

Pivotal

Creating a DevOps* culture, whatever that means

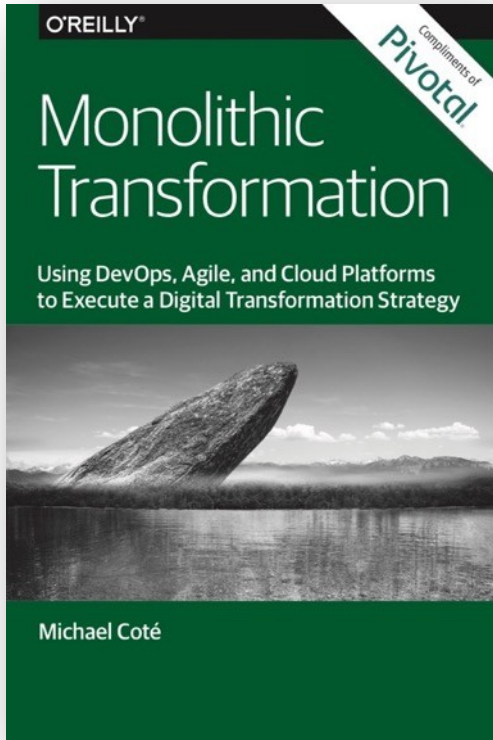
Continuous Lifecycle London
May, 2019
@cote

* 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. If you don't like that, pick from [one of the four other definitions I know of](#).

The technology is easy.
It's the people that are hard.



Everyone who's ever opined on DevOps



<https://cote.io/books>

Pivotal

@cote

Pivotal™



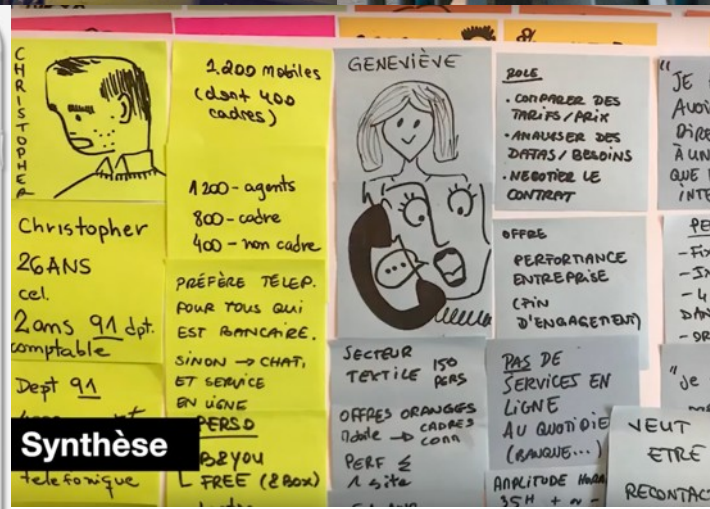
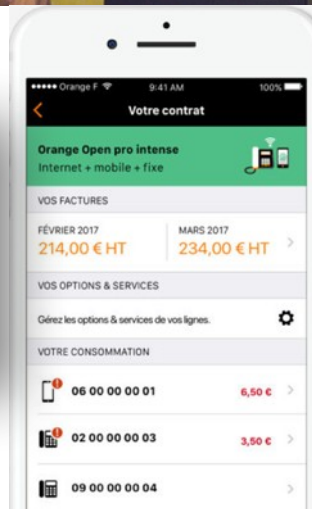
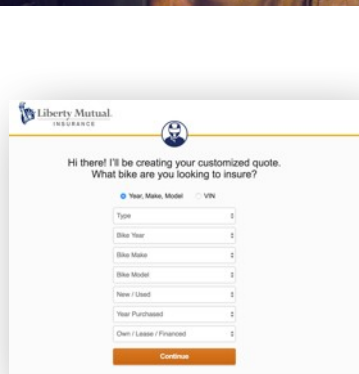
The Register®

RedMonk 451 Research®



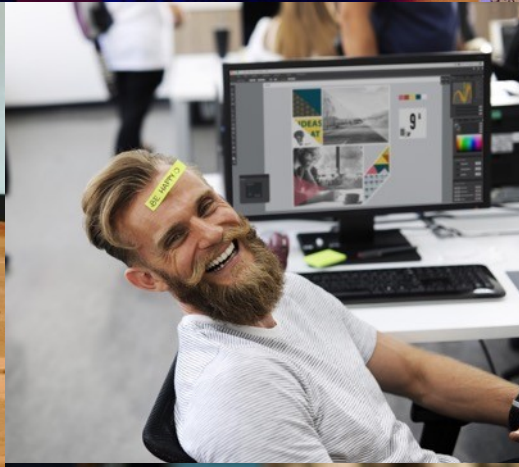
bmc software







What even is
“culture”?



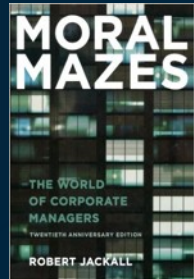
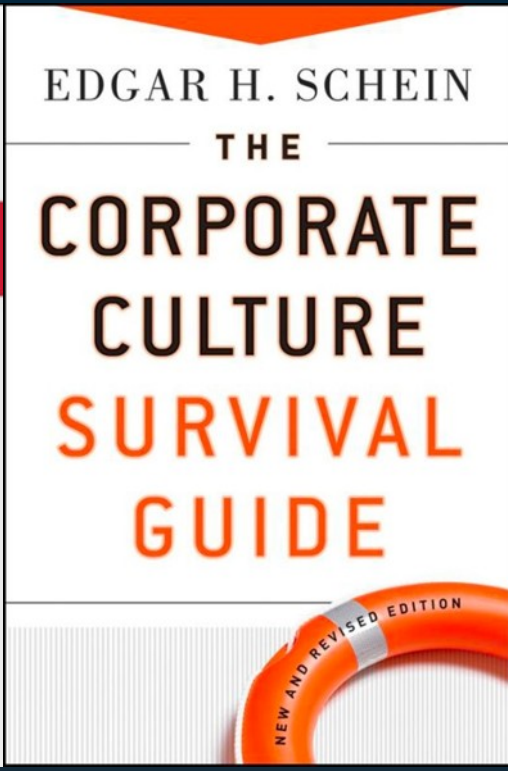
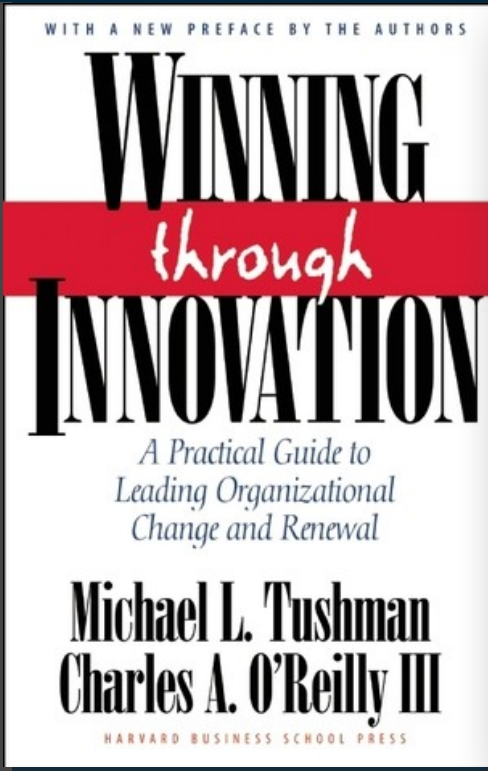
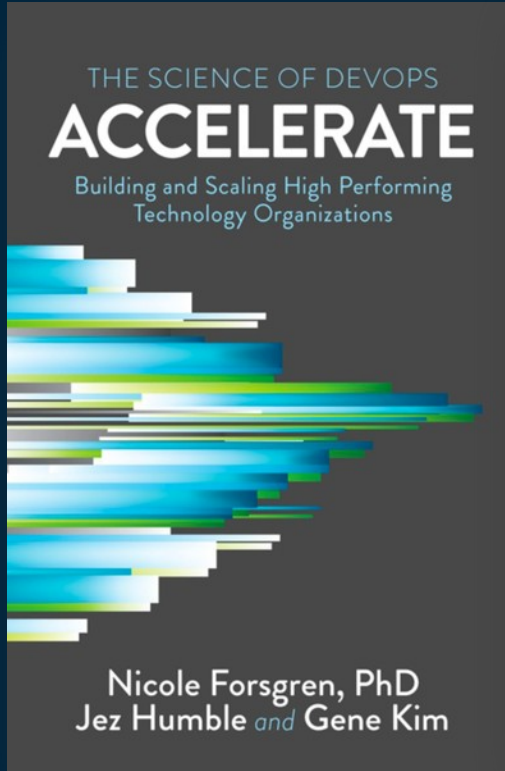
Life is so... Just change way you...
KISS Keep...
Set fire to the world with your brokenness.
Be single be virgin be unhappy...
Tanu now ka...
With all the members of group...

Pathological (power-oriented)	Bureaucratic (rule-oriented)	Generative (performance-oriented)
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.”

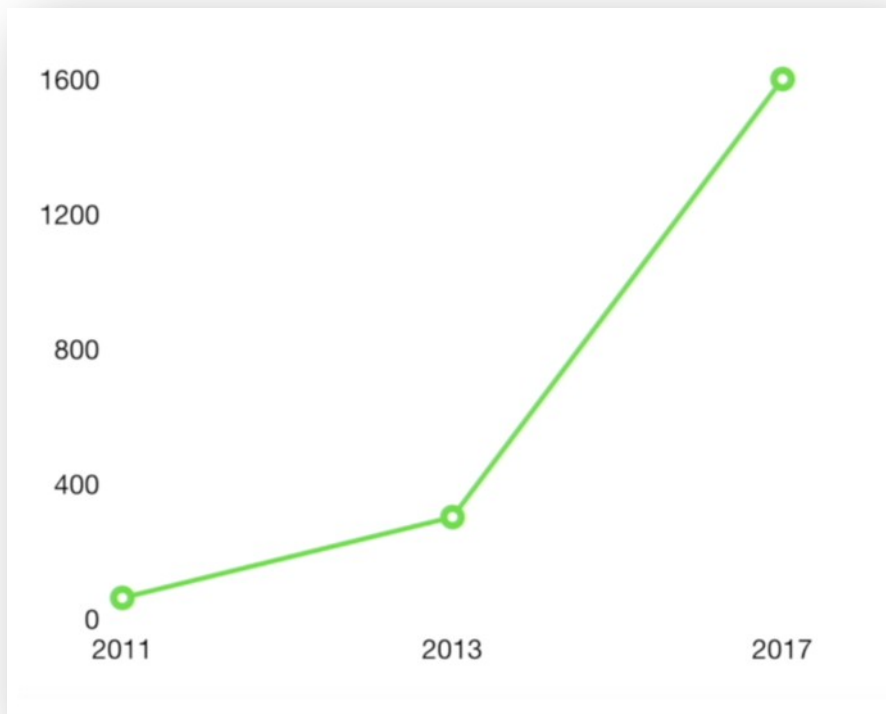
“ [Culture is] a pattern of shared tacit assumptions that was learned by a group as it solved its problems of external adaptation and internal integration, that has worked well enough to be considered valid and, therefore, to be taught to new members as the correct way to perceive, think, and feel in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

How we do things around here.





Tech staff growth



““ J.P. Morgan CFO Marianne Lake said the US bank now employs **40,000 technologists**. That's an **increase of 10,000 in two years** (in a 2013 investor letter, JPM said it employed 30,000 technologists). **18,000 of those 40,000 are developers**, said Lake.”

People are:

- Innovative
- Risk takers
- People-centric



Source: [Pivotal Conversations #113](#), Sep 2018.





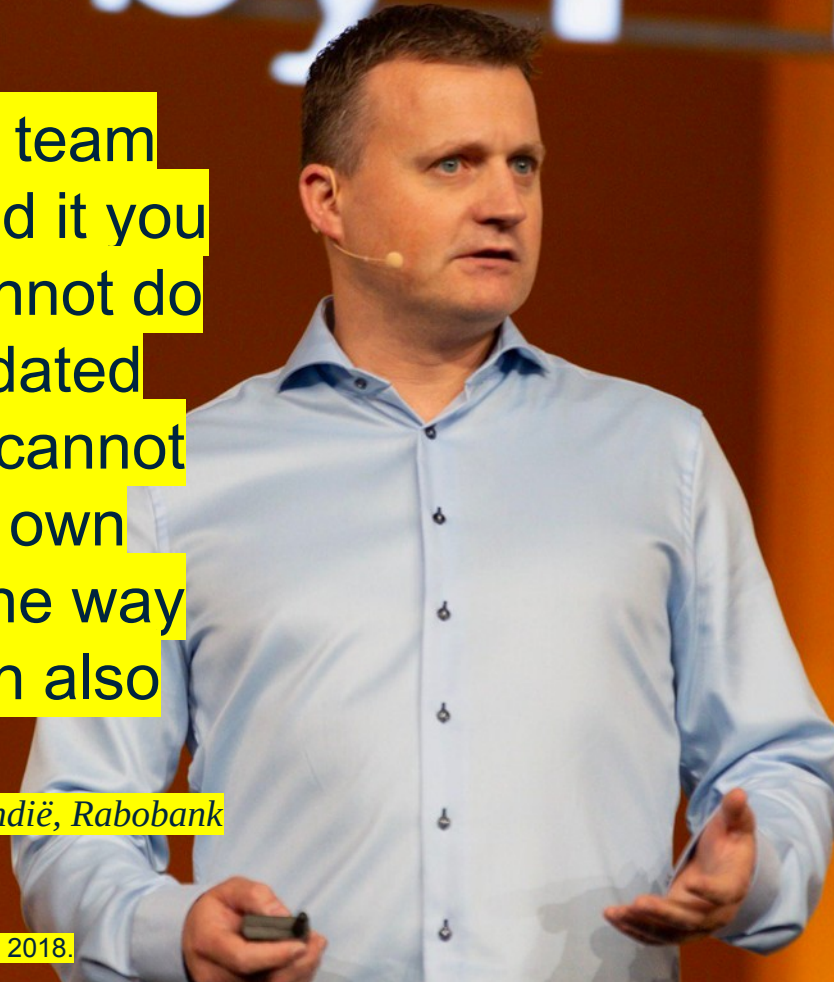
Leaders give them:

- 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



Leader tactics:

- Delegate
- Give feedback
- Celebrate failure



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 believe that we need to reimagine banking to make banking simple, seamless, as well as invisible to allow our customers to **live more bank less.**”

Siew Choo Soh, DBS Bank



false urgency. If the organisation hasn't fallen over yet, there's little chance it's about to right now.

Confronted with a large bundle of big, urgent problems and a sceptical audience, there's a great temptation to become purely reactive. That won't be enough to get you past managing a decline. To get on the front foot, you must begin by setting out how you're going to work.

Design principles

The ten [design principles](#) were one of the first things published by the GDS:

1. Start with user needs.
2. Do less.
3. Design with data.
4. Do the hard work to make it simple.
5. Iterate. Then iterate again.
6. This is for everyone.
7. Understand context.
8. Build digital services, not websites.
9. Be consistent, not uniform.
10. Make things open: it makes things better.

Lots of organisations have something like the design principles. Some call them values, or a philosophy. Unfortunately, most are awful things dreamt up in boardrooms or management away days, in isolation from the way work is actually being done by the organisation. The most important quality of the design principles was that the GDS didn't publish or even draft

them until we'd done quite a lot of designing. Writing down the principles didn't precede delivery, they were written as a result of delivery. Moreover, they weren't written by the 'leadership'. They were written by a team with lots of actual designers working alongside a wide variety of other experts.

The principles sat behind all the best things the team delivered, and helped the digital team avoid the trap of being drawn into reactive firefighting. They've since been endorsed by the [World Bank](#), and emulated by countries and companies all over the world. Tim O'Reilly, the driving force behind open source movement, described them as the 'most significant piece of user interface guidance since Apple's in the 80s'.²³ We forgot them at our peril.

There are several reasons to publish your design principles. For new digital institutions, the most important is to start capturing a new approach that can work at scale for the whole of a huge, distributed organisation. In the UK government's case, the principles were not written to replace the civil service's own four long-established and admirable values: honesty, integrity, impartiality and objectivity. They were written to do something those values were not designed to do – provide instructions for how to actually deliver things. The original values offered a guide for how officials should provide policy advice to ministers; one of the fundamental tasks of central government. Many public officials aren't in the business of giving advice though;

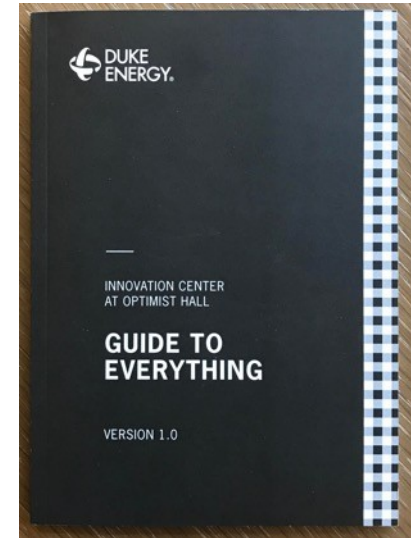
“ We hire people with the required capabilities and pair them up with the incumbents, and the result is that those capabilities are multiplied across the organization. This is rooted in the simple belief that in order to learn something, you must experience and practice it rather than hear about it in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

Piyush Gupta, DBS Bank



Internal marketing, branding, etc.

Organizational Learning



A woman with long dark hair, wearing a grey sleeveless dress, sits at a conference table. She has a skeptical expression and her right hand is raised in a shrug. In the background, several other people are seated at the table, some with their heads resting on the table, suggesting they are asleep or disinterested. A large yellow thought bubble with a black outline is positioned above her, containing two lines of text. The setting is a modern office with a light-colored wall and a window with blinds on the right.

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
\$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
\$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).

“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

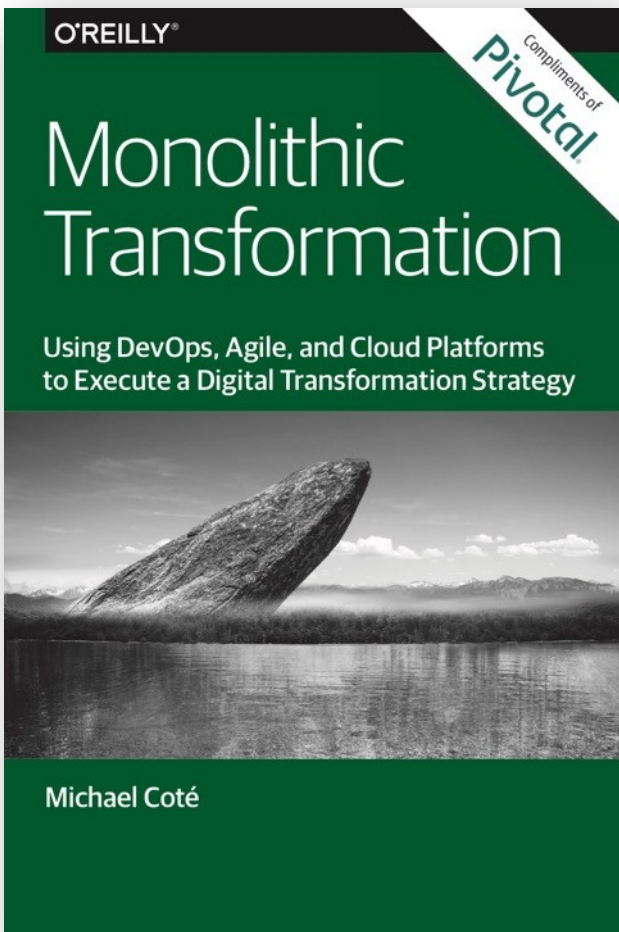


Changing is often too hard, so create a new organization



Things missing:

- Transparency
- Monitoring
- Recruiting
- \$ Comp.



<http://cote.io/books>

@cote | cote@pivotal.io