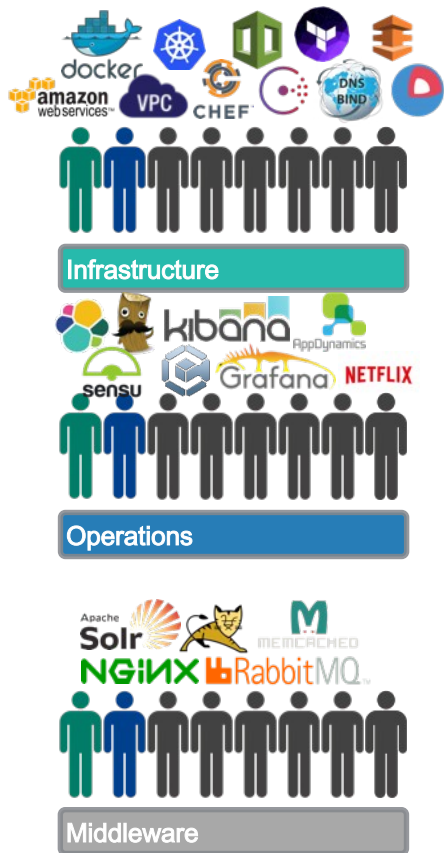
Standardize on a platform: special snow flakes are slow, expensive, & error-prone



Reference Architecture for Cloud-Native Platforms

51 Source: [“The Upside-Down Economics of Building Your Own Platform.”](#) Jared Ruckle and Matt Walburn, 2017.

Schedule 2 years & ~\$14m to build your platform



Notes: 50 people at \$140,000 all in comp. is \$7m/yr. Does not include costs for sriacha cashews & chunky coconut water. See more: [“The Upside-Down Economics of Building Your Own Platform.”](#) Jared Ruckle and Mat Walburn, 2017.



Pivotal Cloud Foundry®

MODERN CLOUD NATIVE PLATFORM

PIVOTAL CLOUD FOUNDRY APPLICATION RUNTIME

DYNAMIC ROUTE SERVICES / API MANAGEMENT

YOUR
APPLICATIONS

APP MICROSERVICES
TECHNOLOGY

Spring Boot

Steeltoe

Spring Cloud
Services

DATA MICROSERVICES
TECHNOLOGY

Spring
Cloud Data
Flow

Cloud
Cache

RabbitMQ

MySQL

PLATFORM

Elastic Runtime

Concourse

App
Autoscaler

PCF Metrics

CredHub

Orgs, Spaces,
Roles and
Permissions

SERVICE
BROKER
API

PIVOTAL CLOUD FOUNDRY BOSH

EMBEDDED OS

Windows

Linux

CONTAINER ORCHESTRATION

CLOUD ORCHESTRATION

MULTI
CLOUD

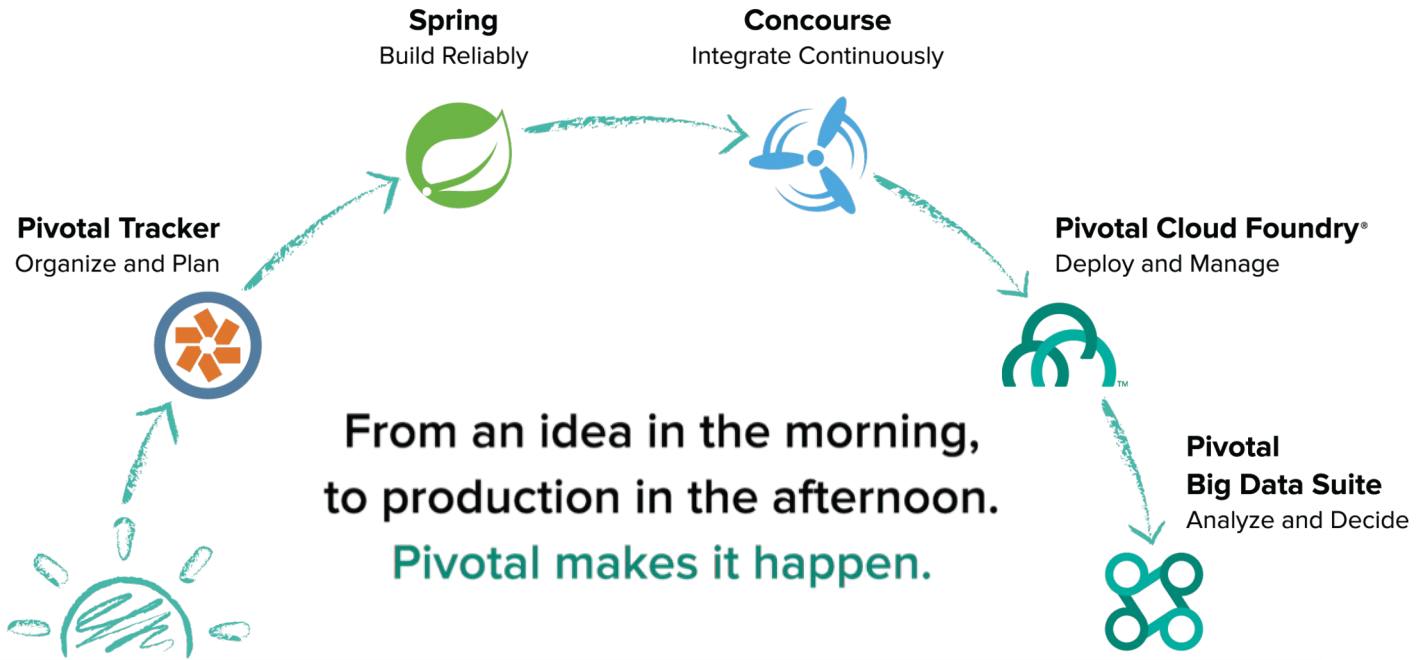
Amazon
Web Services

Microsoft
Azure

Google
Cloud
Platform

Open Stack

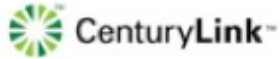
VMWare



Pivotal helps organizations do better software



Mercedes-Benz



*What's in the
box?*

*Digital
transformation.*



Crafting Your Cloud-Native Strategy

by Michael Coté

Pivotal

EBOOK

Table of Contents

Crafting Your Cloud-Native Strategy	1
IT's Role in the Era of Transient Advantage	3
The Goals of Cloud-Native and What the Term Means	3
"Survival is Optional. No One has to Change."	4
The Era of Transient Advantage Demands Cloud-Native IT	5
PART 1:	
Changing Your Process and Organization	6
Small Batch Thinking	7
Become a Learning, not a Status-Checking Organization	8
Moving from failure management to risk management	8
Case Study: "What Have You Learned?"	10
Shift to User-Centric Design	10
Case Study: No One Wants to Call the IRS	11
How much do I owe the IRS?	11
DevOps as the Overarching Process	12
Put it All Together to Enable Small Batches	14
The Shape of a Cloud-Native Organization	14
Product: Balanced Teams	15
PART 2:	
Cloud-Native Transformation—Doing the Work	23
Creating the Game	23
Creating a Continuous Learning Organization	25
Building Trust and Defibrillating Staid Corporate Culture	25
The Three Finance Questions You Meet in Drab Conference Rooms	28
Success is the Best ROI	31
Case Study: IRS	31
Case Study: "It Was Way Beyond What We Needed to Even be Doing"	33
Risk Management with Small Batches	33
Security: Same Old Story, New Spiffy Tools	34
Revitalizing Legacy Code	38
Living with Legacy	39
Avoiding Legacy Pitfalls	41
Case Study: Avoiding Portfolio Paralysis Analysis	42
Cloud Platform Reference Architecture	43
Building Your Own Cloud Platform is Probably a Bad Idea	44
For Just \$14m, You Too, Can Have Your Very Own Platform in Two Years	45
Pivotal Cloud Foundry	46
Picking Projects by Portfolio Pondering	47
Planning Out the Initial Project	48
Crafting Your Hockey Stick	49

“We are uncovering better ways of developing software by doing it and helping others do it.”

- [The Agile Manifesto](#), 2001

Thanks!

@cote | cote@pivotal.io
More: <https://cote.io/cloud3/>

Pivotal®

SpringOnePlatform.io

**\$100 DISCOUNT CODE:
S1P_EVENT_CFS100**

SpringOne Platform

REGISTER NOW.

December 4–7, 2017

Moscone West, San Francisco

Pivotal.

