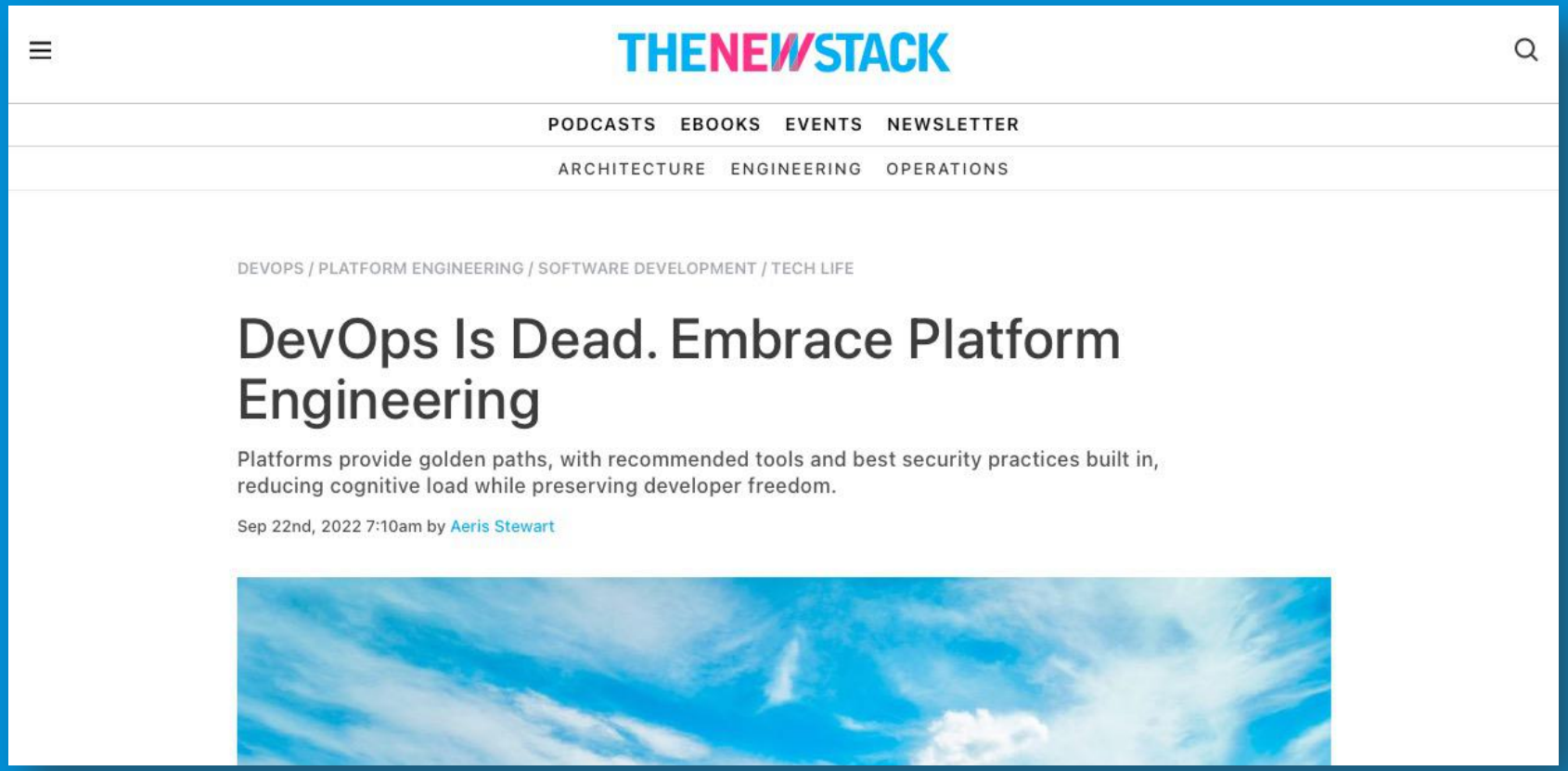
Slides, etc.



Hello.

Welcome to my ongoing therapy session

Things we know are true but do not do

1. The people who do the work should determine how the work is done.
2. Revisit governance frequently, remove when no longer needed.
3. The software factory requires maintenance just like a real factory. (Automation, tech debt.)
4. Switch to product management (also: developers are your customers).
5. Beware "change or die."
6. Sellers want you to buy new things, whether you need them or not.
7. If the technology is *so* complex, why use it?
8. If it's not working, have you tried following the directions?
9. Be a late adopter.)Be OK with being "slow.")
10. Use small batch thinking to be a learning organization.
11. Change in large organizations requires tops down re-engineering.
12. To change, tell people how their lives will be better.
13. Focus on outcomes over activities
14. Make sure your customer is a human, not a dashboard.

DevOps is like flossing...

Hello, I'm Coté

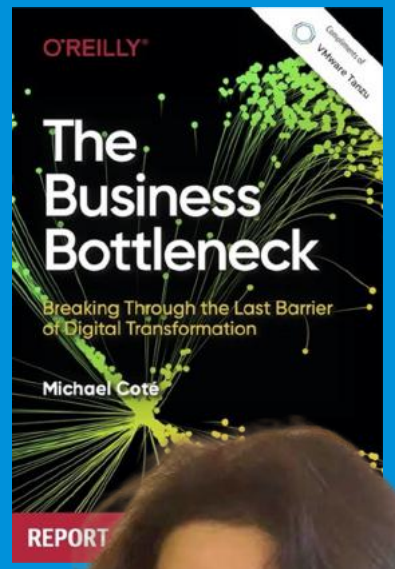
I work at vmware®

. . . and,
I'm a recovering
thought-leader

How its started.

Photo: Simon Phipps, May, 2006

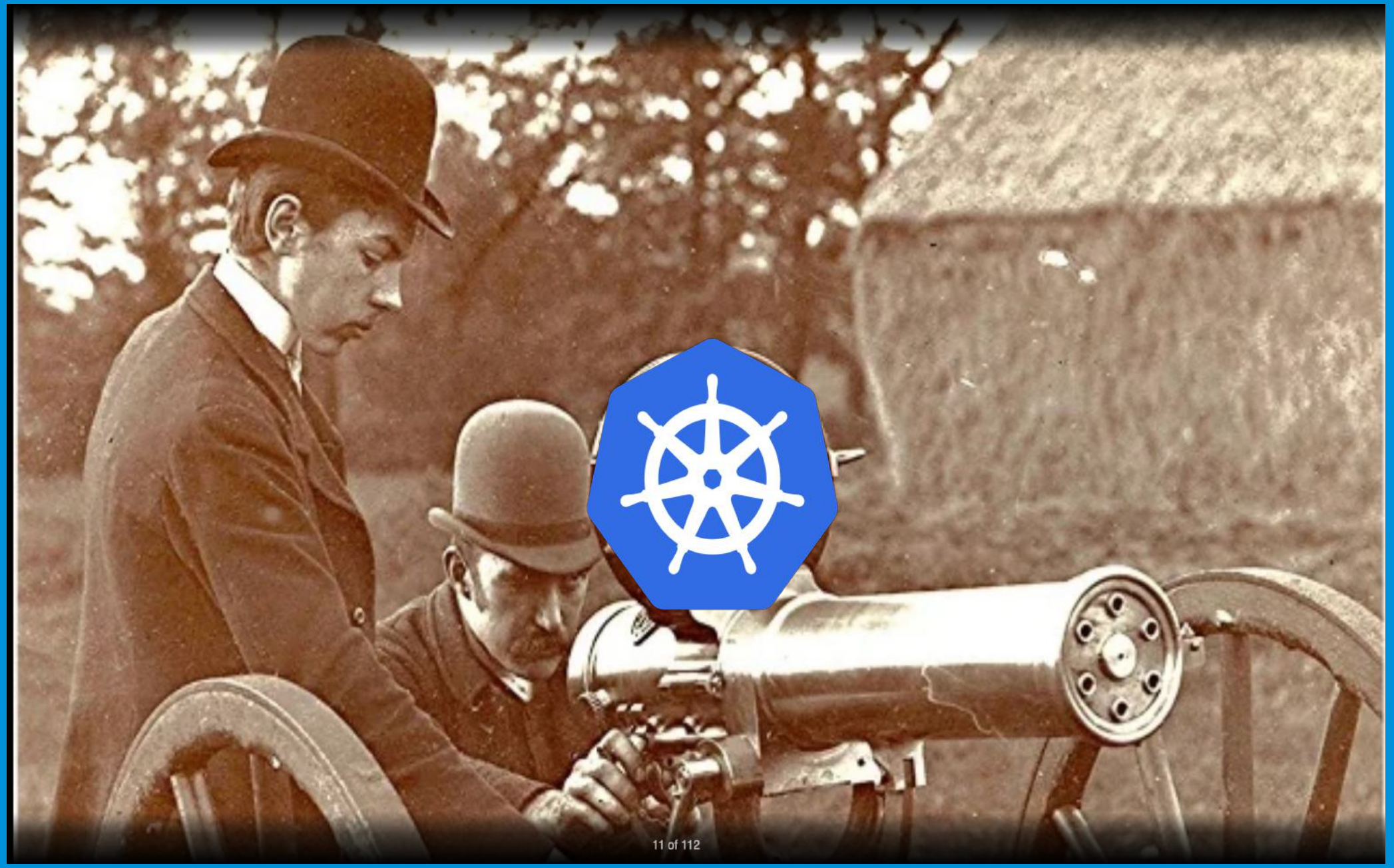
SOFTWARE DEFINED TALK



[INSERT PICTURES GESTUICULATING AT PODIUMS]

Photos: Andrew Shafer, Mark Hinkle(?), Tasha Isenberg, Bridget Kromhout.

How it's going.


11 of 112

DEVOPS / PLATFORM ENGINEERING / SOFTWARE DEVELOPMENT / TECH LIFE

DevOps Is Dead. Embrace Platform Engineering

Platforms provide golden paths, with recommended tools and best security practices built in, reducing cognitive load while preserving developer freedom.

Sep 22nd, 2022 7:10am by Aeris Stewart

Image via Unsplash.

Error:
No Thoughts Found

Things we know are true but do not do

1. The people who do the work should determine how the work is done.
2. Revisit governance frequently, remove when no longer needed.
3. The software factory requires maintenance just like a real factory. (Automation, tech debt.)
4. Switch to product management (also: developers are your customers).
5. Beware "change or die."
6. Sellers want you to buy new things, whether you need them or not.
7. If the technology is *so* complex, why use it?
8. If it's not working, have you tried following the directions?
9. Be a late adopter.)Be OK with being "slow.")
10. Use small batch thinking to be a learning organization.
11. Change in large organizations requires tops down re-engineering.
12. To change, tell people how their lives will be better.
13. Focus on outcomes over activities
14. Make sure your customer is a human, not a dashboard.

#1

The people who do the work determine how the work is done.

Leaders at the Genba

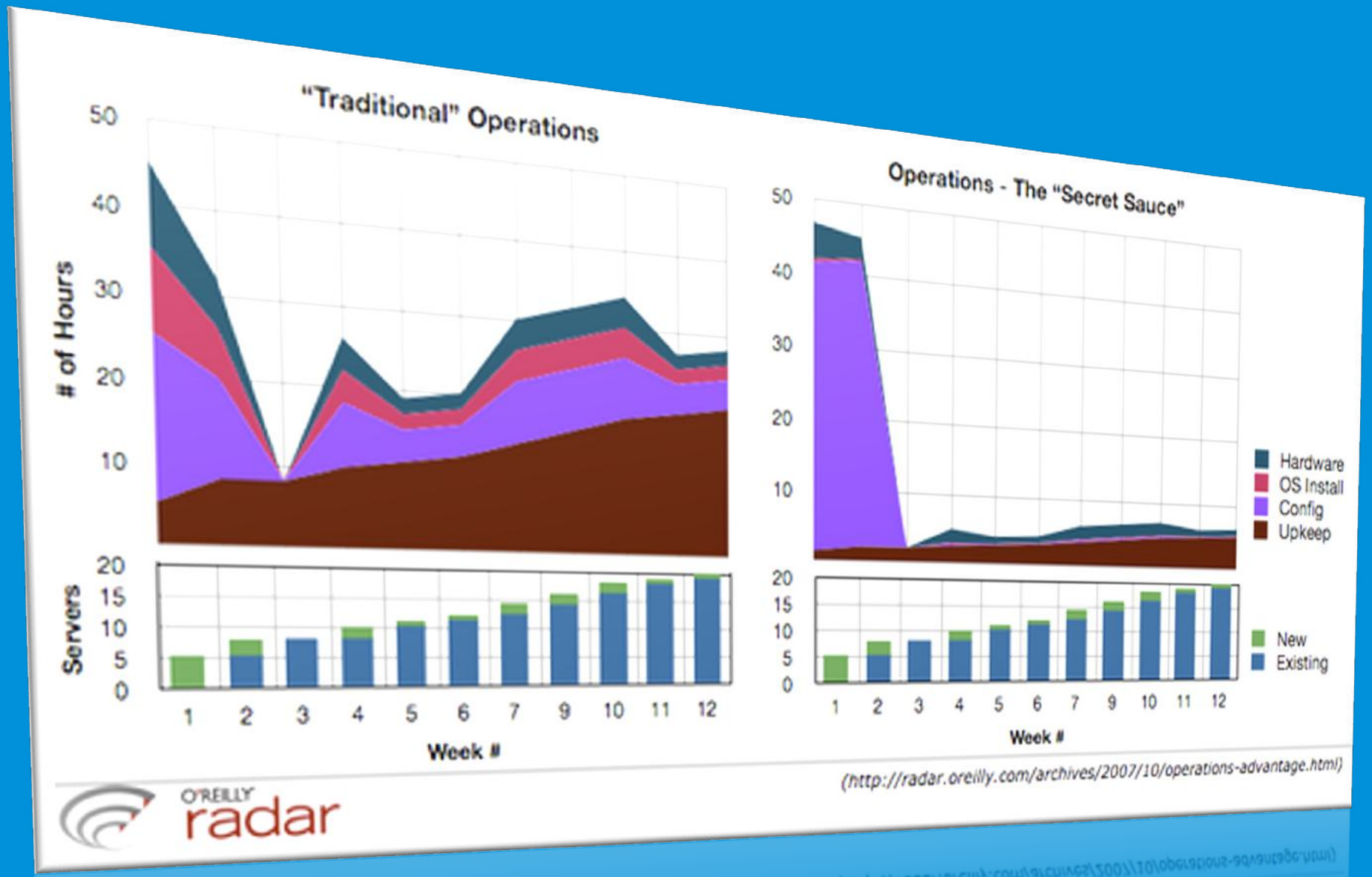
The boss made immediate changes once I put him on the line!

Sources: "DevOps is Enterprise Wide," Nigel Thurlow, DevOpsDays Dallas 2022.


OLLAS DE ALUMINIO
OLLAS DE ALUMINIO
OLLAS DE ALUMINIO

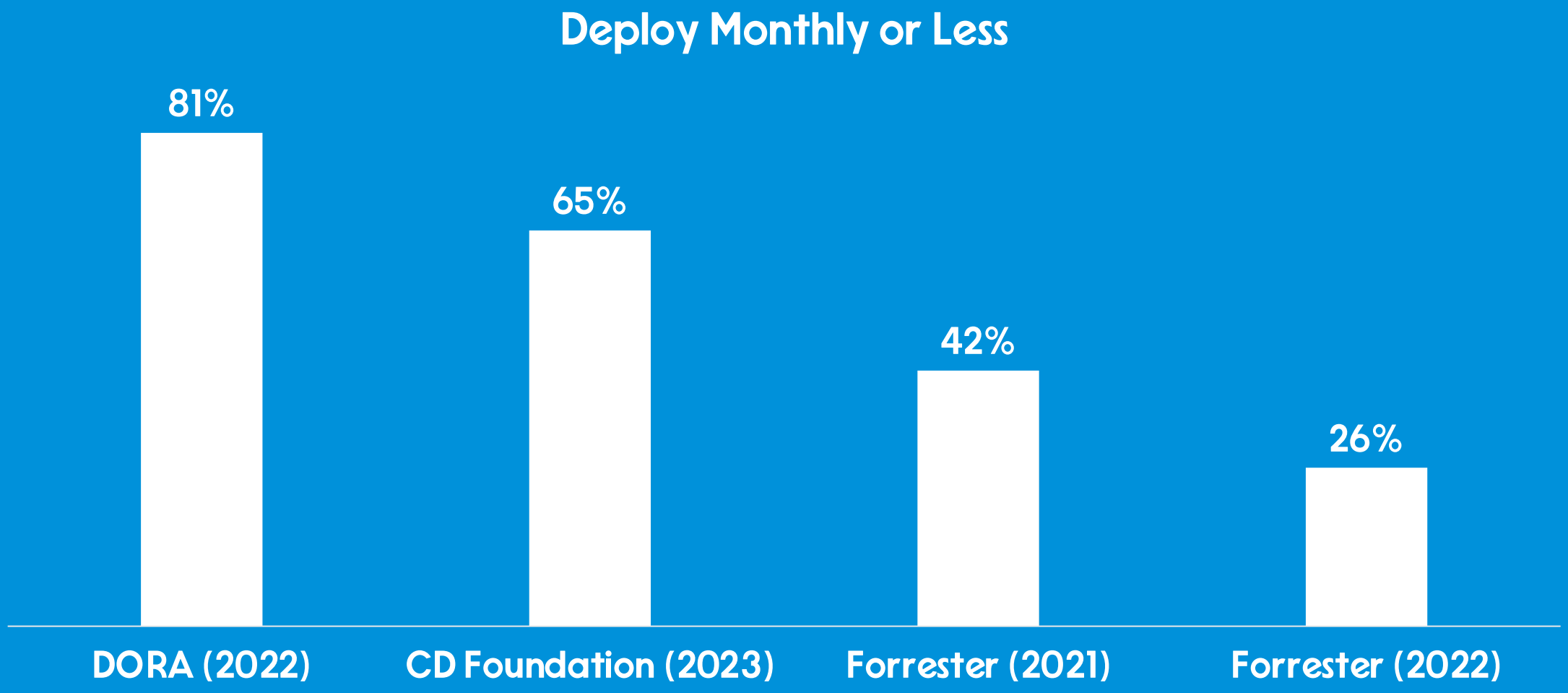
#3

CI/CD, but for real this time

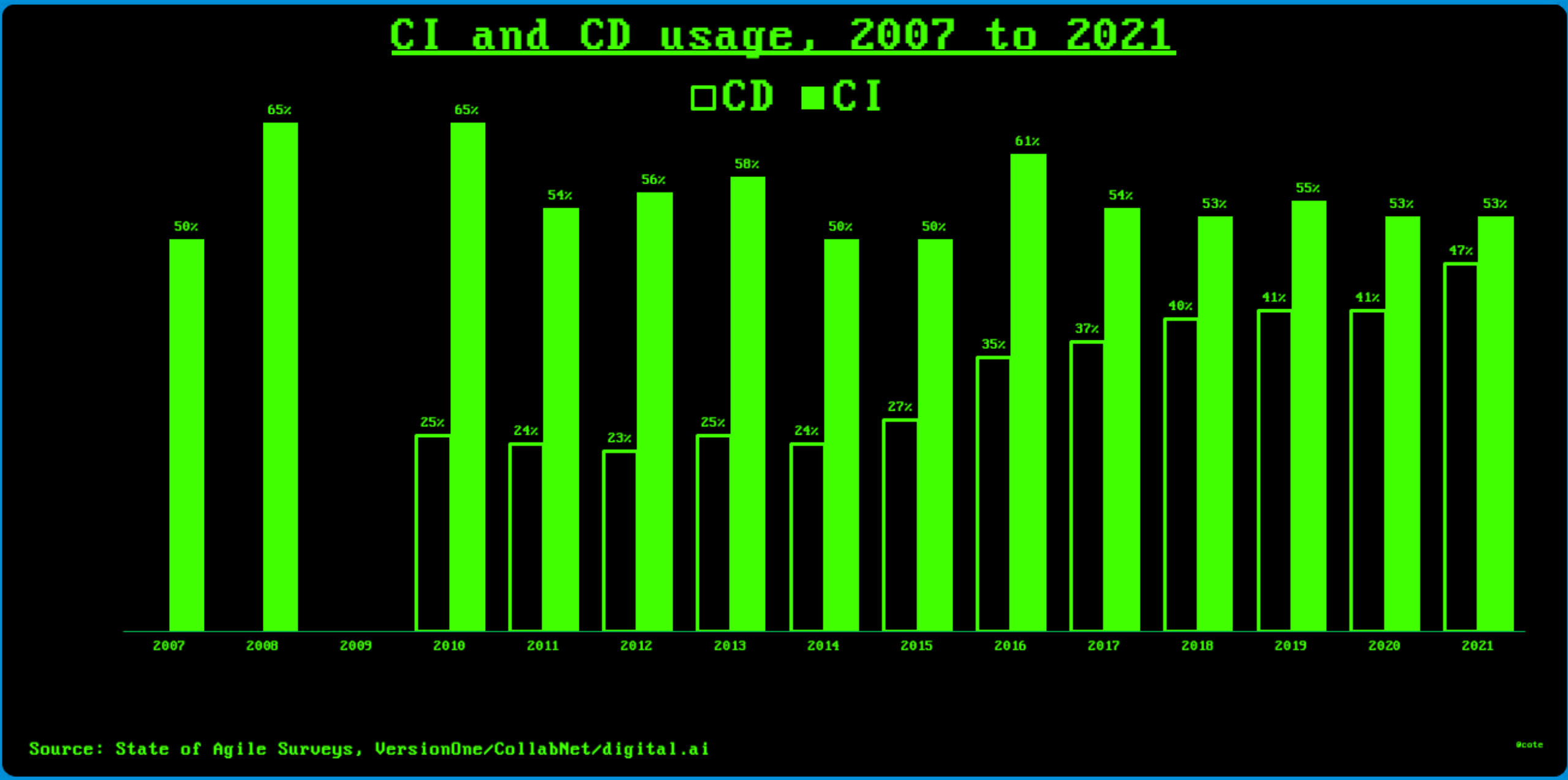


Sources: "Operations is a competitive advantage… (Secret Sauce for Startups!)" Jesse Robins, Oct 2007.

Accounts of deployment rates vary wildly



Sources: see "How's DevOps been going?," Coté, June 2023 for citations and links to sources.



More: "How's DevOps been going?," Coté, June 2023.

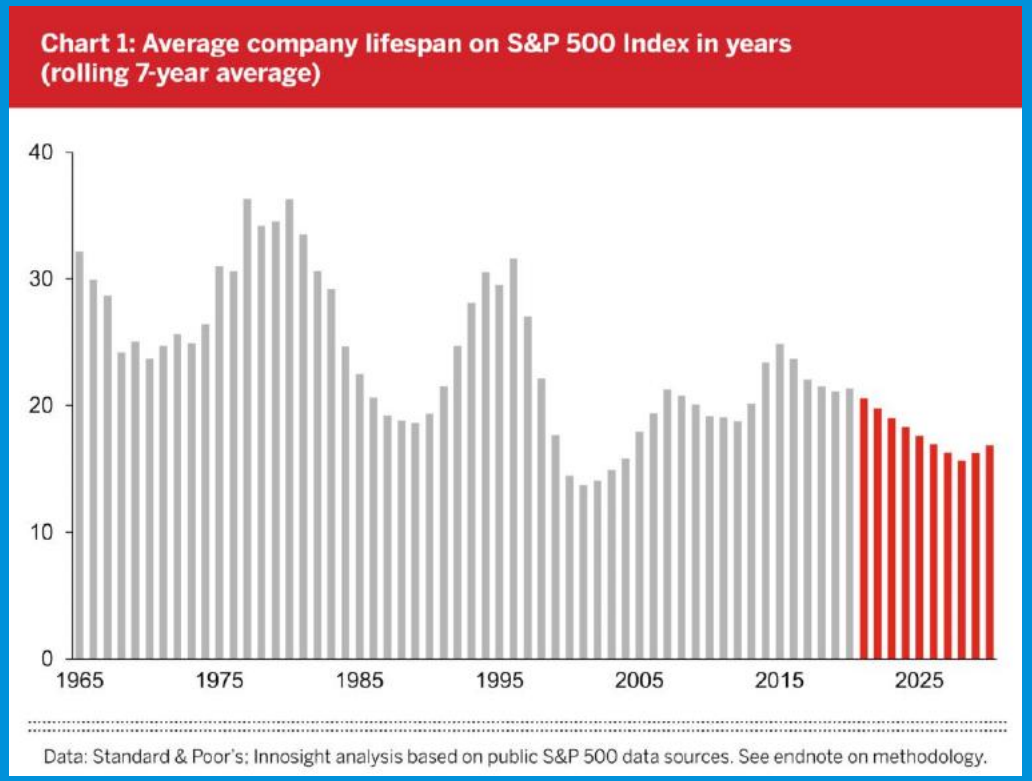
#5

Beware “change or die”

“Software is eating the world.”

“It is not necessary to change. Survival is not mandatory”*

* “Survival is optional. No one has to change,” according to Clare Crawford-Mason via Mark Graban.

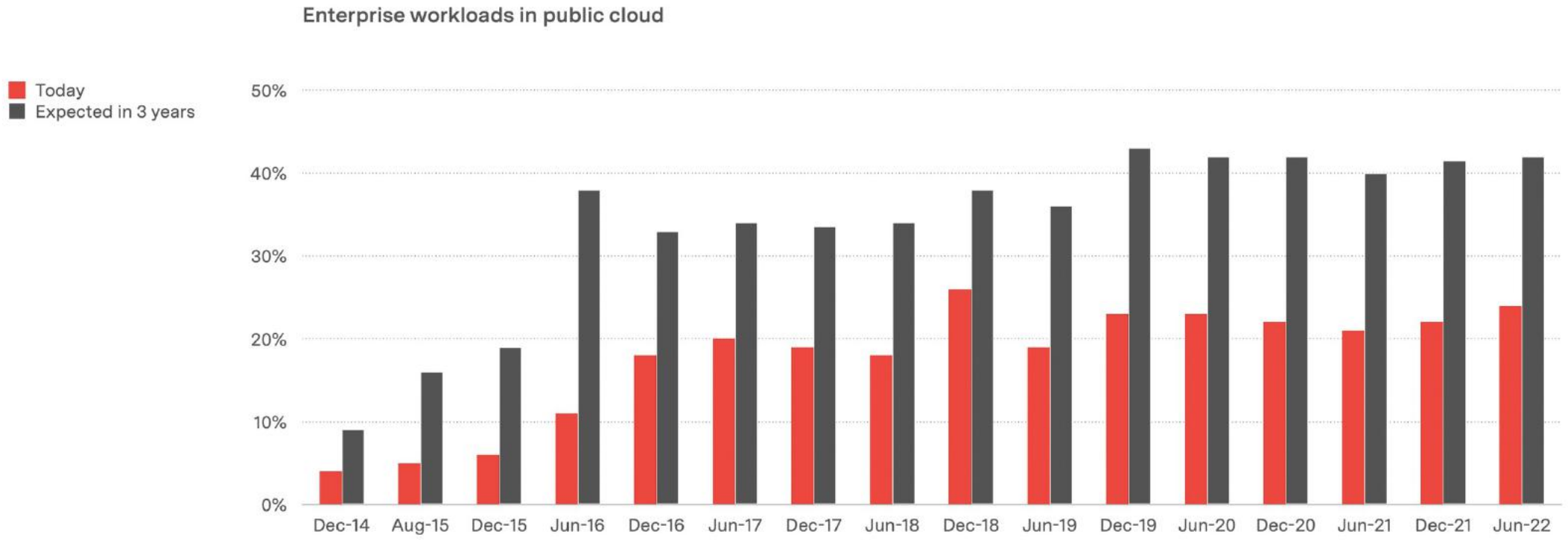


Sources: Innosight's Corporate Longevity Forecast, 2014 to 2021.

Source: DevOpsDays Austin 2015, Jamie Dimon, CEO@JPMC, 2015 letter to shareholders.

The future can take a long time

The cloud might seem like an old and boring idea, but it's still only a quarter of enterprise workloads



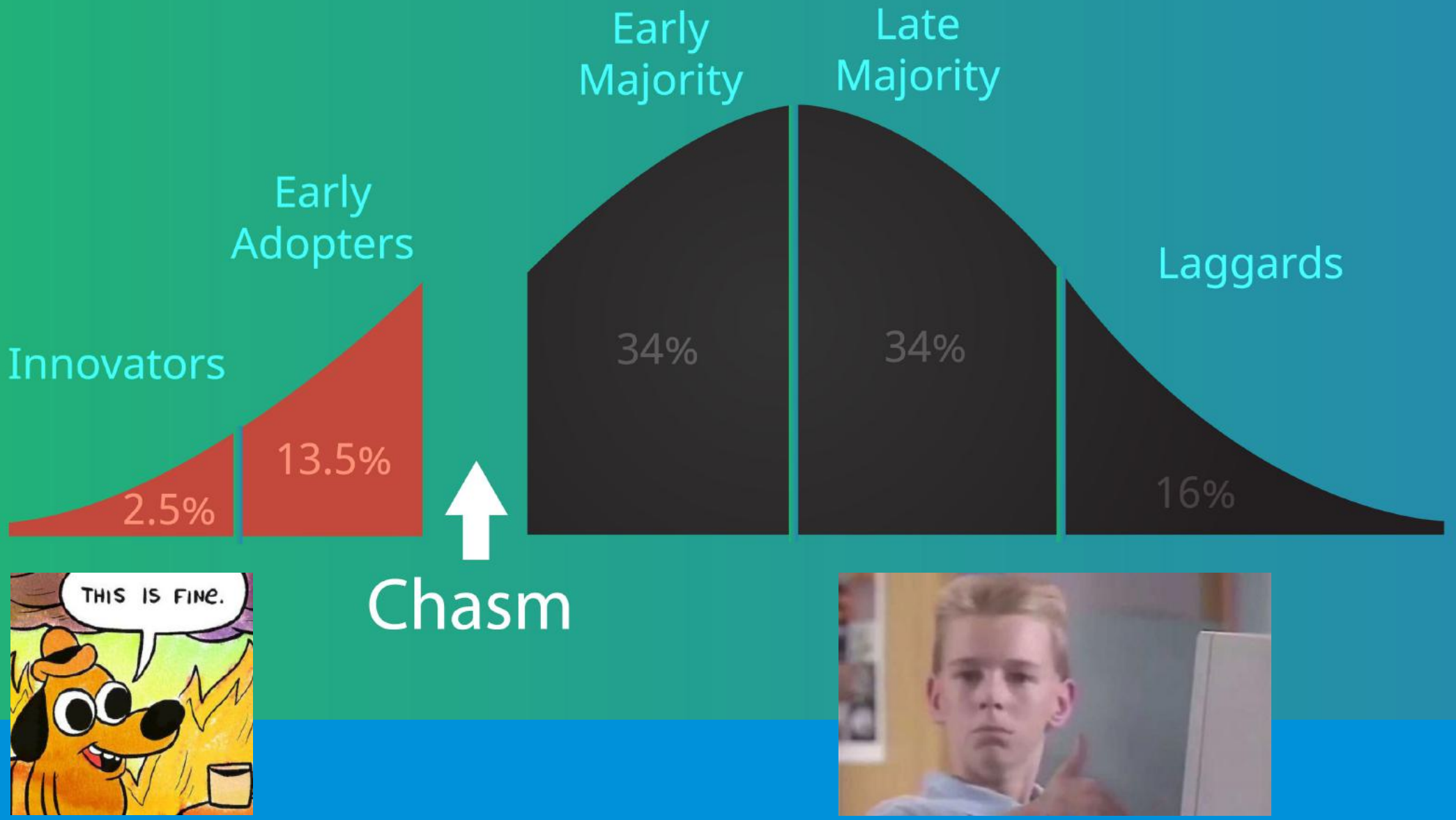
Source: Goldman Sachs CIO Survey

Sources: "AI and the automation of work," Benedict Evans, July 2023.

#9

Be a late adopter

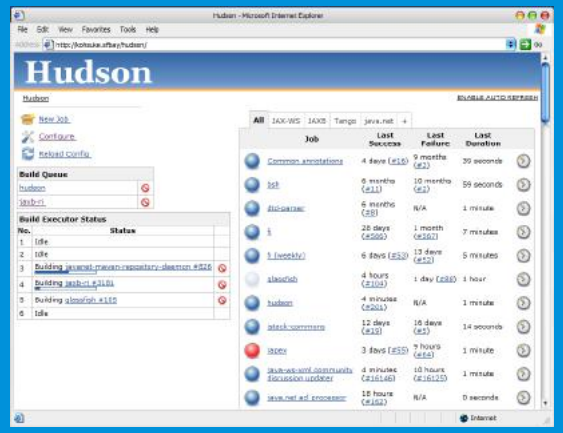
Or, be OK with taking a long time


Early
Majority
Late
Majority
Early
Adopters
Laggards
Innovators
34%
34%
13.5%
16%
2.5%
Chasm

THIS IS FINE.

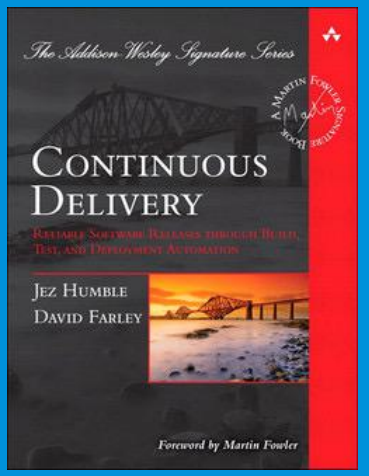
. . . there's a difference between being late adopter and being stubborn

2001

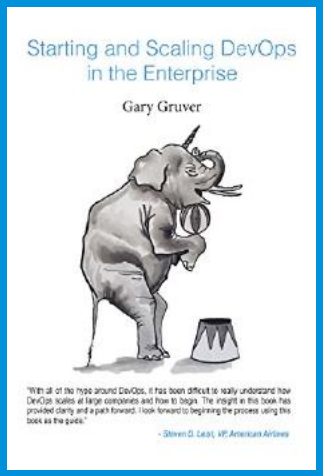


2005

2011

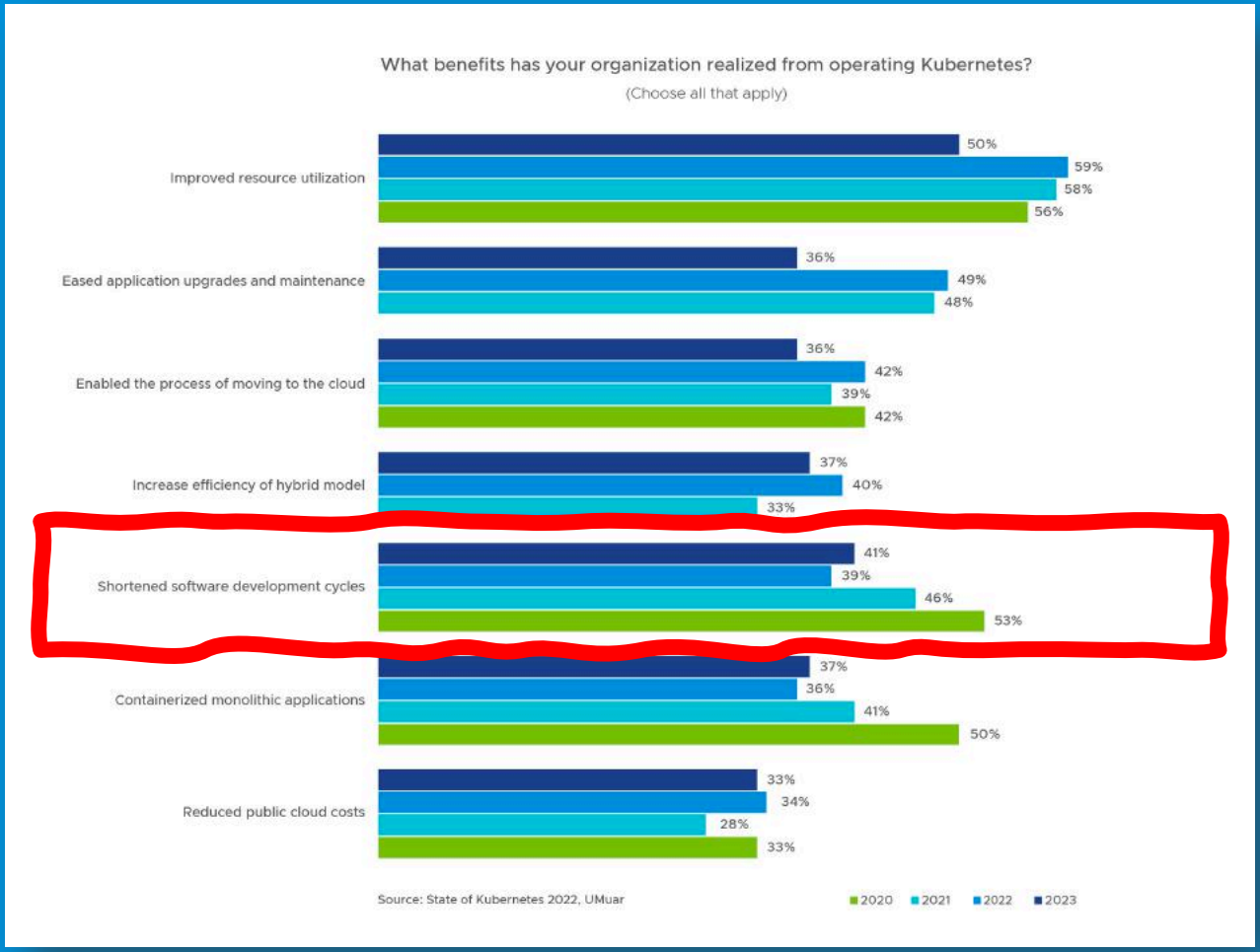


2010

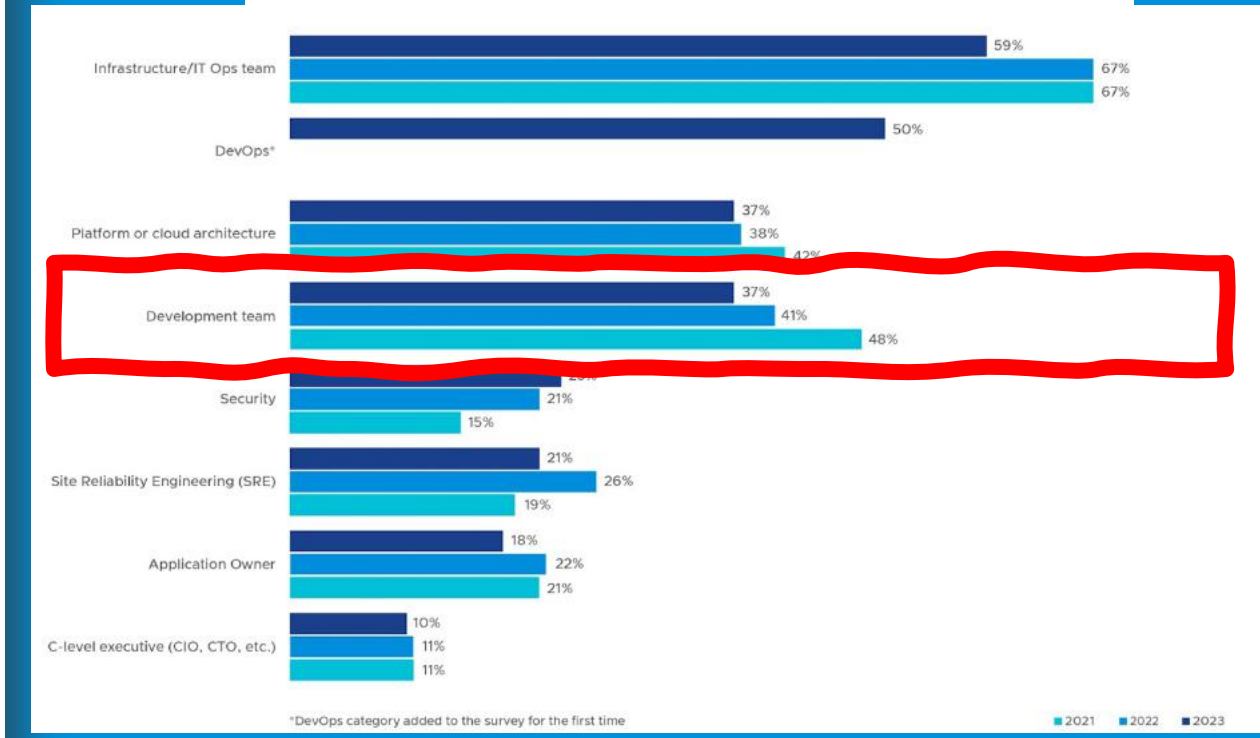


2016

Sources: book listings, Sourceforge(!), Wikipedia on Sep 1st, 2023.

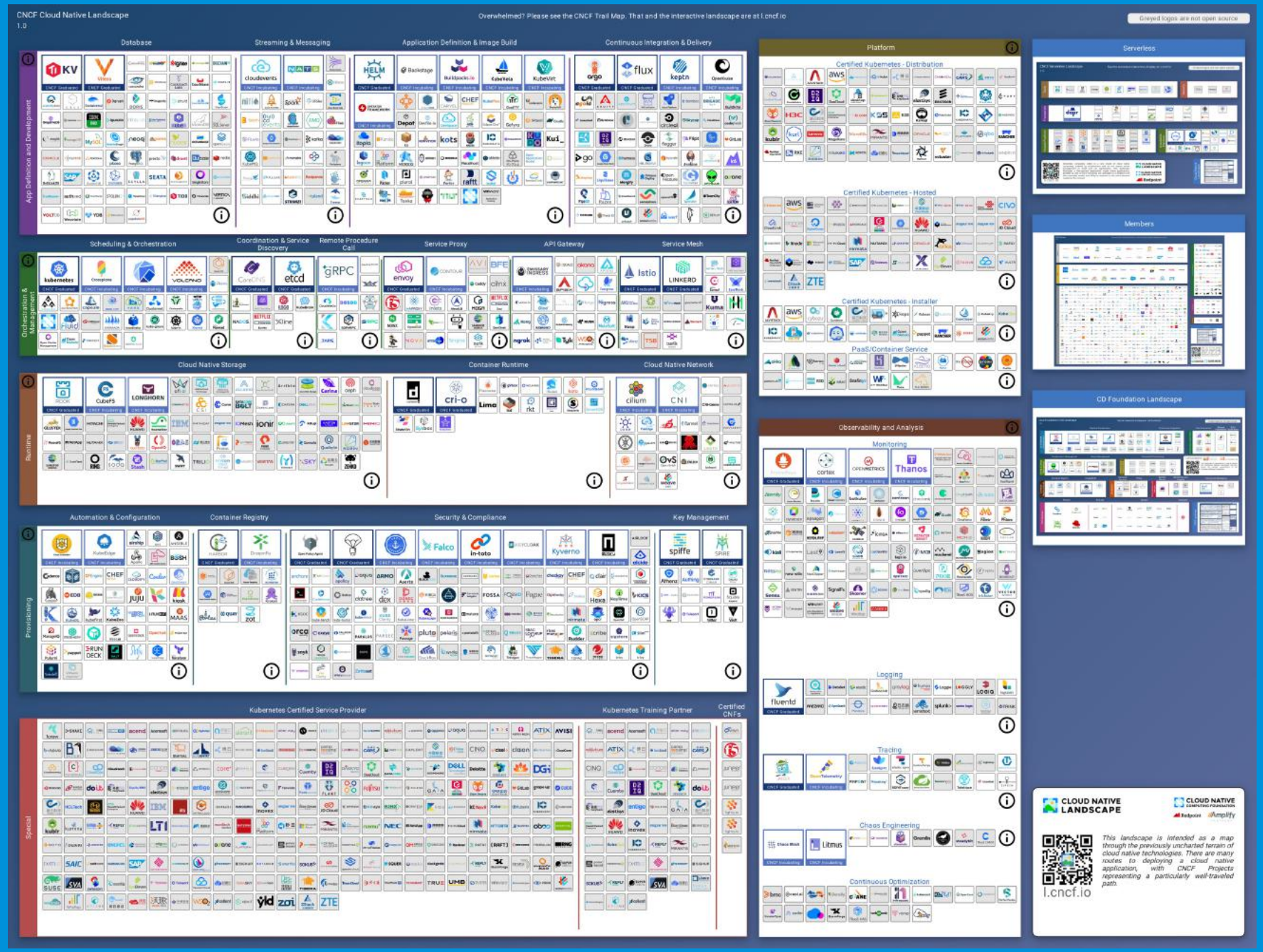


Which teams in your organization own the operation of your Kubernetes infrastructure?



Source: State of Kubernetes 2023, VMware – analysis by Coté.

Source: Torsten Volk, ESG, 2023



Source: CNCF Cloud Native Landscape, retrieved Sep 1st, 2023.

2014

"How do [we] change things up — how do we shake the snow globe in a way that may not be all about Google, but at least gives Google a fighting chance to be able to start grabbing some of these customers, and to start being that balance against the dominance that AWS had at the time."

Joe Beda

2017

"Kubernetes is a platform for building platforms. It's a better place to start; not the endgame."

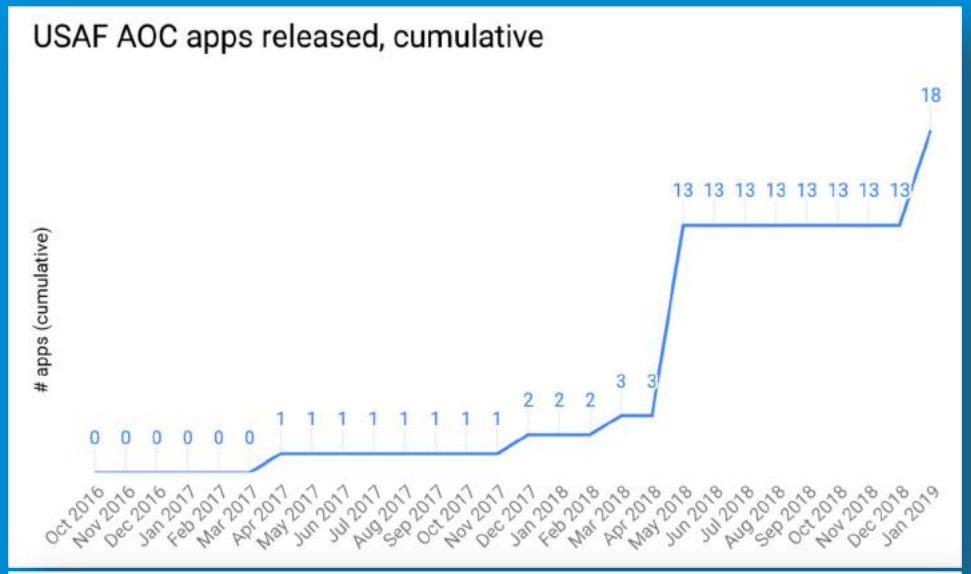
Kelsey Hightower

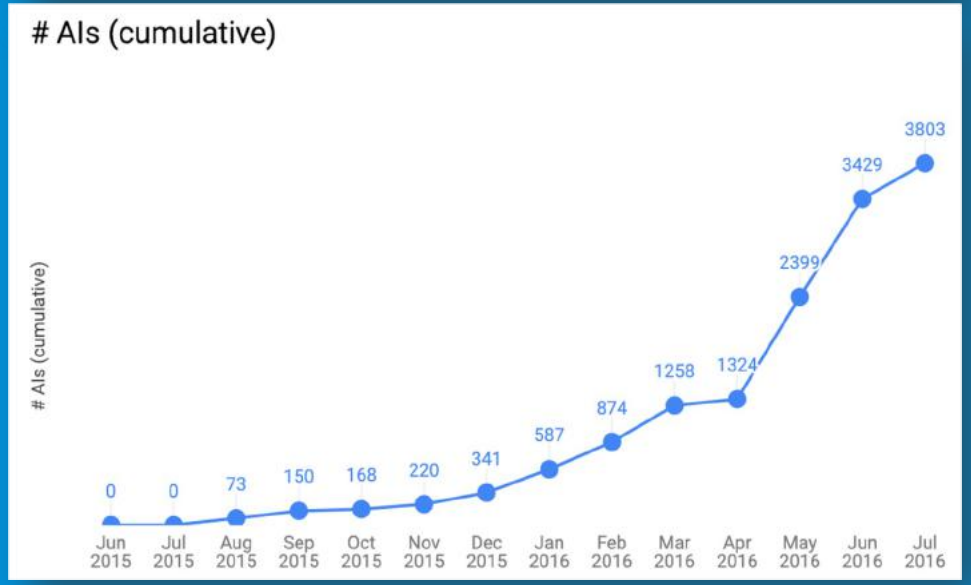
2021

"The initial experience, that 'wall of yaml,' as we like to say, when you configure your first application can be a little bit daunting. And, I'm sorry about that. We never really intended folks to interact directly with that subsystem. It's more or less developed a life of its own over time."

Craig McLuckie

Sources: Kubernetes documentary, X/Twitter, SpringOne 2021.



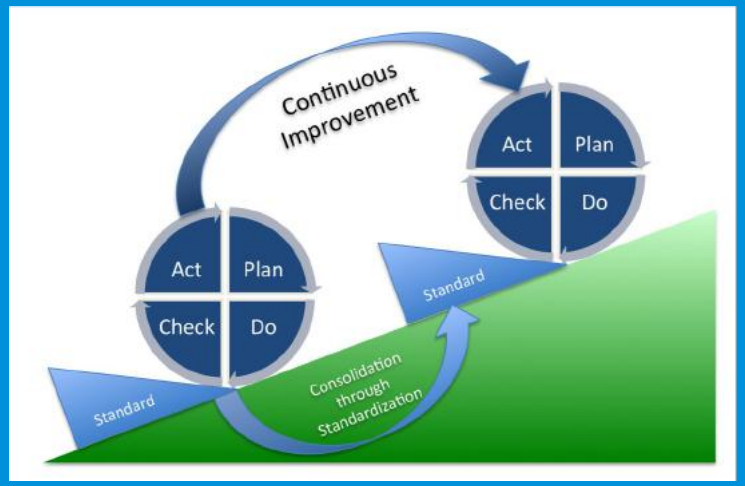


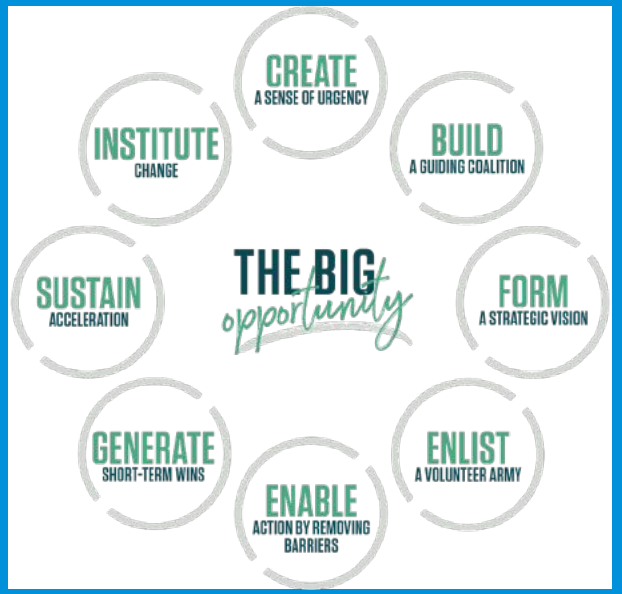
Sources "From 0 to 1000 Apps: The First Year of Cloud Foundry at The Home Depot," Anthony McCulley, The Home Depot, Aug 2016; "Cloud Native at The Home Depot, with Tony McCulley," Pivotal Conversations #45; USAF presentations and write-ups.

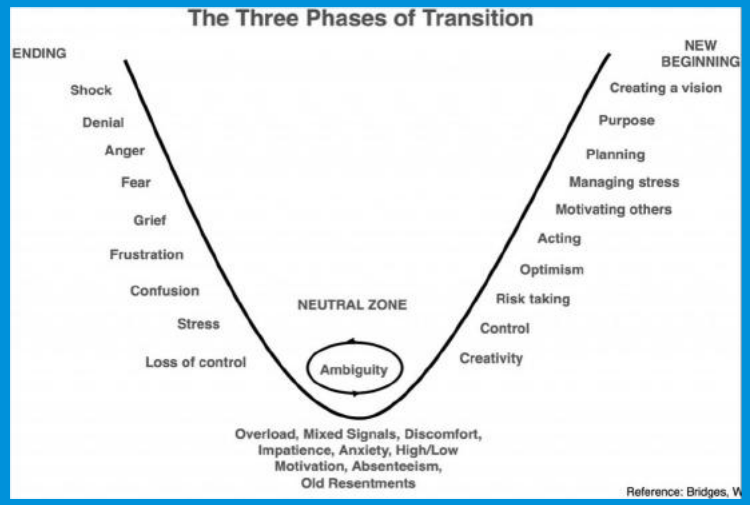
#12

To change, tell people how their lives will be better

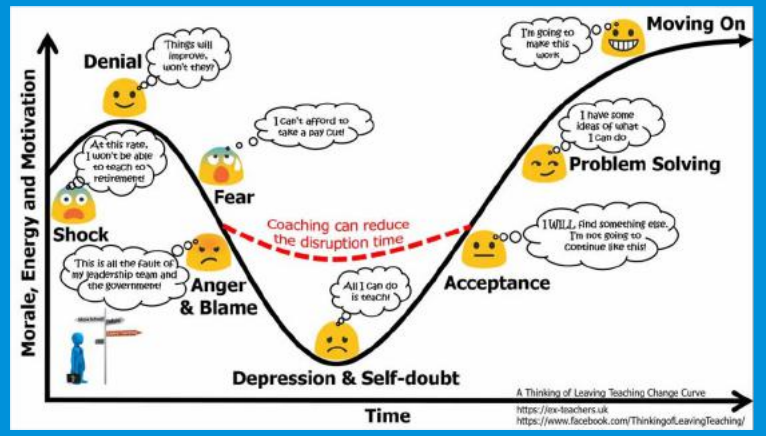
(Among many other things)







Bridges Transition Model



Kübler-Ross Change Curve

Pictures: PDCA from Wikipedia, KotterInc.com; Bridges from Global Leadership Foundation, Kubler-Ross from ex-teachers.uk.

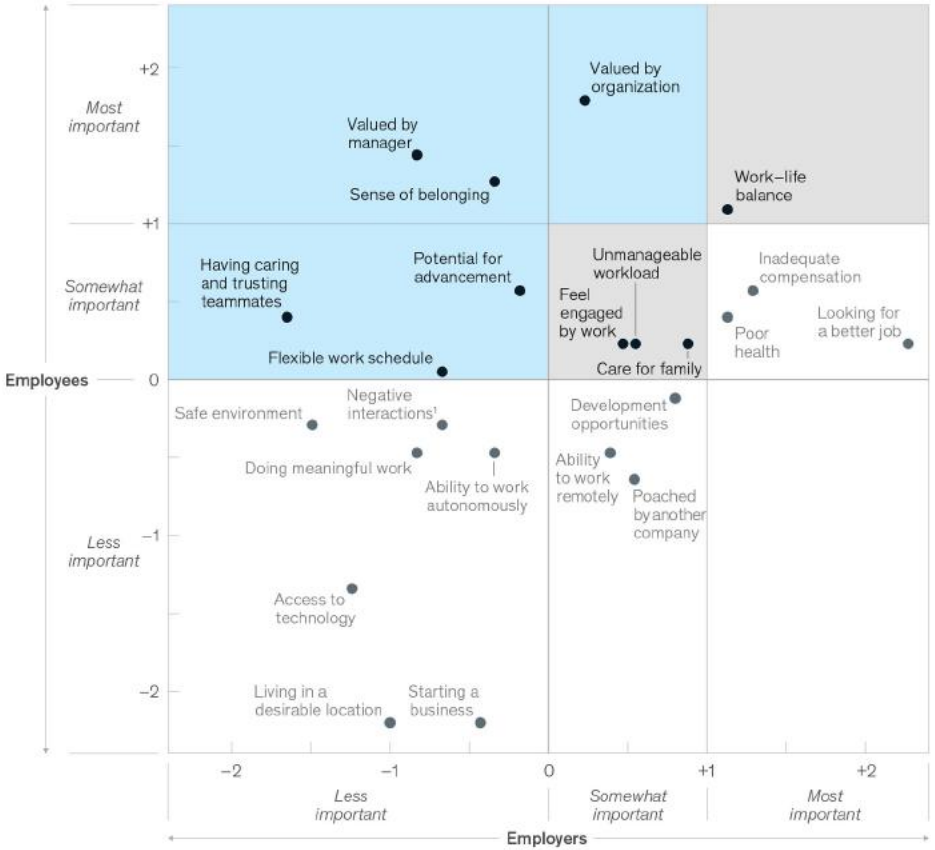
Factors that are important to employees versus what employers think is important

Employer views

▶ **Employers seem to overlook the relational elements** that are key drivers for why employees are leaving, such as lack of belonging or feeling valued at work.

More important to employees than employers appreciate

As important to employees as employers think



Note: Standardized scores are reported for both employee and employer perspectives. Employees were asked to respond to the following question: To what extent did the following factors impact your decision to leave your last job? (Not at all, slightly, moderately, very much, extremely); employers were asked to respond to the following question: Why do you think employees are choosing to leave your organization now? (select all that apply)

[1]Includes clients, customers, patients, and students.

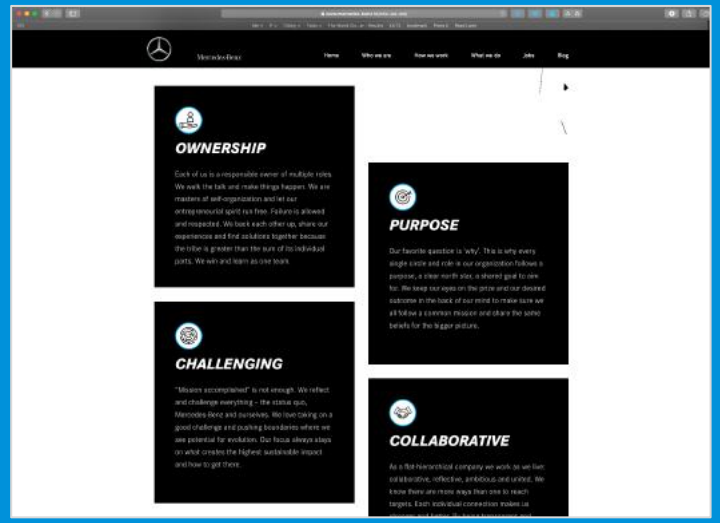
McKinsey & Company

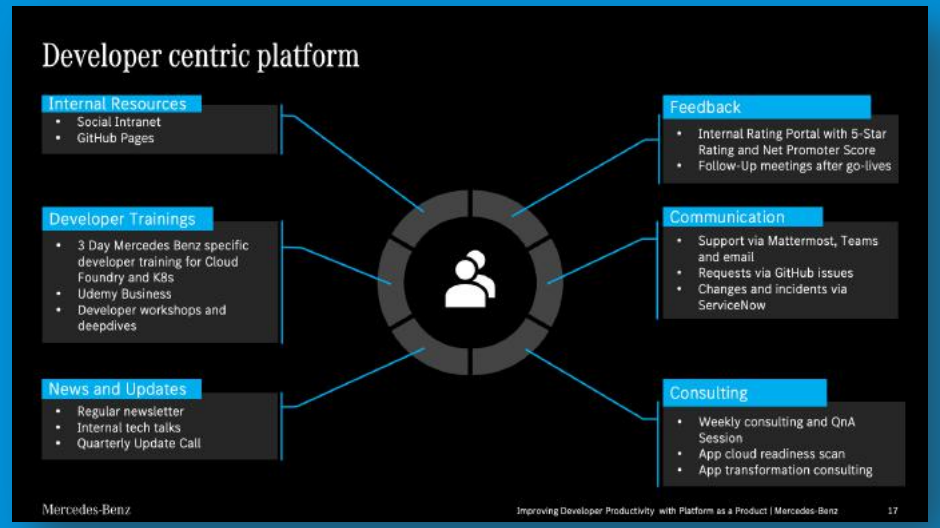
FIGURE 1: **EXAMPLE METRICS**

LEVEL	SATISFACTION & WELL-BEING How fulfilled, happy, and healthy one is	PERFORMANCE An outcome of a process	ACTIVITY The count of actions or outputs	COMMUNICATION & COLLABORATION How people talk and work together	EFFICIENCY & FLOW Doing work with minimal delays or interruptions
INDIVIDUAL One person	*Developer satisfaction *Retention† *Satisfaction with code reviews assigned *Perception of code reviews	*Code review velocity	*Number of code reviews completed *Coding time *# Commits *Lines of code†	*Code review score (quality or thoughtfulness) *PR merge times *Quality of meetings† *Knowledge sharing, discoverability (quality of documentation)	*Code review timing *Productivity perception *Lack of interruptions
TEAM OR GROUP People that work together	*Developer satisfaction *Retention†	*Code review velocity *Story points shipped†	*# Story points completed†	*PR merge times *Quality of meetings† *Knowledge sharing or discoverability (quality of documentation)	*Code review timing *Handoffs
SYSTEM End-to-end work through a system (like a development pipeline)	*Satisfaction with engineering system (e.g., CI/CD pipeline)	*Code review velocity *Code review (acceptance rate) *Customer satisfaction *Reliability (uptime)	*Frequency of deployments	*Knowledge sharing, discoverability (quality of documentation)	*Code review timing *Velocity/flow through the system

† *Use these metrics with (even more) caution — they can proxy more things.*

Source: "'Great Attrition' or 'Great Attraction'? The choice is yours," Aaron De Smet, Bonnie Dowling, Marino Mugayar-Baldocchi, Bill Schaninger, McKinsey, Sep 2021.
"The SPACE of Developer Productivity," Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler, 2021.





Sources: BT Canvas team; MB.io; Duke Energy; Allstate; "Take DevOps to 11 and Sprinkle Cloud on it with Rainbows and Unicorns," Matt Curry, s1p 2017. "Improve Developer Productivity with Platform as a Product," VMware Explore, Nov. 2022.

#14

Make sure your customer is a human, not a dashboard.

Or, “obligatory platform engineering comment”

“We are building this platform not for us, we are building it for Mercedes-Benz developers.”

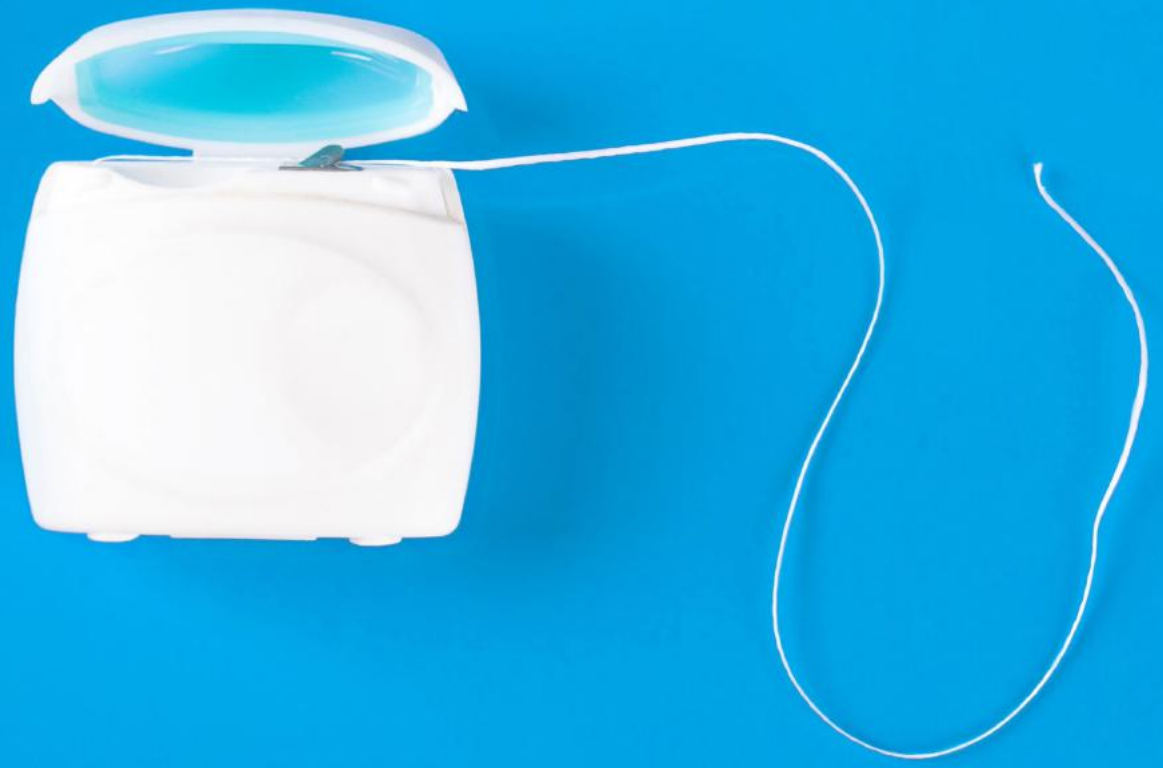
Thomas Müller, Mercedes-Benz

Find the Developer Toil, Confusion, Blockers

Find the Developer Toil, Confusion, Blockers

- What are we making?
- We have a strong vision for our product, and we're doing important work together every day to fulfill that vision.
- I have the context I need to confidently make changes while I'm working.
- I am proud of the work I have delivered so far for our product.
- I am learning things that I look forward to applying to future products.
- My workstation seems to disappear out from under me while I'm working.
- It's easy to get my workstation into the state I need to develop our product.
- What aspect of our workstation setup is painful?
- It's easy to run our software on my workstation while I'm developing it.
- I can boot our software up into the state I need with minimal effort.
- What aspect of running our software locally is painful? What could we do to make it less painful?
- It's easy to run our test suites and to author new ones.
- Tests are a stable, reliable, seamless part of my workflow.
- Test failures give me the feedback I need on the code I am writing.
- What aspect of production support is painful?
- We collaborate well with the teams whose software we integrate with.
- When necessary, it is within my power to request timely changes from other teams.
- I have the resources I need to test and code confidently against other teams' integration points.
- What aspect of integrating with other teams is painful?
- I'm rarely impacted by breaking changes from other tracks of work.
- We almost always catch broken tests and code before they're merged in.
- What aspect of committing changes is painful?
- Our release process (CI/CD) from source control to our story acceptance environment is fully automated.
- If the release process (CI/CD) fails, I'm confident something is truly wrong, and I know I'll be able to track down the problem.
- What aspect of our release process (CI/CD) is painful?
- Our team releases new versions of our software as often as the business needs us to.
- We are meeting our service-level agreements with a minimum of unplanned work.
- When something is wrong in production, we reproduce and solve the problem in a lower environment.

Sources: "Developer Toil: The Hidden Tech Debt," Susie Forbath, Tyson McNulty, and Coté, August, 2022. See also Michael Galloway's interview questions for platform product managers.



DevOps is like flossing...?

Thanks!

 https://newsletter.cote.io

 cotem@vmware.com