

Pivotal

# Creating a DevOps\* culture, whatever that means

---

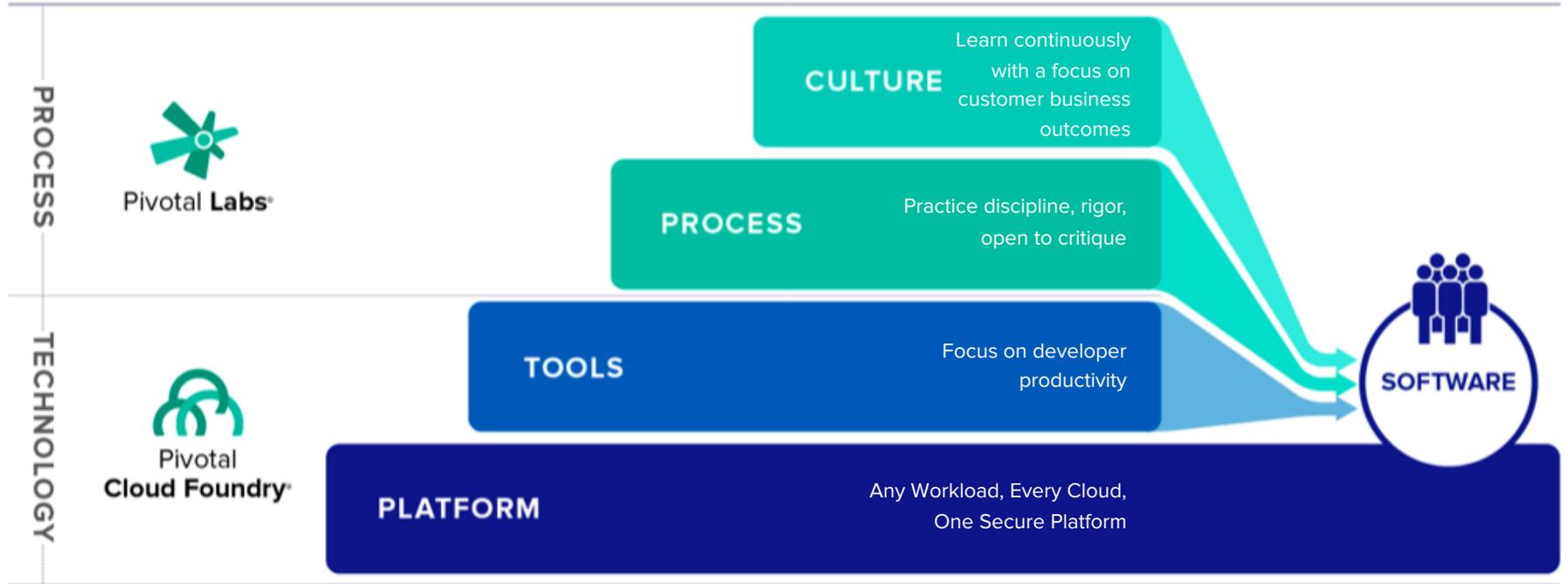
[Singapore Spring User's Group Meetup](#)

Nov, 2018

@cote

\* Disclaimer: I don't really know what to call "doing all the things and using all the tools that result in better software" so I just say "DevOps," which is much shorter than "doing all the things and using all the tools that result in better software," less cheesy than "digital transformation," and seems to be what people want to hear anyway.

# Pivotal: transforming how the world builds software



# Enterprises are transforming with Pivotal

## AUTO & TRANSPORTATION



## INDUSTRIAL & BUSINESS SVCS.



## FINANCIAL SERVICES



## HEALTHCARE & INSURANCE



## TECHNOLOGY & MEDIA



## CONSUMER & COMMUNICATIONS



# “As Is”



# “To Be”



“Executive sponsorship for us was absolutely critical. Without that we could not have gotten anywhere that we needed to be.”

*Brendan Aye*

**T-Mobile**  
(USA)



## “The Business”



Corporate strategy, revenue goals, mission, etc.

## IT Leadership, EAs, etc.

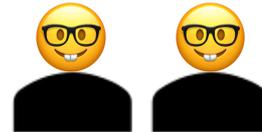


Culture, process, methodology, governance, compensation, etc.

## Product teams

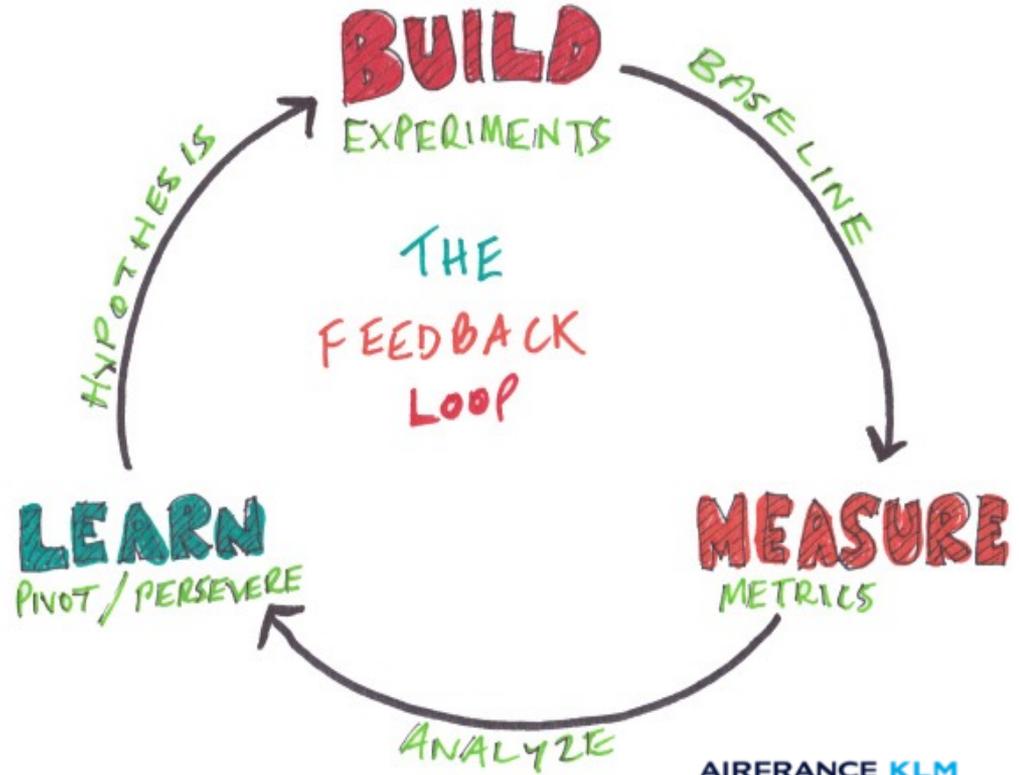
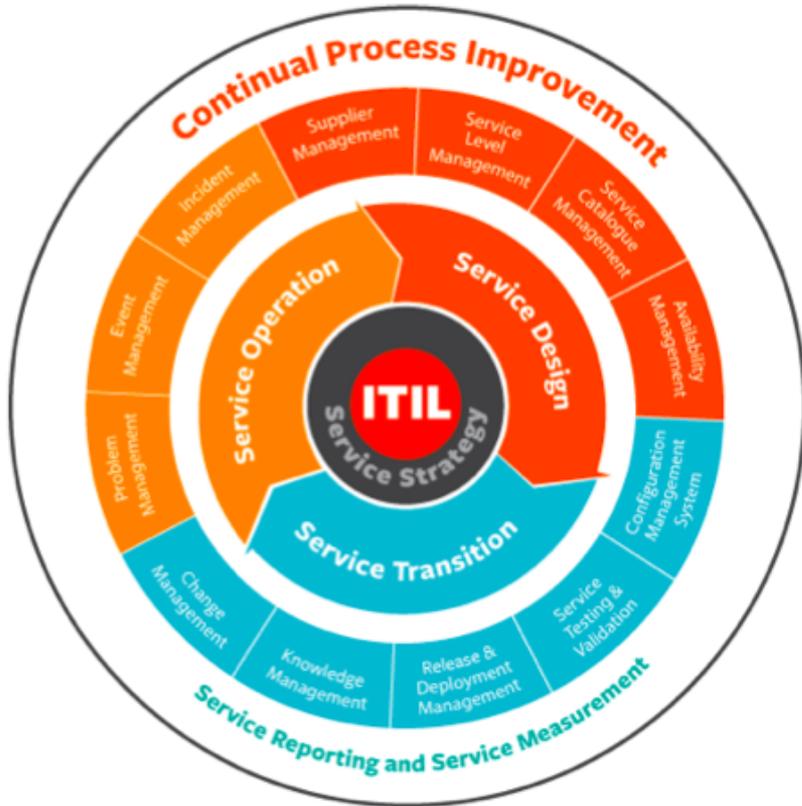


## Platform Engineers



☁️ Private or public IaaS ☁️

# Move from projects to products



AIRFRANCE KLM



What even is  
“culture”?

<b>Pathological (power-oriented)</b>	<b>Bureaucratic (rule-oriented)</b>	<b>Generative (performance-oriented)</b>
Low cooperation	Modest cooperation	High cooperation
Messengers shot	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to enquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

“Culture can be seen in the norms and values that characterize a group or organization that is, organizational culture is a system of shared values and norms that define appropriate attitudes and behaviors for its members.”

“[Culture is] a pattern of shared tacit assumptions that was learned by a group as it solved its problems of external adaptation and internal integration, that has worked well enough to be considered valid and, therefore, to be taught to new members as the correct way to perceive, think, and feel in relation to those problems.”

How we do things around here.

# Beyond 10 ways to make coffee: leadership's role



- Create a culture of innovation, risk taking, and focusing on people (“users”)
- Give product teams autonomy, trust, and voice
- Establish trust by doing what you say, delegating, and celebrating failure learning.
- Feedback loop: surveys (ENPS), silence, churn, results.
- Giving feedback: OKRs, praising process & stories, not just schedule

## People are:

- Innovative
- Risk takers
- People-centric



## Leaders give them:

- Autonomy
- Trust
- Voice

“

[I]f you say to your team that 'when you build it you also run it,' you cannot do that with a consolidated environment. You cannot say to a team 'you own that stuff, and by the way somebody else can also break it.'"

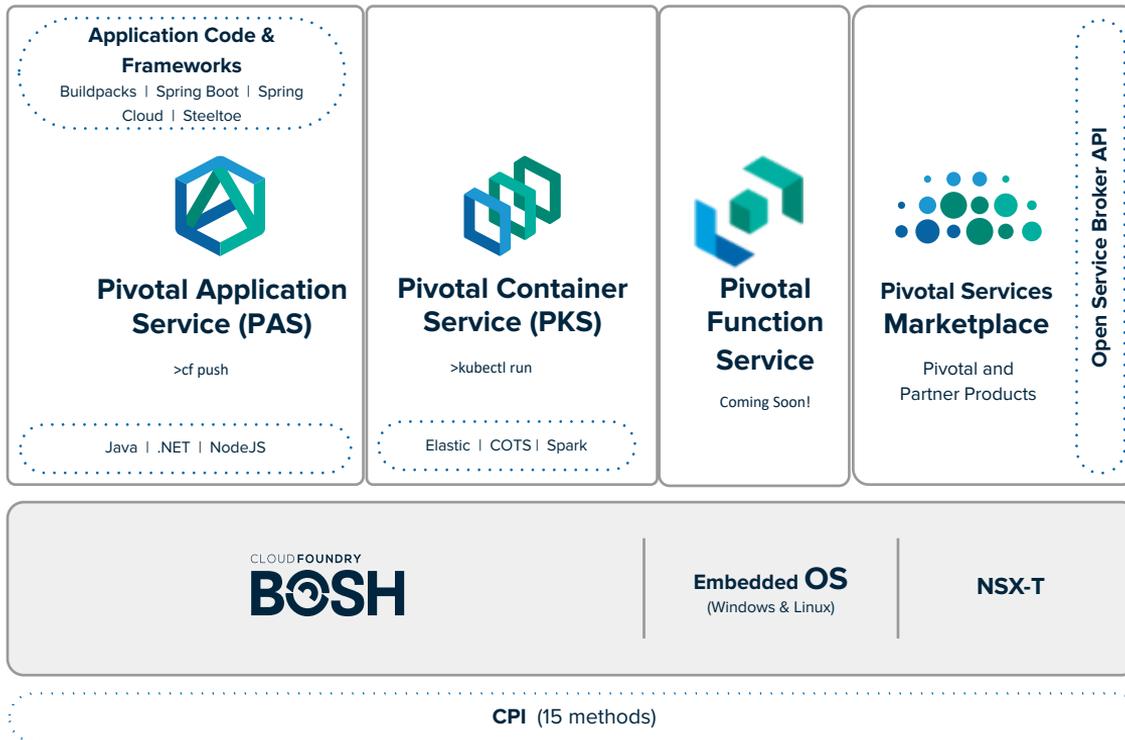
*Vincent Oostindië, Rabobank*



Rabobank



# Pivotal Cloud Foundry®



## Leader tactics:

- Delegate
- Celebrate failure
- Give feedback



## For Kids

**Instead of** “You’re so smart” or “You picked that up so quickly,” **say** “That was a clever approach” or “I’m proud of your persistence.”

**Instead of** “You’re a natural” **say** “Practice is really making you better.”

**Instead of** “Did you win?” **say** “Did you give your best effort?”

**Instead of** “How was your day?” **say** “What did you learn today?” or “What mistakes did you make that taught you something?”

**Instead of** “What do you want to do when you grow up?” **say** “What are your plans for reaching your goals?”

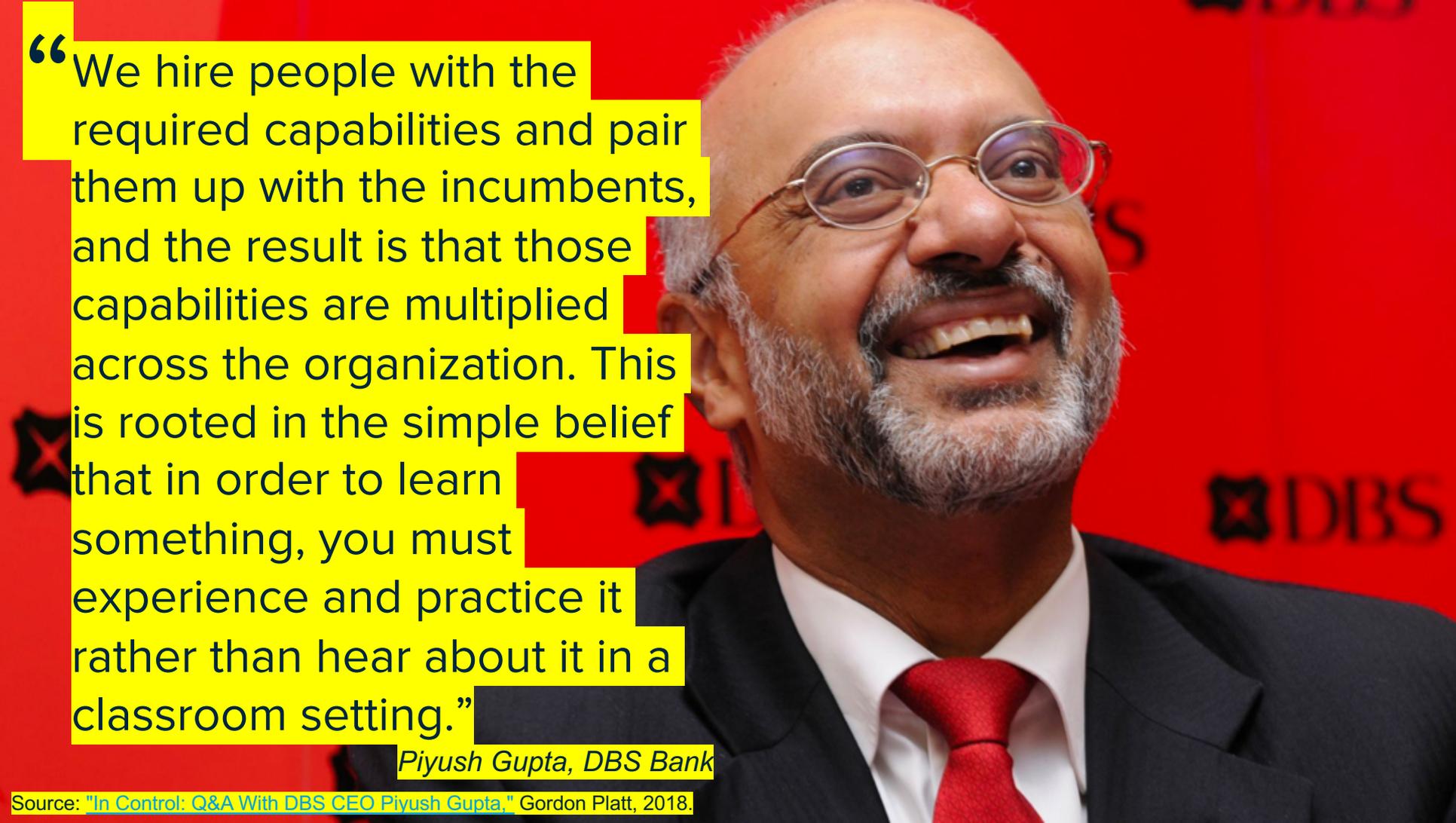
Never let failure progress from an action to an identity.

Never label kids, e.g. “Jimmy is the artist” or “Susie is the computer geek.”

When a child doubts her ability, ask her to think of areas where she once had low ability and now excels, or to recall a time when she saw someone learn something or improve in ways no one thought possible.



But, how to scale?



“We hire people with the required capabilities and pair them up with the incumbents, and the result is that those capabilities are multiplied across the organization. This is rooted in the simple belief that in order to learn something, you must experience and practice it rather than hear about it in a classroom setting.”

*Piyush Gupta, DBS Bank*



# Internal marketing: Everything, frequently



Sounds great!

But no one cares.

# Un-motivated executives: focus on cost reductions

**82% average reduction in 'path to production' time (from 84 days to 15 days)**

## Software Development Lifecycle



**Provisioning**  
**90% faster**

90% time reduction

87% process time reduction

Total time: from 65 to 6 days

Process time: from 81 to 11 hours



**Developing**  
**37% more time coding**  
**E.g.: \$4.80m/year**

37% relative improvement in time spent coding (from 56% to 77%)

21% absolute improvement in time spent coding (from 56% to 77%)



**Releasing**  
**61% faster**  
**Time-to-market**

61% time reduction

82% process time reduction

Total time: from 14 to 5 days

Process time: from 115 to 21 hours

## On-going Ops



**Operating (Day 2)**  
**69% more efficient**  
**E.g.: \$7.35m/year**

69% time reduction

74% process time reduction

Total time: from 79 to 24 hours

Process time: from 34 to 9 hours

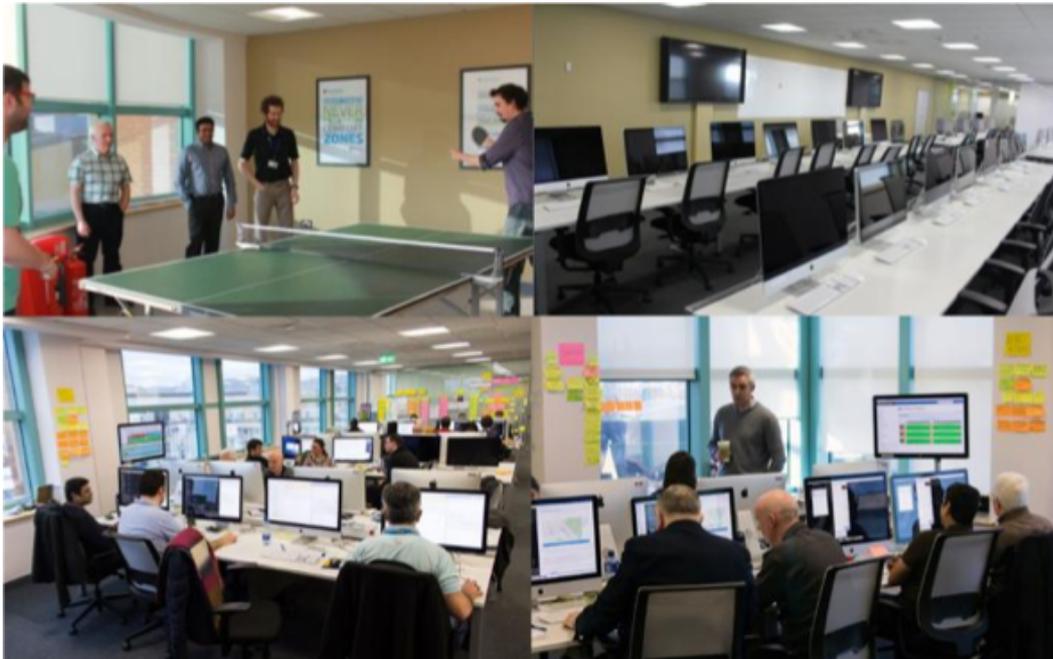
Notes 1: 2017 & 2018 Pivotal customer case studies. n=15. "Process Time" = hands-on work (vs. "total time" being the time to deliver).

Note 2: Averages are exclusive of highest and lowest measures.

Note 3: savings based on insurance co.'s strategy to move 17 app portfolio to Pivotal Cloud Foundry. Dev:ops ratios: 30:1 (insurance co.), 375:1 (Comcast), 62:1 (Northern Trust).



# Changing is often too hard, so create a new organization



- **CompoZed Labs now handles 40% of Allstate's software development**
- **Road-side assistance app delivered in 6 months**
  - **25k rescues/month**
  - **30 minute wait to 15**
  - **+\$10m, +\$100m planned**

# “If that crusty, old .Net developer can do it, anyone can,” transforming people

1. Most people are skeptical for good reasons
2. They enjoy doing IT if it's rewarding
3. Volunteer based at first, building up peer-to-peer marketing
4. Also, there's plenty of more comforting IT for grumpy people to work on

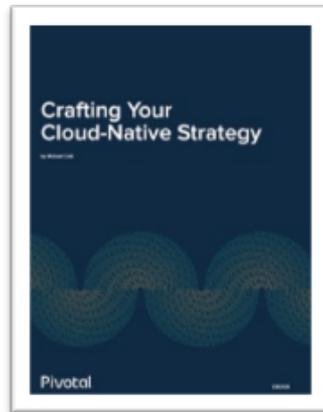


## Things missing:

- Transparency
- Monitoring
- Recruiting
- \$ Comp.

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

- [The Agile Manifesto](#), 2001



<http://cote.coffee/bettersoftware/>