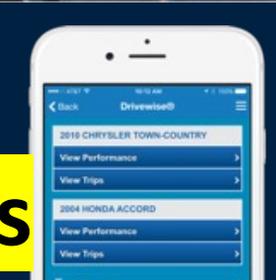
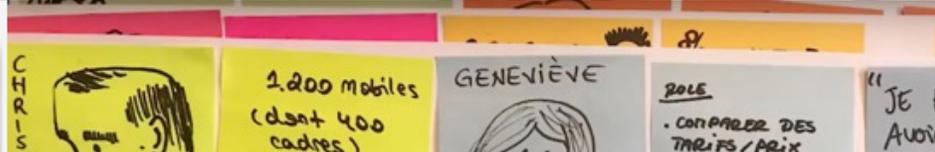
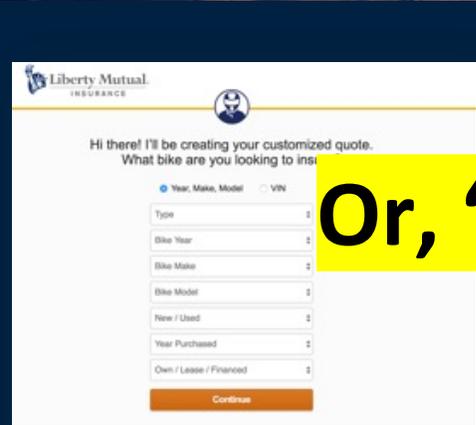


Creating and Sustaining a Culture of Innovation

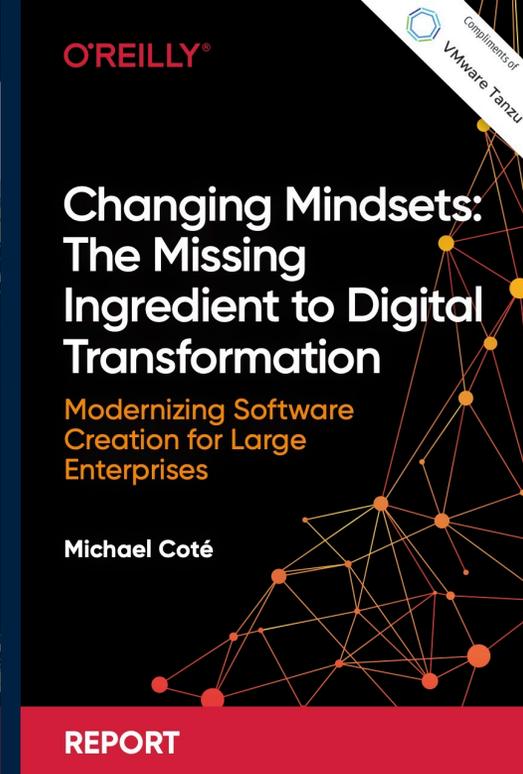
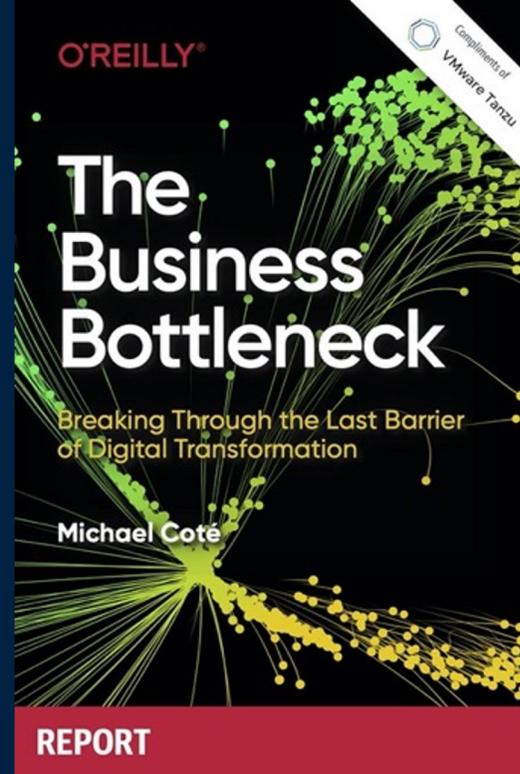
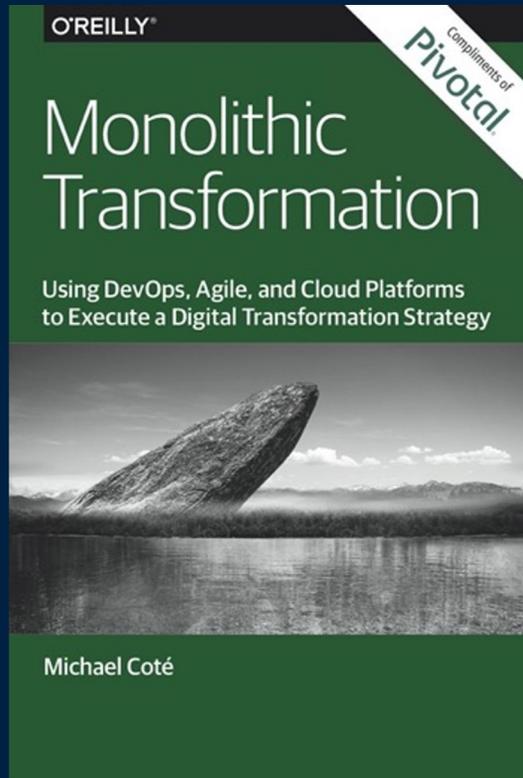


Or, "DevSecOps Culture": Whatever That Means



Coté

@cote | <http://cote.io>





Special Guests!
(Potentially)



How to draw an Owl.

"A fun and creative guide for beginners"



Fig 1. Draw two circles



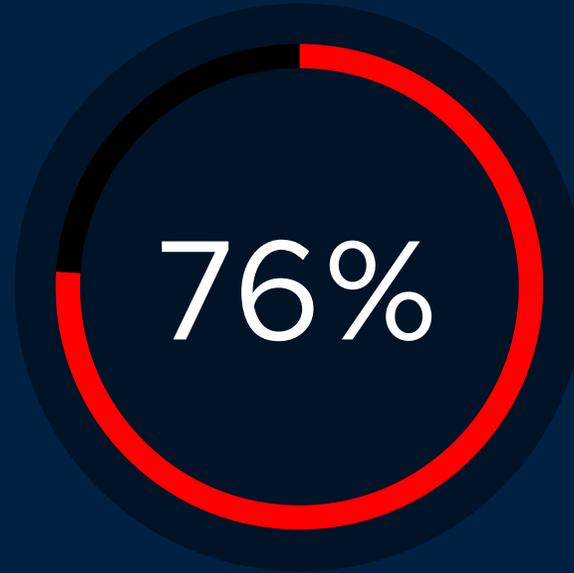
Fig 2. Draw the rest of the damn Owl

How do I focus on
building apps that
improve customer
experience and
improve how we do
business?

30% of software projects are successful, 70% challenged or failing



Have not made app improvements in 1+ years

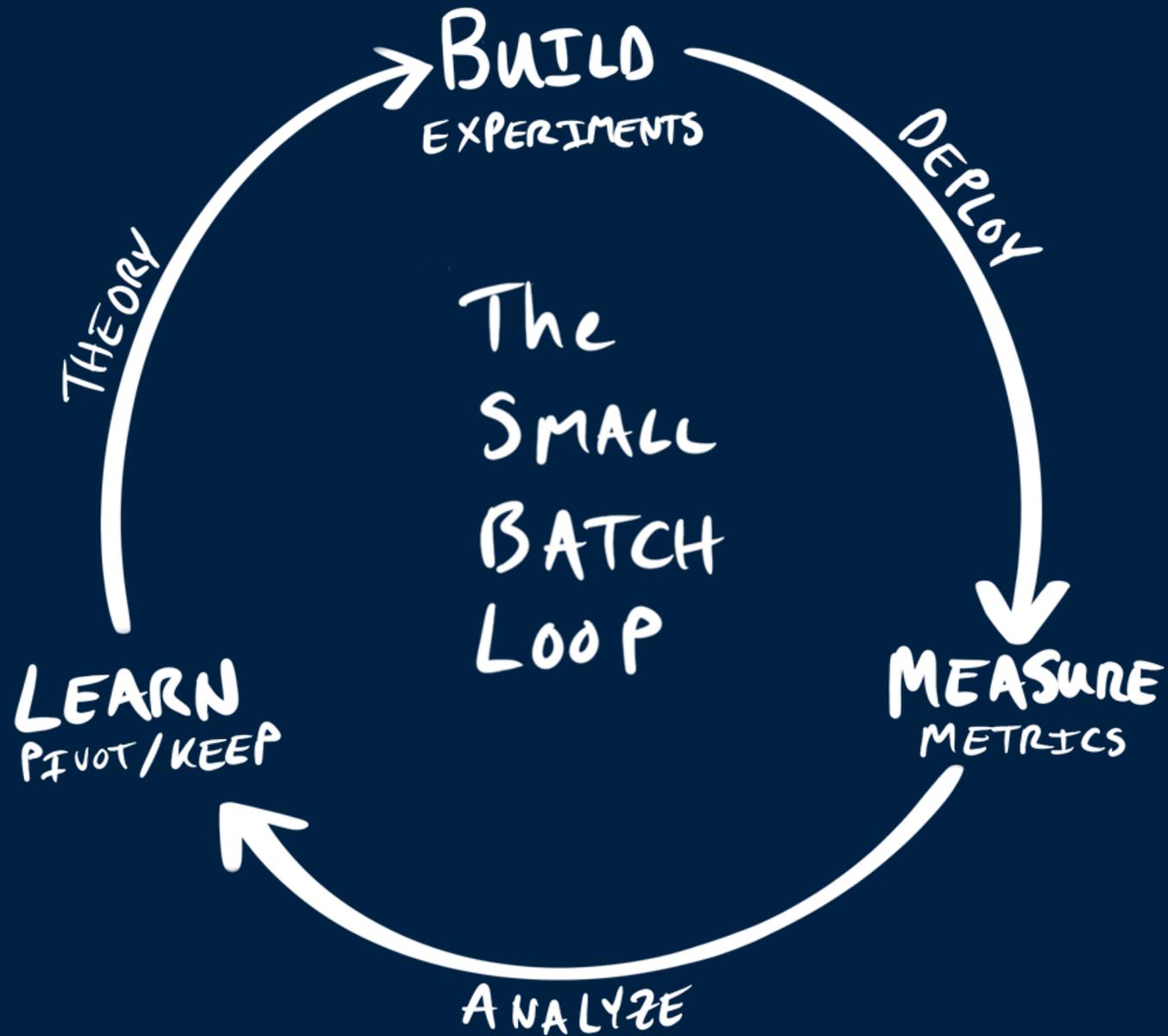


Too invested in legacy apps to change



Release apps twice a year or less



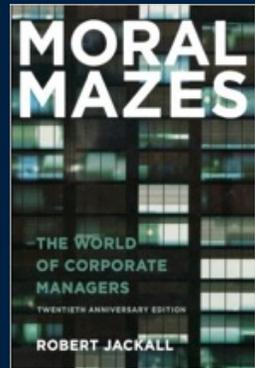
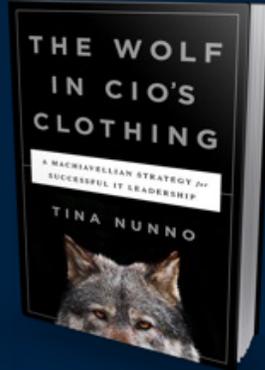
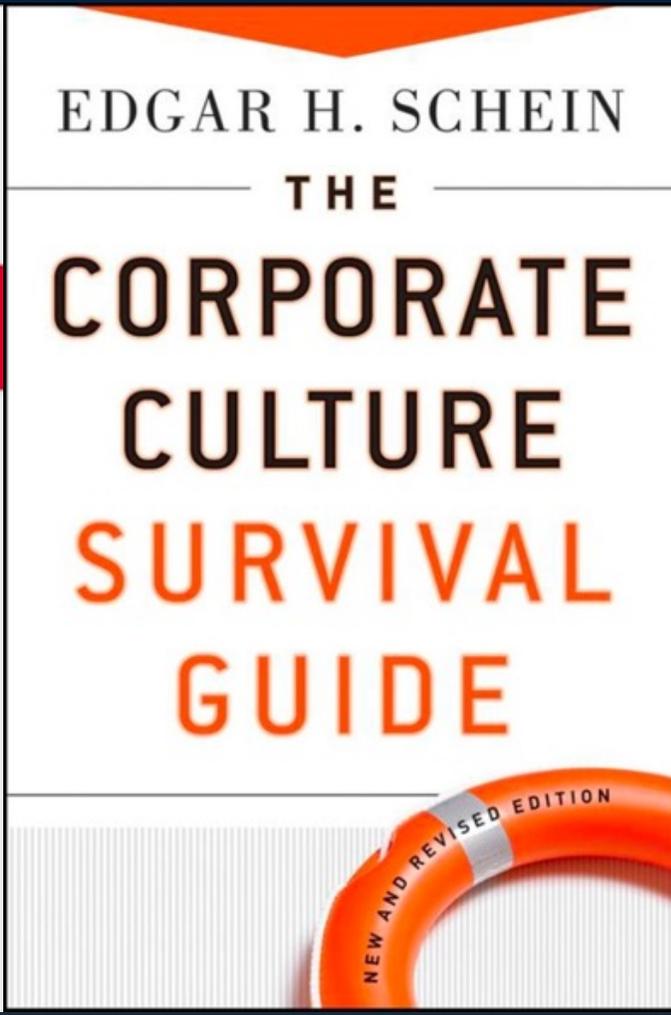
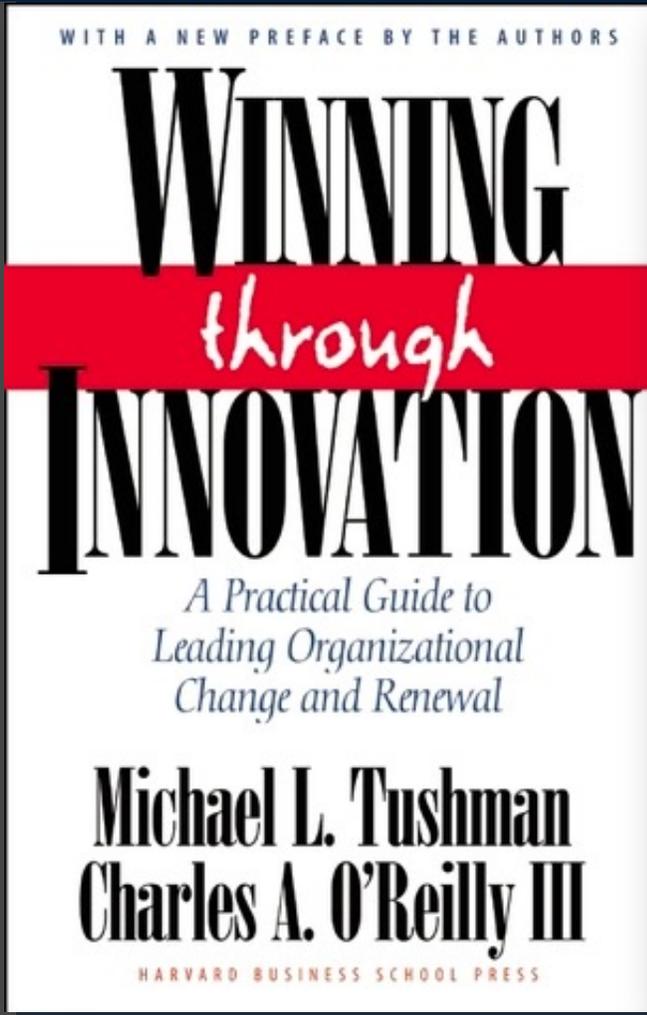
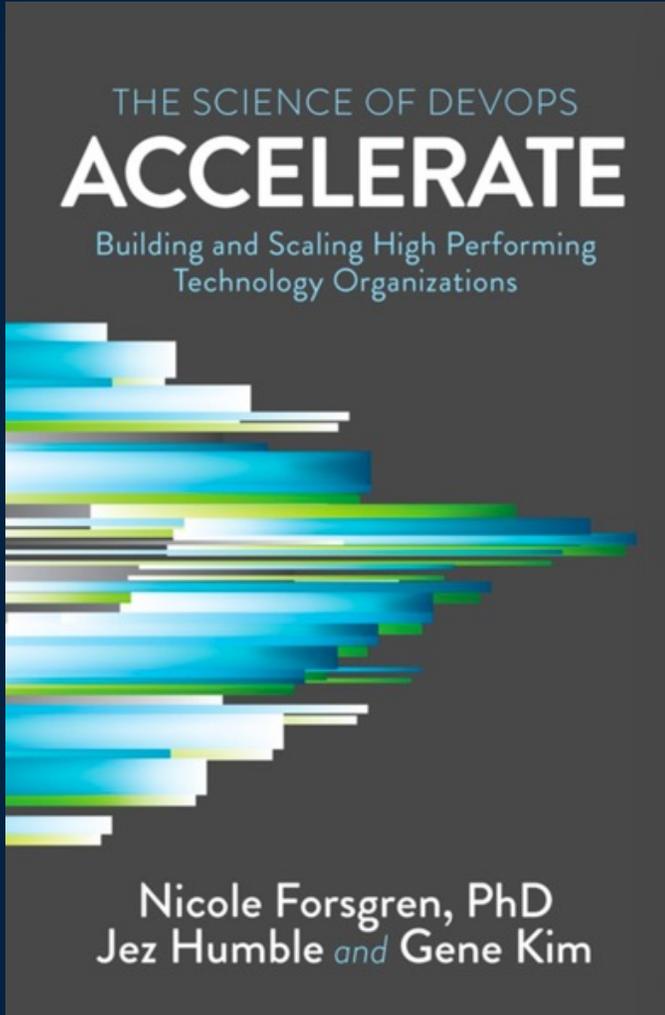




Sources: [Sophie Seiwald](#), Daimler, case study in *The Business Bottleneck*. Pics: [Lüke Kanies](#).



What even is
“culture”?



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.”

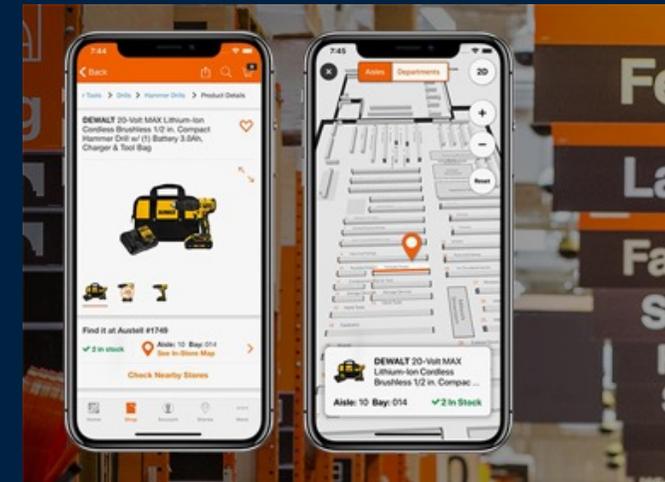
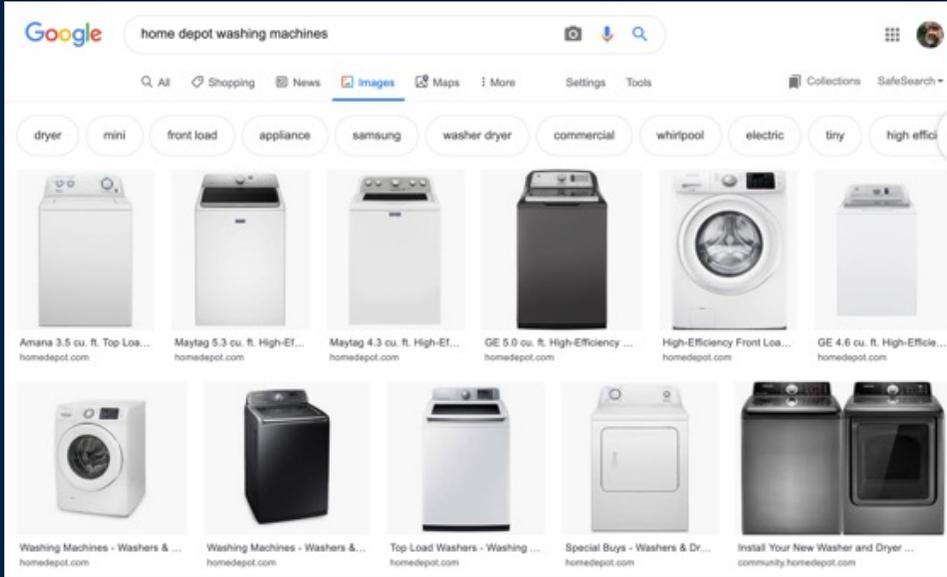
“[Culture is] a pattern of shared tacit assumptions that was learned by a group as it solved its problems of external adaptation and internal integration, that has worked well enough to be considered valid and, therefore, to be taught to new members as the correct way to perceive, think, and feel in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

How we do things around here.

People are:

- Innovative
- Risk takers
- People-centric

Touching washing machines, finding toilets



Leaders give them:

- Autonomy
- Trust
- Voice





But, how to scale?

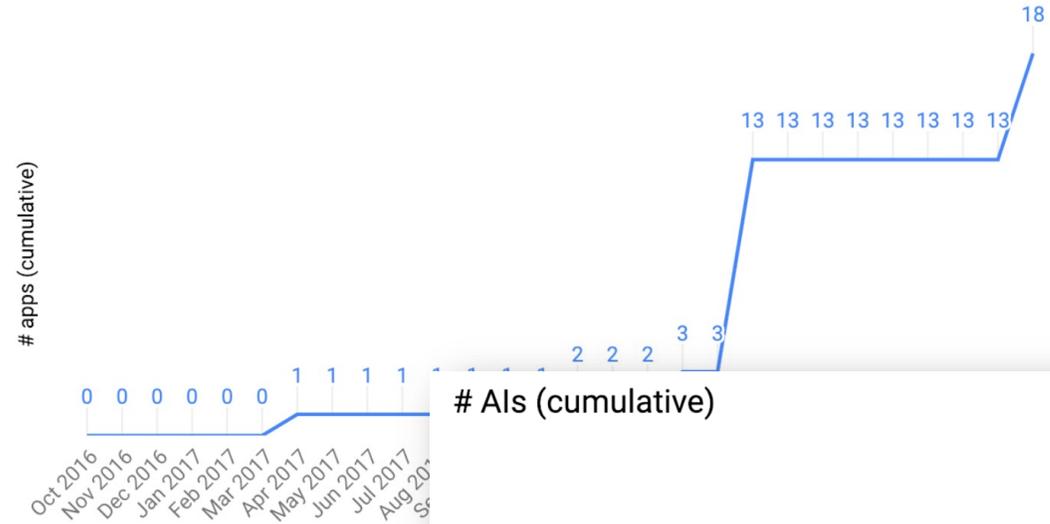
“We believe that we need to reimagine banking to make banking simple, seamless, as well as invisible to allow our customers to live more bank less.”

Siew Choo Soh, DBS Bank

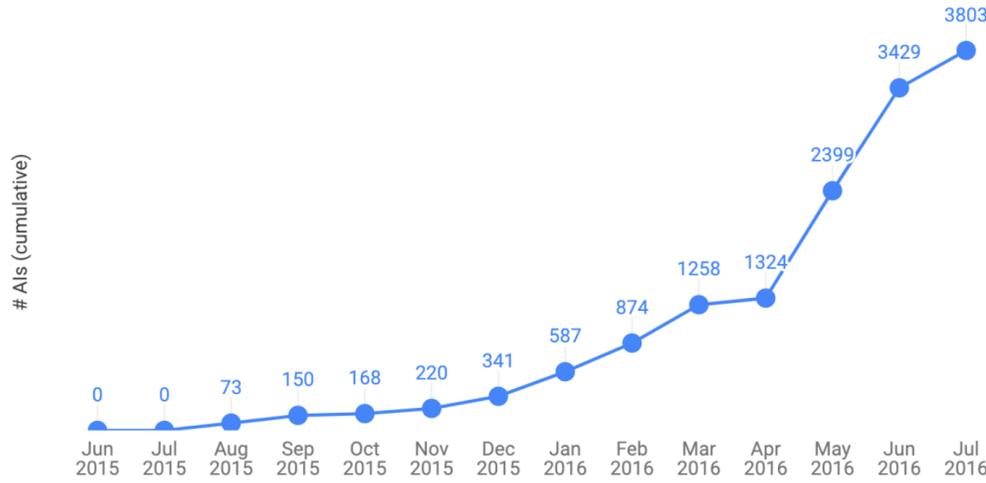


Start small & steady, expand by seeding & winning trust

USAF AOC apps released, cumulative



AIs (cumulative)



Further analysis of just high and elite performers shows the following use profiles:

- 46% - Communities of Practice, Grassroots, and PoCs
- 23% - Grassroots efforts and Communities of Practice
- 22% - all by Big Bang and DOJOs, but inc. PoC but stall
- 9% Centers of Excellence, Communities of Practice, and Training Centers

“Most of our software development is supporting legacy capabilities that are in-house and don’t provide a competitive advantage, which is opposite of what I would want to have.”

CIO at US Retail Company

- **76%** of executives said they are **too invested in legacy applications** to change.
- **69%** of executives say **high technical debt is very or extremely important to overcome** to improve apps.
- **48%** say it has been **more than a year since they made improvements** to their application portfolios.

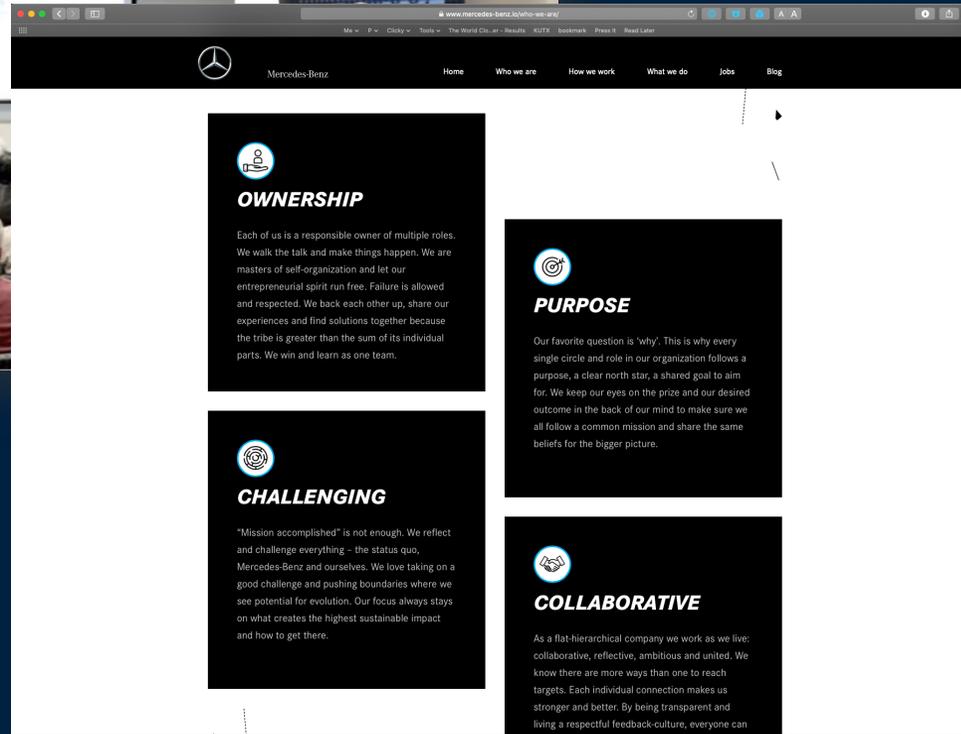
Motivation: “If that crusty, old .Net developer can do it, anyone can.”

1. Money, gold stars, equity
2. Quality work
3. Flow
4. Quality of life
5. Open source contributions
6. Autonomy
7. Moving pixels on the screen (business outcomes)

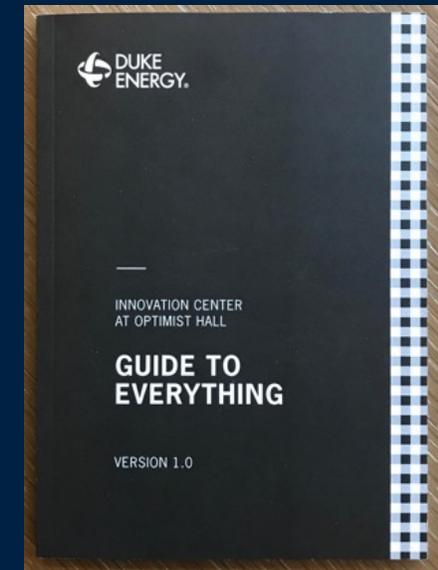


Consulting, internal marketing, branding, etc.

Organizational Learning



Focus on ways of working.....



Use metrics to manage and build trust, focus

DevOps/Technical

Are we doing it right? Does the software work? Can we troubleshoot & restore?

1. Deployment frequency
2. Lead time for changes
3. Time to restore service
4. Change failure rate

Also, SRE: latency, traffic, errors, saturation.

Business

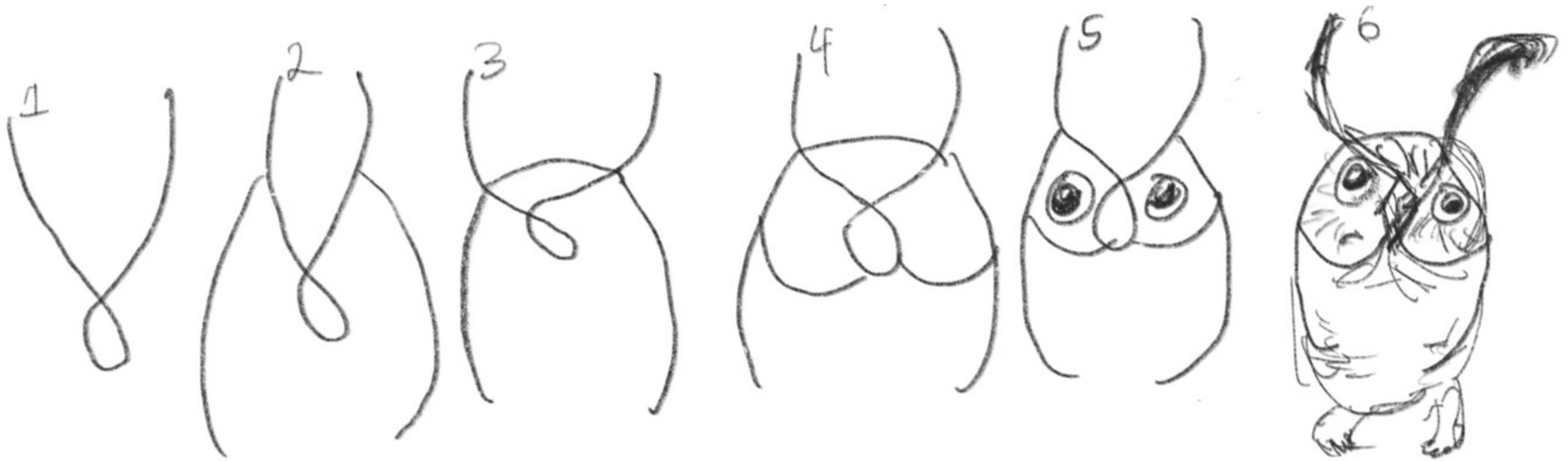
Are we achieving the non-IT goals we're here for? (What even are those goals?)

1. Sales/workflow completion
2. Use
3. Costs – time, money
4. Cost per transaction
5. Customer experience and satisfaction

Culture

Are the executives building the right system?

1. Employee NPS (eNPS)
2. Staff belief in leaders, mission, and strategy
3. # of experiments, learning
4. Staff retention and churn rate



Thanks!

<http://cote.io> | cotem@vmware.com | @cote

Q&A Starters: all too common blockers

People

- Skills, hiring
- Reluctance to change
- Scaling new roles
- Org. structure

Corporate

- Budgeting
- Misaligned executives
- IT is still in the basement
- Compliance

Technical

- Stalled, incomplete platforms
- Overwhelming legacy portfolio
- Local optimization, no CI/CD
- "We already do agile."