Beyond the accidental platform or, success is no accident

August 2015
@cote

Chicago Cloud Foundry Meetup

Pivotal

Conclusions

- The goal of cloud is improve the quality of your custom software, making it the core business enabler
- Technology problems are preventing this, but it's mostly meatware problems
- An integrated, full cloud native platform takes care of the technology problems as best as we can right now
- Meatware issues: management needs to manage; portfolio management to free up resources for innovation; focus on "small" instead of "big"; not enough QA; creating the right organization.

Pivotal

Hello!

- @cote – Director, Technical Marketing at Pivotal for Pivotal Cloud Foundry
- Former industry analyst at 451 Research and RedMonk
- Corporate Strategy & M&A at Dell
- Former software developer

- More: http://cote.io or cote@pivotal.io

Pivotal

Goals, problems

Pivotal

The goal is not “cloud,” rather shifting to using software as your core enabler of business

Release weekly, if not daily

Software continually updated to match evolving business models

IT is the enabler of growth

Software Defined Business

For the business “oh crap” motivations, see “The 3 Horsemen of the Digital Apocalypse” and the first part of of my DevOpsDays talk.

Pivotal

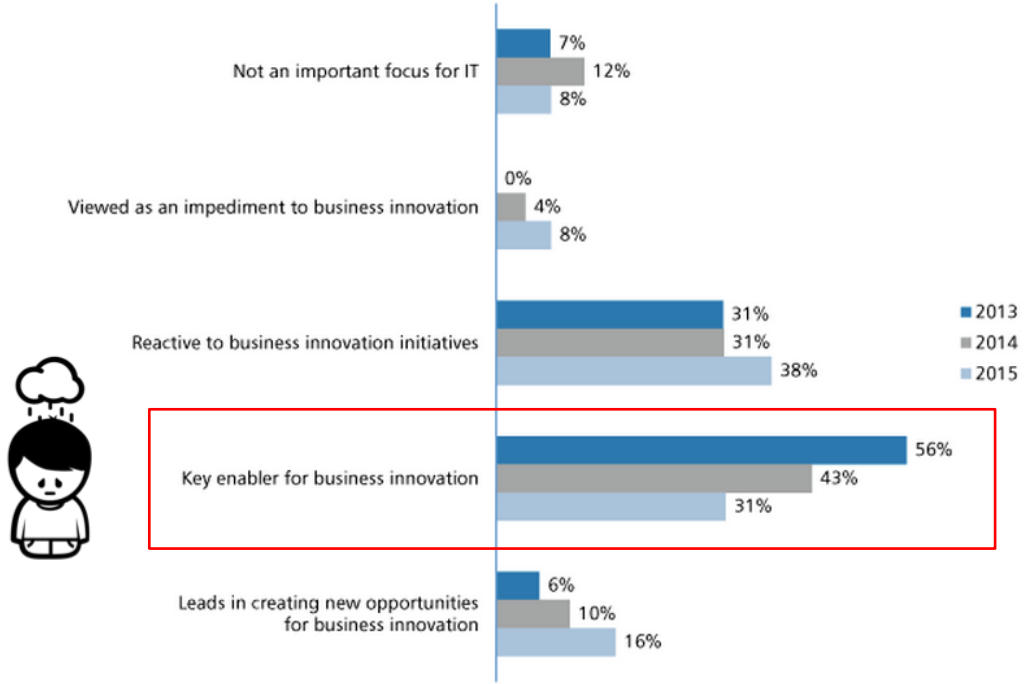
The IT Department of Slow/No

- 6-12 months to release features
- Business struggles to have software match market opportunities
- IT is a cost center

Clipart from Gerd Altmann.

Businesses are suffering from an agility gap

Only 25% of respondents felt that their companies were innovating in agile ways.



What is your IT organization's role in business innovation?

Source: Institute for the Future study, April 2015, n=3,600; Cutter Consortium, July 2015

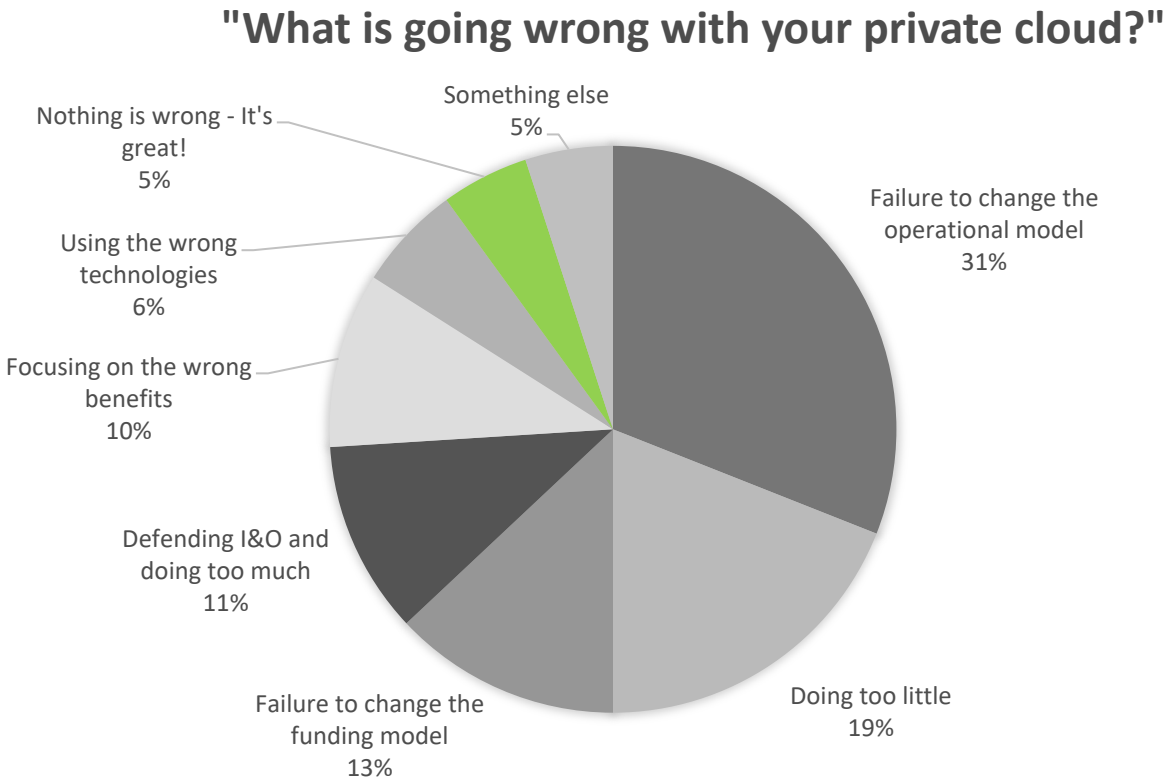
Pivotal

Many problems are in meatware

Source: "Strategy, not Technology, Drives Digital Transformation," 2015 Digital Business Global Executive Study and Research Project, MIT Sloan Management Review & Deloitte University Press, July 2015. n=4,800,conducted in Fall of 2014.

Pivotal

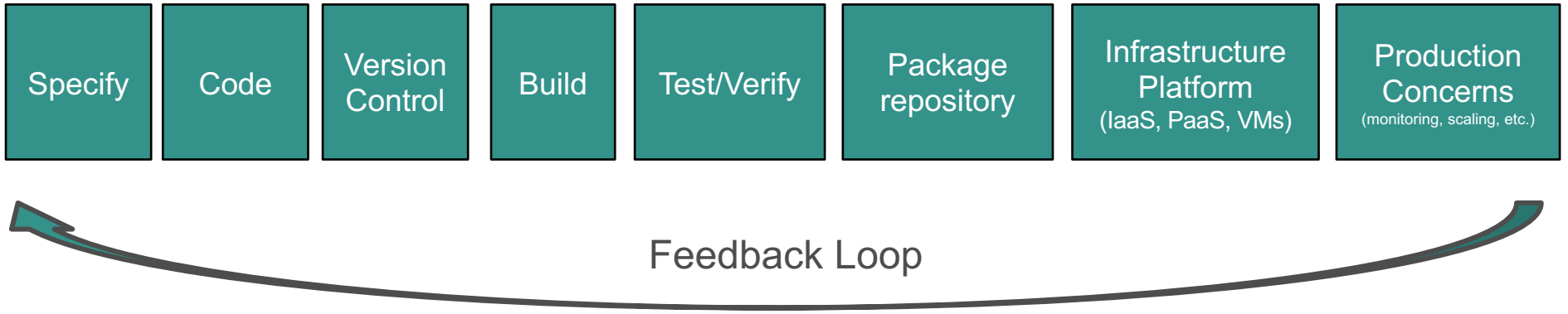
Meatware bleeds into IT



Sources: "Problems Encountered by 95% of Private Clouds," Gartner, Feb 2015.

Pivotal

What “cloud” projects really want is this…



* OK, sure, some of them just want to forklift unchagning applications to drive down costs or dump ops all together and move to SaaS. You got me! But, those writing custom software need this pipeline.

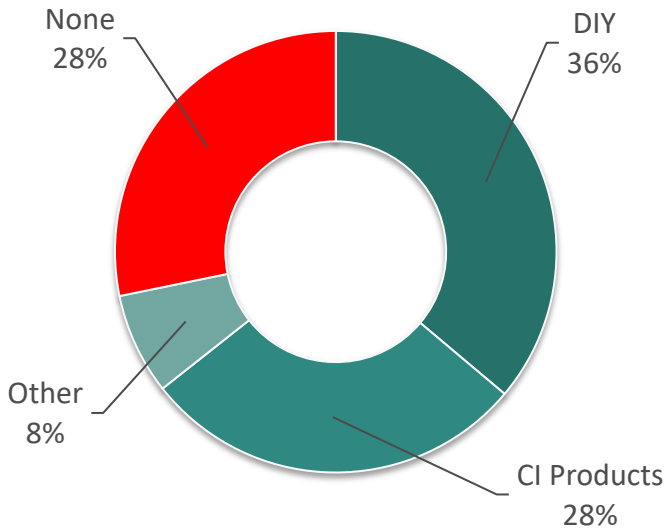
Pivotal

How's it going?

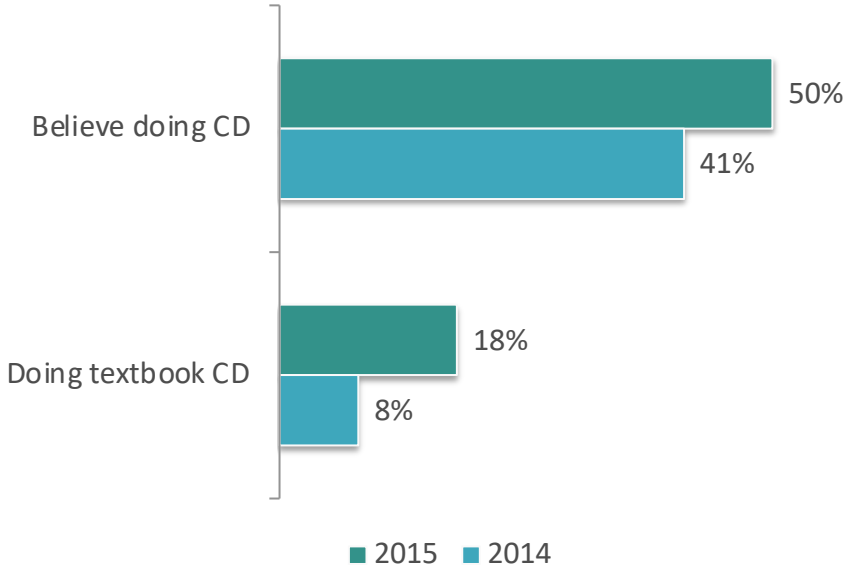
Pivotal

Things are improving, but need an accelerant

What build automaton or CI/CD tools are you using?
(451 Research study, 2014)



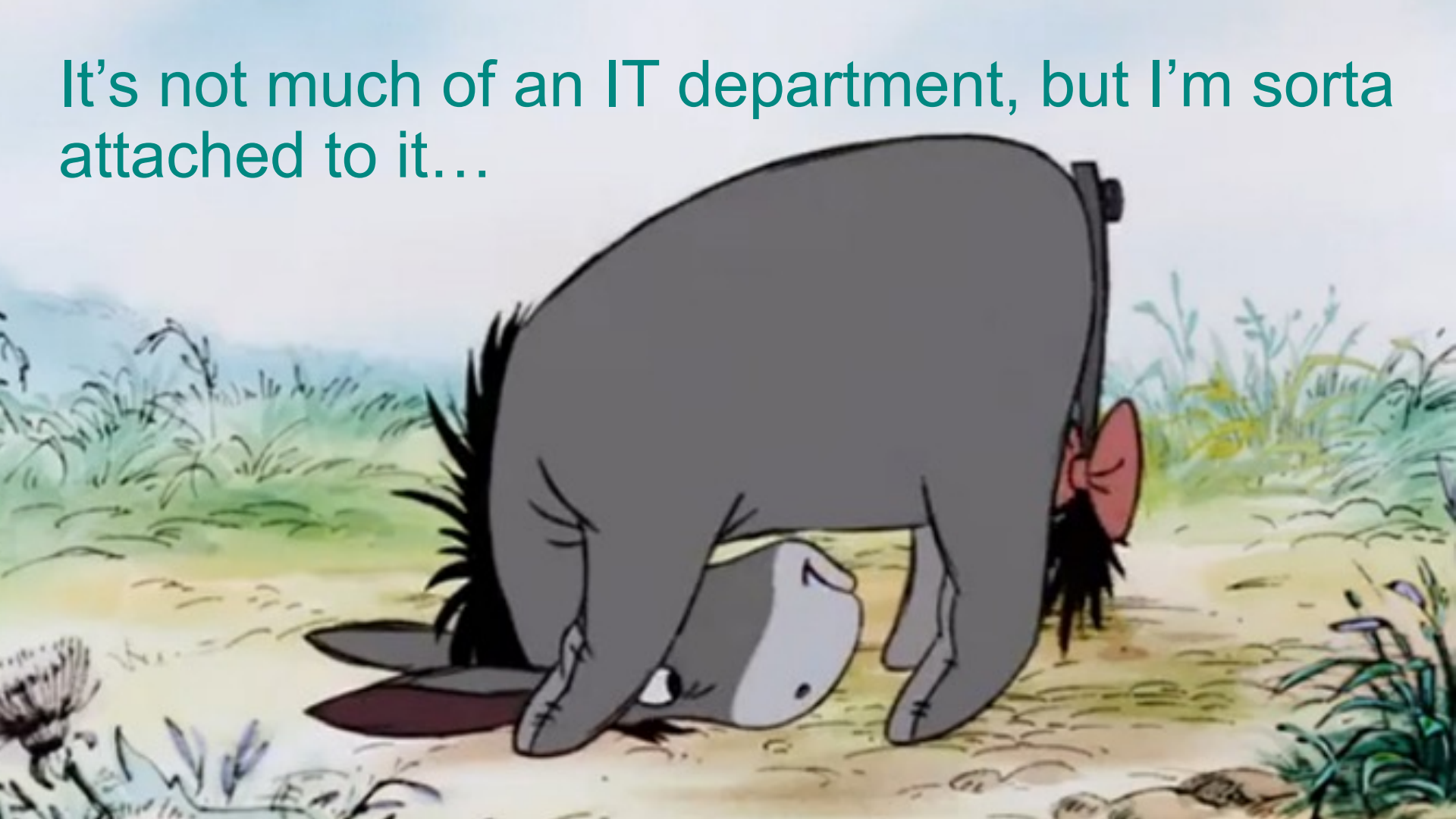
Use of CD is growing
(DZone studies)



Sources: 2014Q1 451 Research DevOps Study, n=201. In second study (n=300), 38% used “build and continuous integration tools”; "DZone's 2014 Guide to Continuous Delivery," n=500; The DZone Guide to Continuous Delivery, Vol. 2," Feb, 2015, n=900.

Pivotal

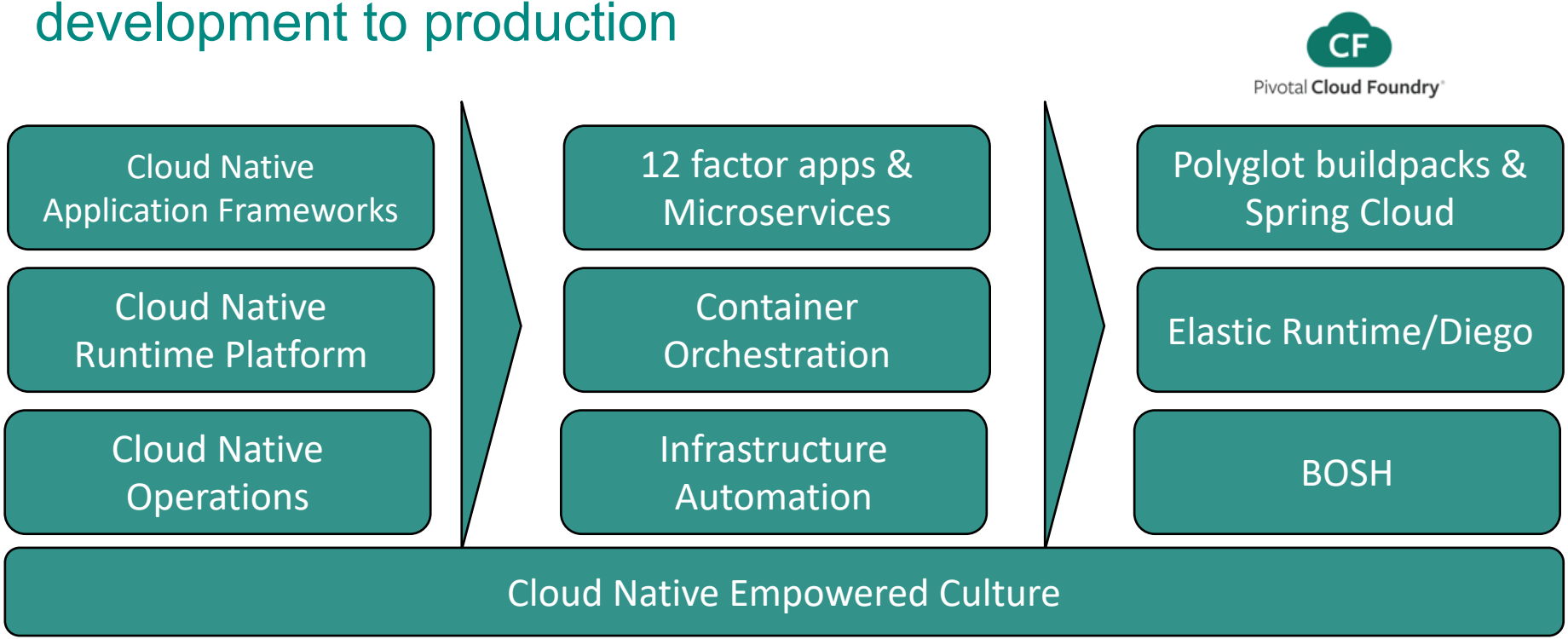
It’s not much of an IT department, but I’m sorta attached to it…



“Cloud Native” addresses the meatware & software problems

Pivotal

A cloud native platform is composed of three layers, that span & support the entire life-cycle of an application from development to production



Source: slides in this section based on "Patterns of Cloud Native Architecture," Agile 2015. See also "The cloud native future" for more discussion.

Pivotal

Cloud Native Application Frameworks

General Principals

- Polyglot programming
- 12 factor contract
- Microservices
- Service oriented
- Full of middleware
- Composable
- Fault tolerant

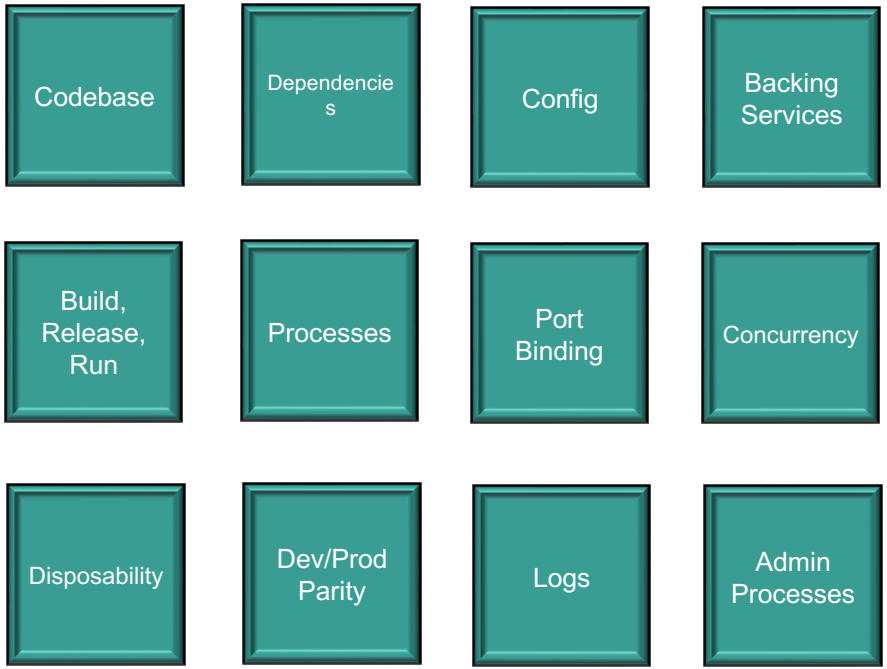
Buildpacks, Containers, & Spring Cloud

- **Java, Ruby, Python, Node, PHP, Go**
- Buildpacks & platform enforces 12 factor style
- Automated or manual container creation & use
- Spring Boot & Cloud for composable patterns, managed microservices
- Configuration service
- Service registry & discovery
- 30+ middleware services like mobile, DB, queues, CI/CD, etc.
- Load balancing
- Circuit breaker
- Micro-proxy
- API gateways

spring NETFLIX OSS Pivotal

Use 12 factor app principles to create cloud ready applications

- A set of best practices for developing and deploying cloud-native software.
- Practices translate into platform features and workflow requirements.



Source: "The Twelve-Factor App."

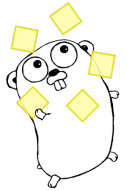
Pivotal

Cloud Native Runtime Platform

General Principals

- Multi-cloud use of containers and VMs
- Manage the create, run, destroy lifecycle
- Predicable resource utilization through constraints
- Process isolation
- Optimized resource utilization through orchestration
- Methods to diagnose & recover from app failure
- Identity, access control, and audit

Pivotal **Elastic Runtime**

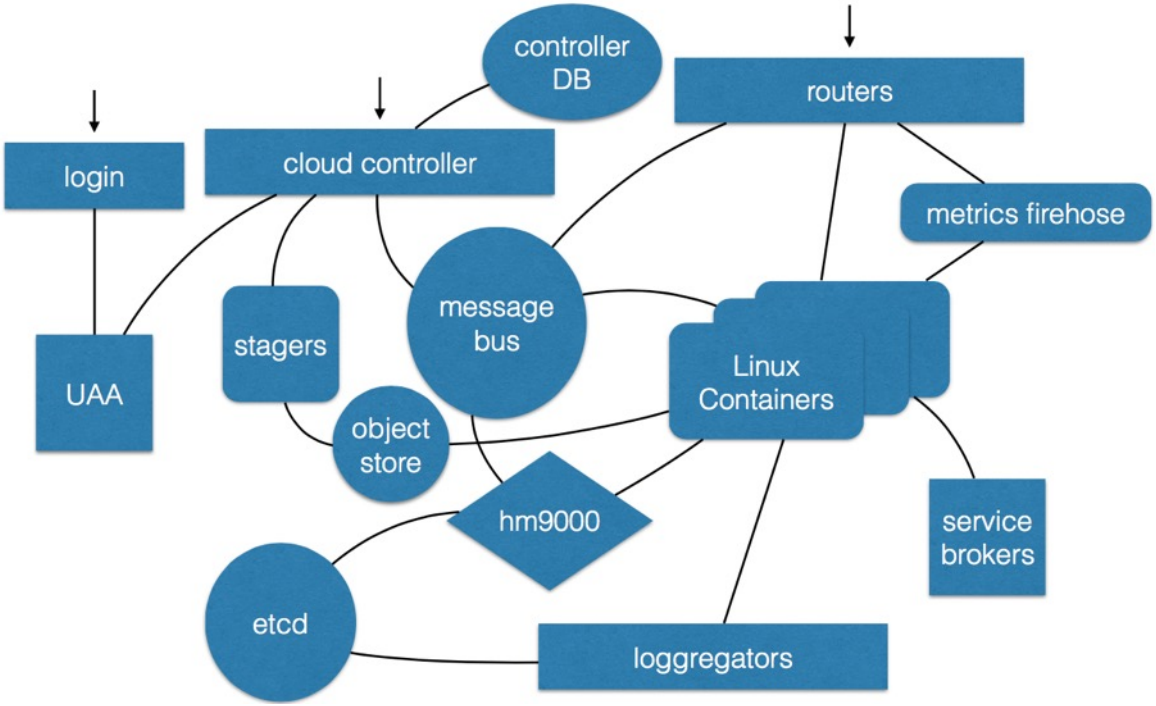


Diego

> Lattice

Pivotal

Elastic Runtime (coming soon: Diego)



* And a couple dozen other services

Pivotal

Cloud Native Operations

General Principals

- Routing & load-balancing
- Automate infrastructure
- API driven
- Enable & automate backing services
- Infrastructure health management, monitoring, recovery

BOSH

- Multi-cloud support
- Clean separation of systems
- Consistent rapid provisioning
- Scale up/scale down
- Built in health monitoring
- Fault remediation
- Canary deployments

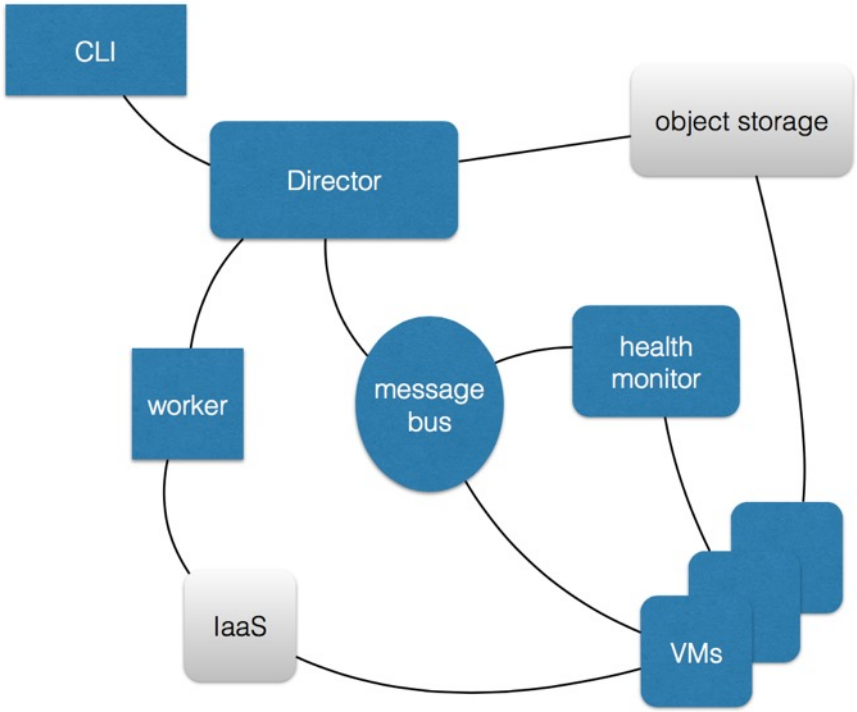
Pivotal

The BOSH Contract

- `current_vm_id`
- `create_stemcell`
- `delete_stemcell`
- `create_vm`
- `delete_vm`
- `has_vm?`
- `reboot_vm`
- `set_vm_metadata`
- `configure_networks`
- `create_disk`
- `delete_disk`
- `attach_disk`
- `snapshot_disk`
- `delete_snapshot`
- `detach_disk`
- `get_disks`

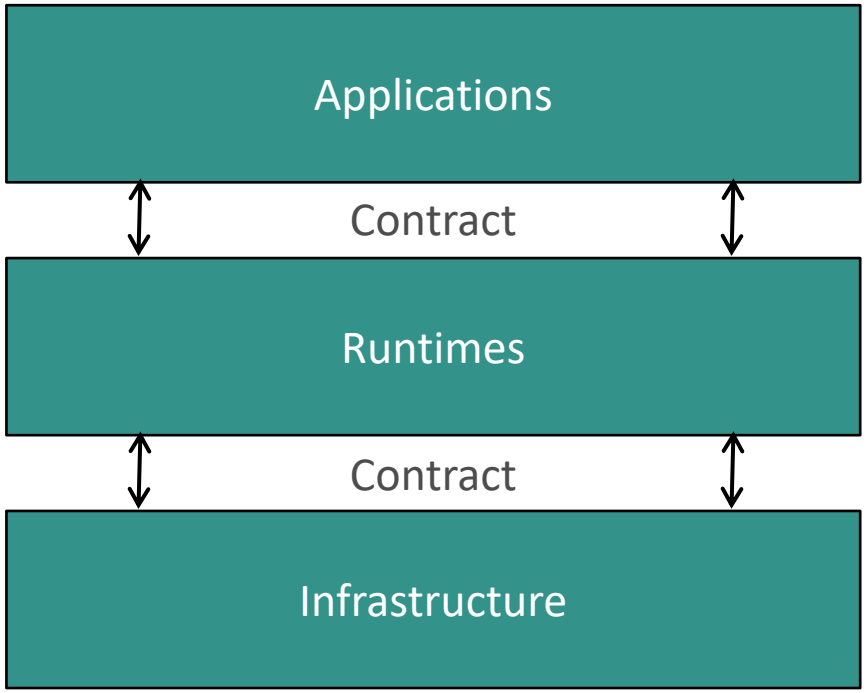
Pivotal

BOSH's moving parts, the really simple edition



Pivotal

The Cloud Native Platform, simplified



spring

NETFLIX OSS

etc.

Pivotal Elastic Runtime

Diego

> Lattice

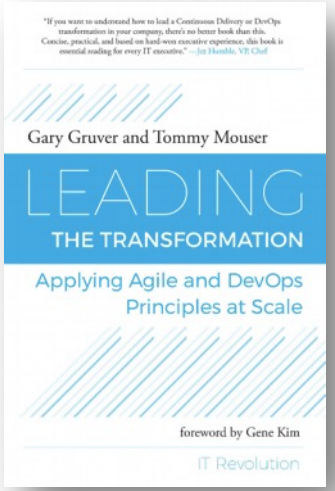
Pivotal

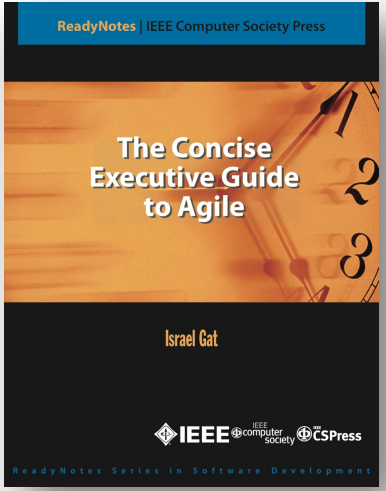
Onto the meatware!

Pivotal

Management creates the game

- Leading change management
- Setting, communicating, tracking goals
- Dramatic organization change, gradually
- E.g.: from autocrat to self-directed teams



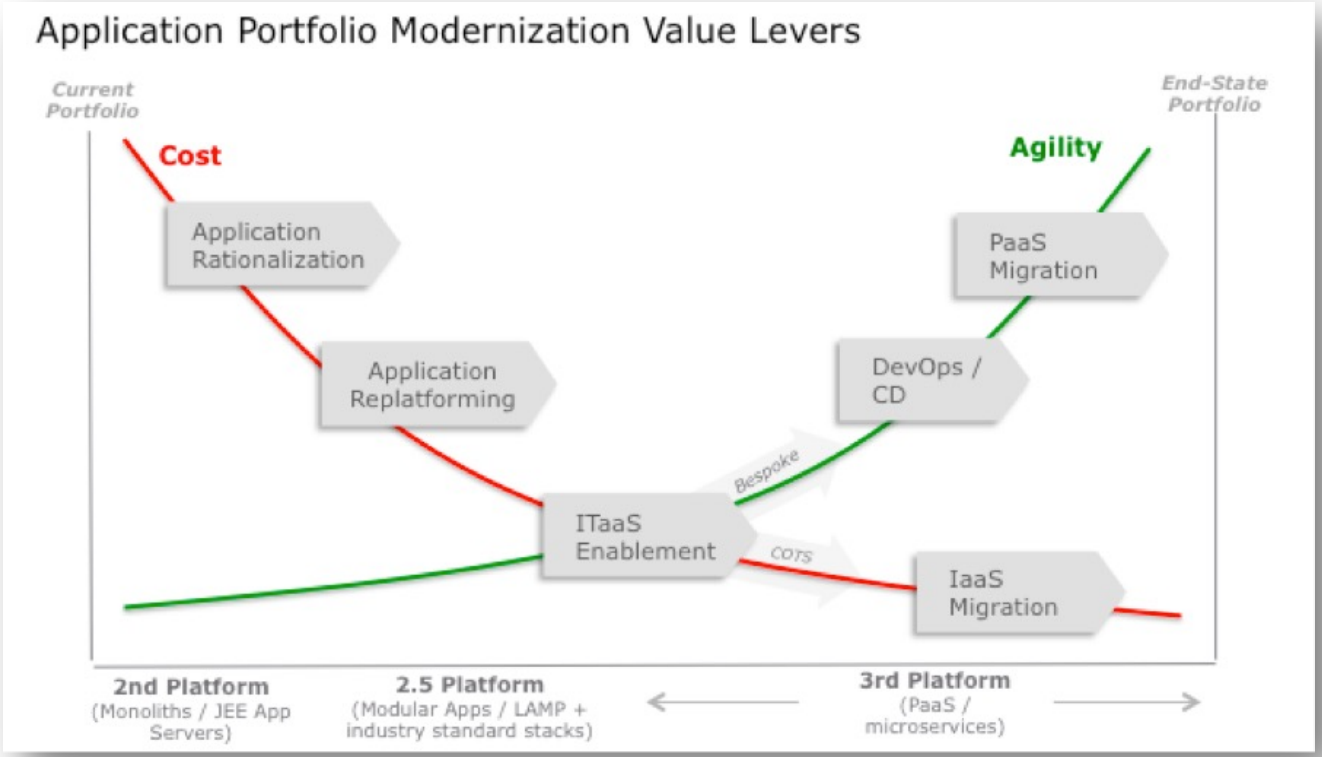


[Pivotal Cloud Native Journey blog series]

Sources: *Leading the Transformation*, 2015; "Management's Job is orchestrating the 'why,'" 2015; *The Concise Executive Guide to Agile*, 2010.

Pivotal

Portfolio management finds “evolutionary” & “revolutionary” apps, balancing resources accordingly



Source: “A Value Framework that Works for Transforming Your Application Portfolio,” June, 2015. For lower level tips, see these two pieces by Josh Kruck, Josh Long, and for WebSphere, Rohit Kelapure.,

Pivotal

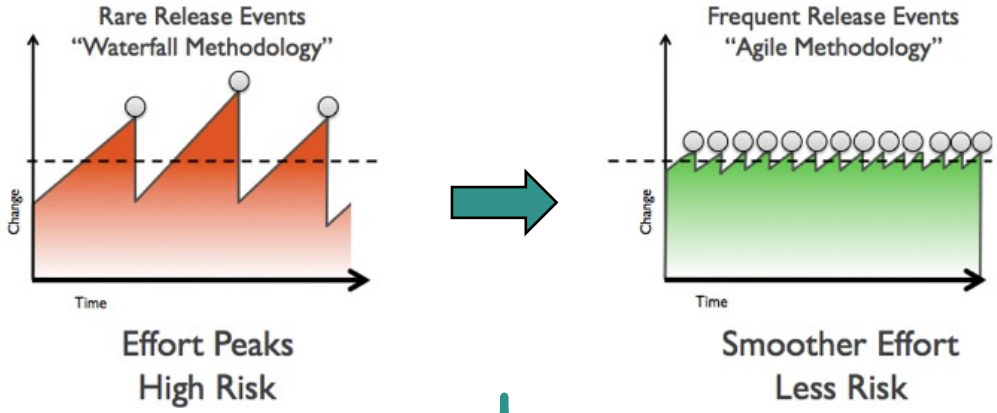
Be successful at a small series of projects

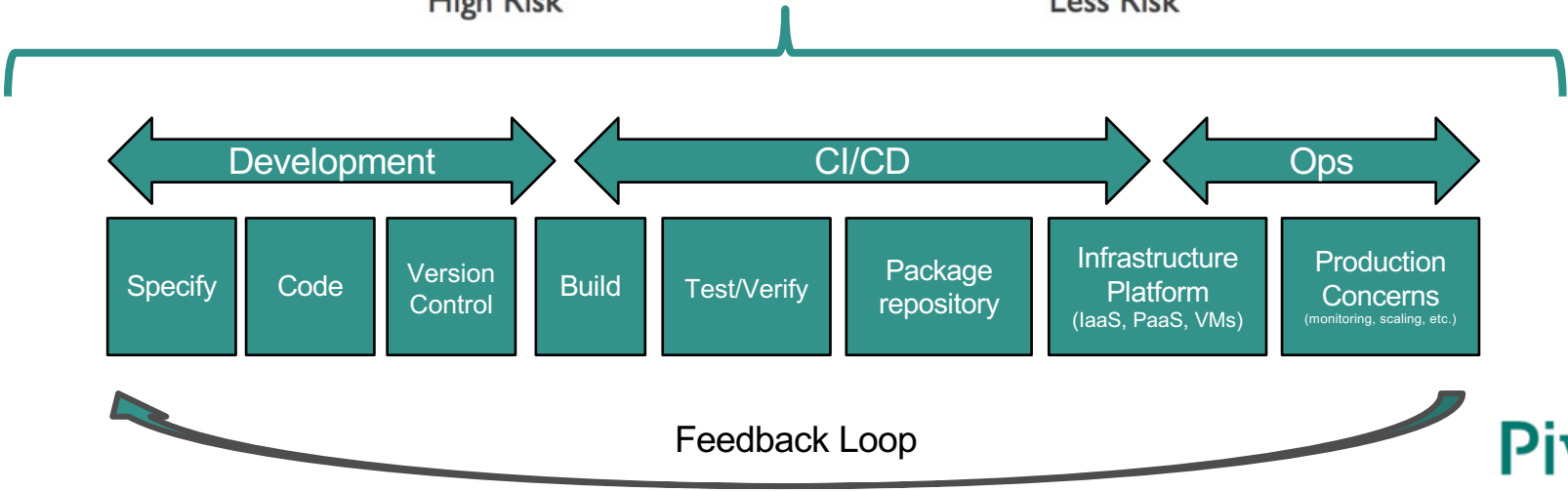
- Vitality drove engagement from 3% to 30%+
- Second project, MyHealth
- Cue Apple Watch app in 5 weeks

Source: Humana keynote, CF Summit 2015. See also BMC Software case study in *The Concise Executive Guide to Agile* and *Cutter Executive Report, Vol. 9, No. 9, 2008*.

Pivotal

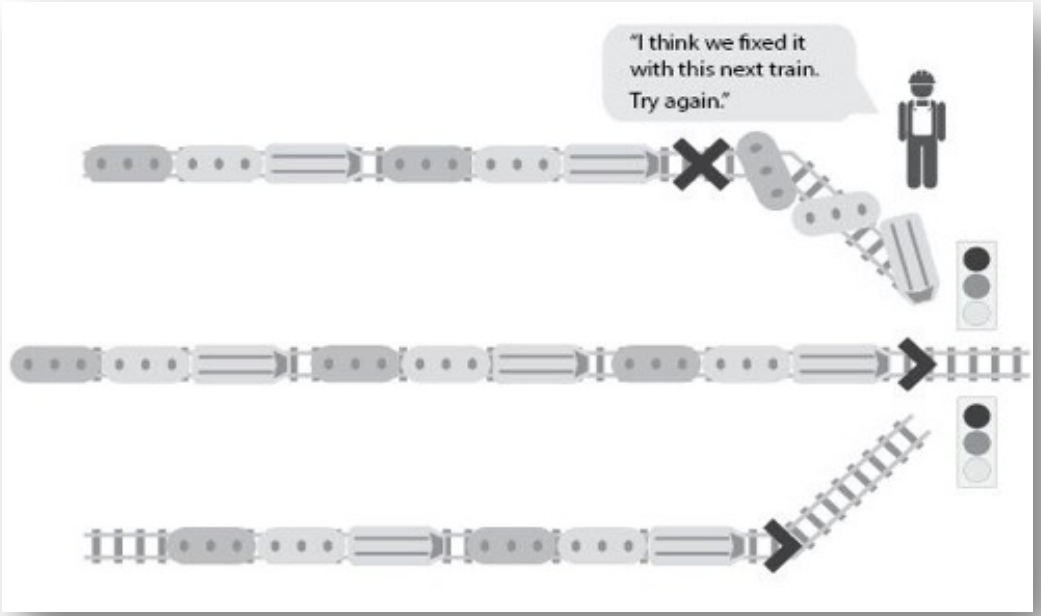
Focus on releasing loosely coupled, small batches



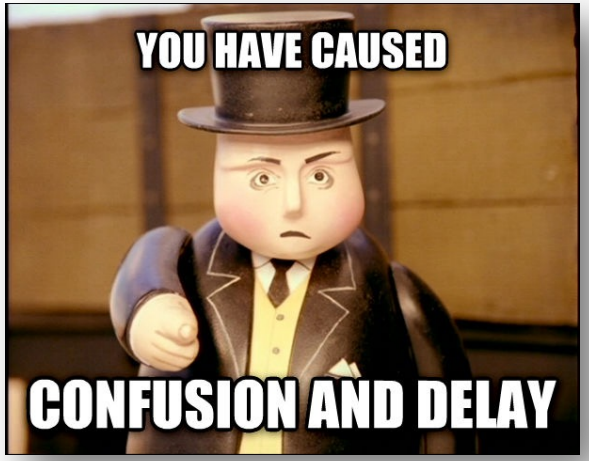


Pivotal

Tight coupling slows down releases, eventually



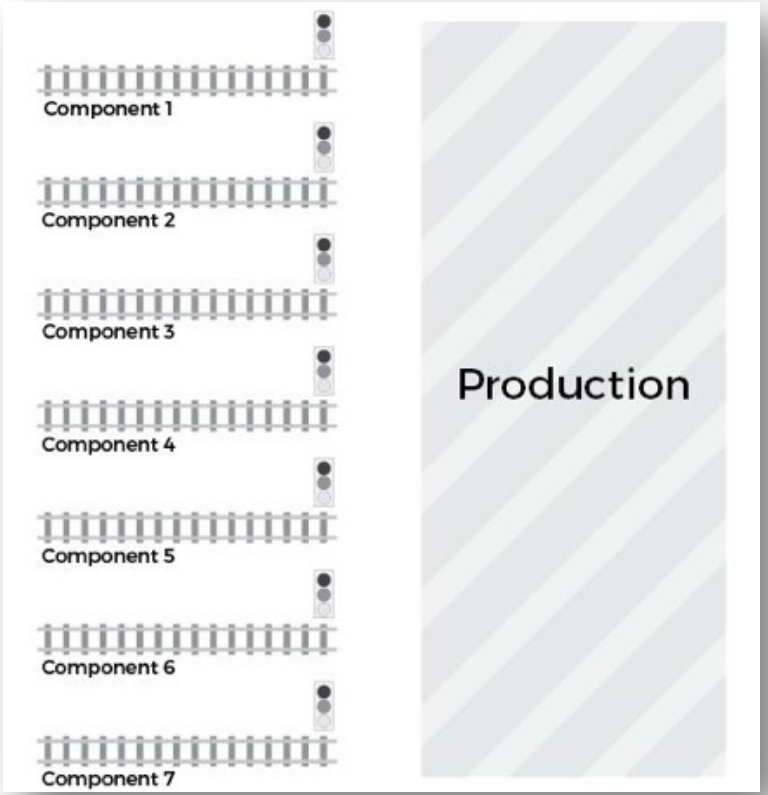
==



Source: diagram from *Leading the Transformation*, 2015.

Pivotal

Managing dependencies in loosely coupled systems is harder, but ensures more frequent releases



- Managed independently
- Not held up by slowest “train”
- Fits cloud native “scale-up” model
- Reduces risk

Source: diagram from *Leading the Transformation*, 2015.

Pivotal

Test all the things, all the the time

- QA - automated testing to avoid technical debt, move fast
- Uptime - testing for resiliency in production
- Design quality - do people actually find your software useful?
- Improvement - testing your process

Pivotal

The emerging cloud native organization model

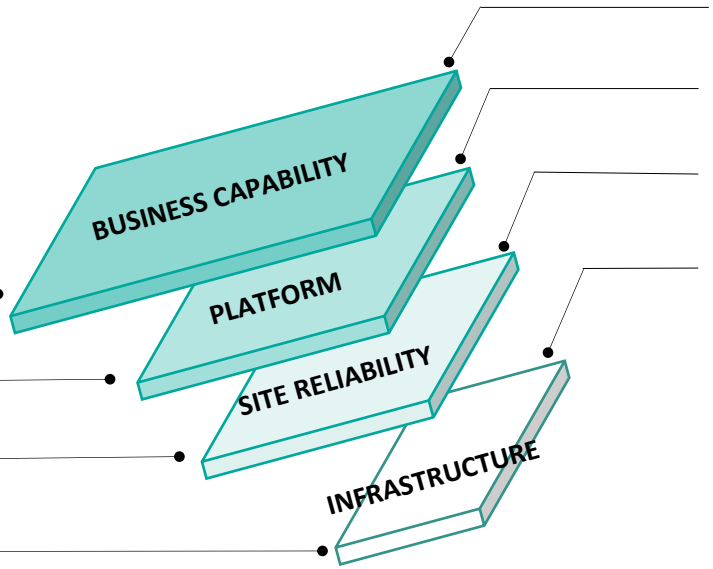
ROLES

Cross-Functional
(Prod. Owner, Dev, QA)

Application Operators

Platform Operators

Engineering
(Storage, Security, Network, etc.)



AREAS OF FOCUS

Innovation: Plan, design, develop and test business capabilities as deployable artifacts

Production Apps: config, deployment, QA, monitoring, scaling

App Platform: upgrade PCF, capacity planning, service mgmt., scale platform

Infra Platform: Rack and stack, networking, data storage, etc.

Source: slide from Pivotal Cloud Foundry Solution

Pivotal

End-state example: SafeMeds

Process & tools that enabled deploying to production every day

Day 1	Day 2	Day 3	Day 4	Days 5-8
- Create dev & prod environments - Code & deploy search story, pulling data from FDA API - User research, persona creation Identify initial user tasks to deliver	- Refine requirements - Wireframes - Rewrite search (API throttling hampers autocomplete) - Create user stories	- Mobile design - Fix & deploy bugs in search - Usability testing with 2 real users - Code & Deploy drug interaction feature - Start on auto-complete	- Team ideates 90 ideas down to 8 - Create validation criteria - More API workaround for autocomplete	- Standup - Paired coding - Design - Usability testing - Feature prioritization - Backlog management - Retro Finished!

Cloud Platform is mentioned once! Success!

Boot-strapping

Steady-state

Source: Pivotal GSA Response Addendum, July 2015, also, see more in the project scrap book.

Pivotal

“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

Pivotal