

Slides, etc.

IncrativeOps

or, don't let software eat you, eat software instead

or, notes from a recovering thought-leader

or, how dungeons & dragons natural 20'ed my love of tech

Coté - September 8th, 202

THENEWSTACK

PODCASTS EBOOKS EVENTS NEWSLETTER

ARCHITECTURE ENGINEERING OPERATIONS

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



Hello.

Welcome to my ongoing therapy session

3

Q

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, you must slowly build up trust and word-of-mouth.
- 13. Focus on outcomes over activities
- 14. Make sure your customer is a human, not a dashboard.

DevOps is like flossing...

Hello, l'm Coté

I work at VMVare®

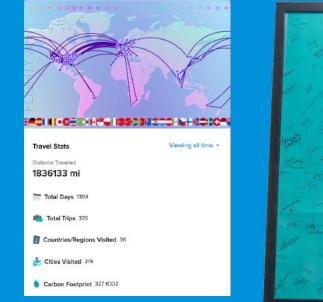


... and, l'marecovering thought-leader

How its started.



Photo: Simon Phipps, May, 2006







OREILLY Changing Mindsets: The Missing Ingredient to Digital Transformation Modernizing Software Creation for Large Enterprises Michael Coté

REPORT

The Business Bottleneck Brecking Through the Last Barrier Digital Transformation Michael Cote

O'REILLY



[INSERT PICTURES GESTUICULATING AT PODIUMS]

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



How it's going.





THENEWSTACK

PODCASTS EBOOKS EVENTS NEWSLETTER

ARCHITECTURE ENGINEERING OPERATIONS

DEVOPS / PLATFORM ENGINEERING / SOFTWARE DEVELOPMENT / TECH LIFE

DevOps Is Dead. Embrace Platform Engineering

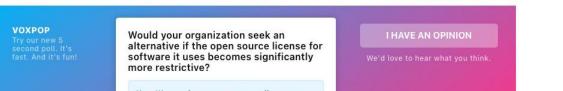
Platforms provide golden paths, with recommended tools and best security practices built in, reducing cognitive load while preserving developer freedom.

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

 \equiv



Image via Unsplash.



Yes. We require an open source license so

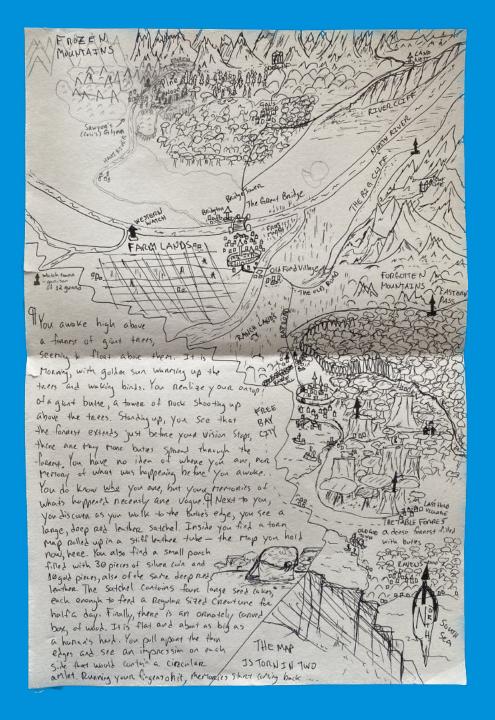
15

Q

"I know my price Because I developed my identity outside of work, there's a cost that if work cuts into it - if it ever costs me a larger part of my identity and my life I know it's not worth it."

From The Good Enough Job.





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, you must slowly build up trust and word-of-mouth.
- 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!

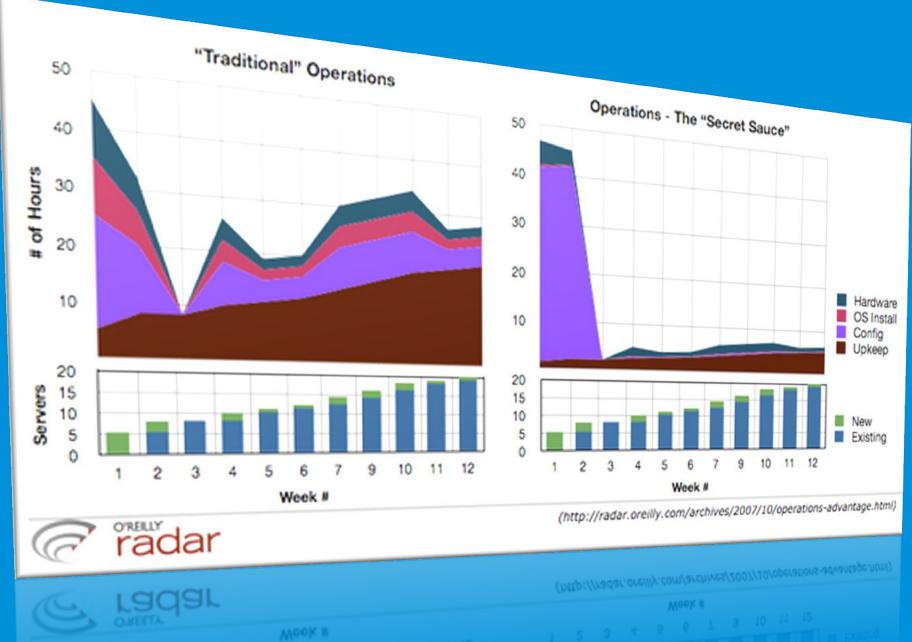
Copyright The Flow Consortium its and Nagel Thurkow 2017 converde. All rights reserved. Confidential Distribution, popyling or commercial use prohibited without permission.

HLOW



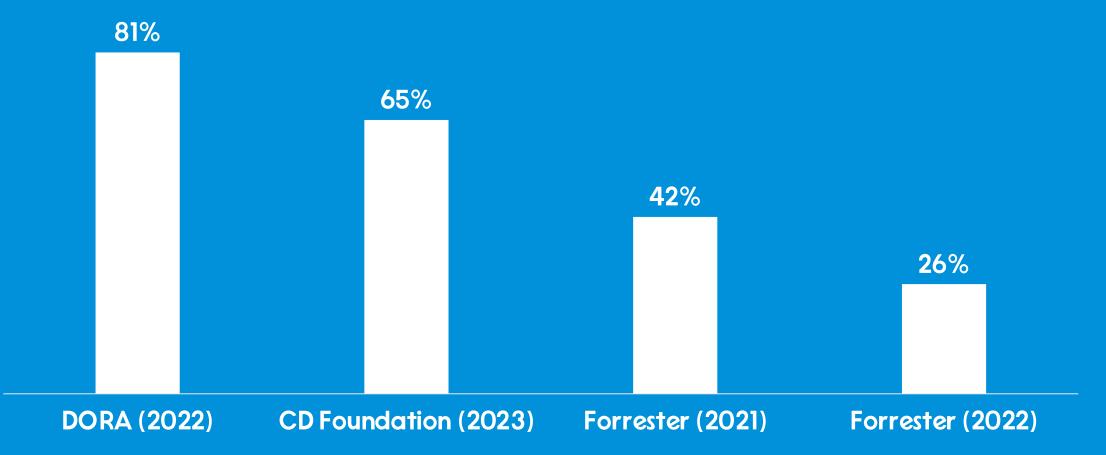


CI/CD, but for real this time



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

Accounts of deployment rates vary wildly



Deploy Monthly or Less

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

CI and CD usage, 2007 to 2021



Source: State of Agile Surveys, VersionOne/CollabNet/digital.ai

Ocote

#5

Beware "change or die"

Or, your business likely won't be "disrupted" if you just avoid being dumb shit & instead be smart We forget all the startups that failed.

"Software is eating the world."

"It is not necessary to change. Survival is not mandatory"*

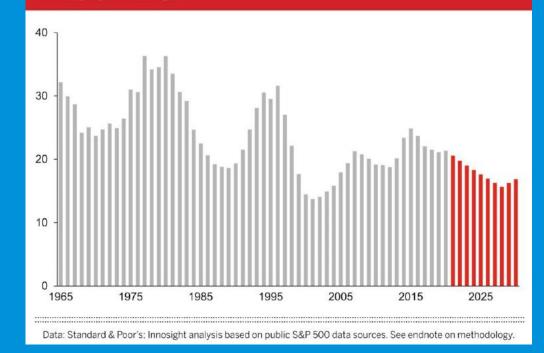
S&P 500 Churn Over the Past Decade

Sample companies that have entered and exited the index since 2002





Chart 1: Average company lifespan on S&P 500 Index in years (rolling 7-year average)



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

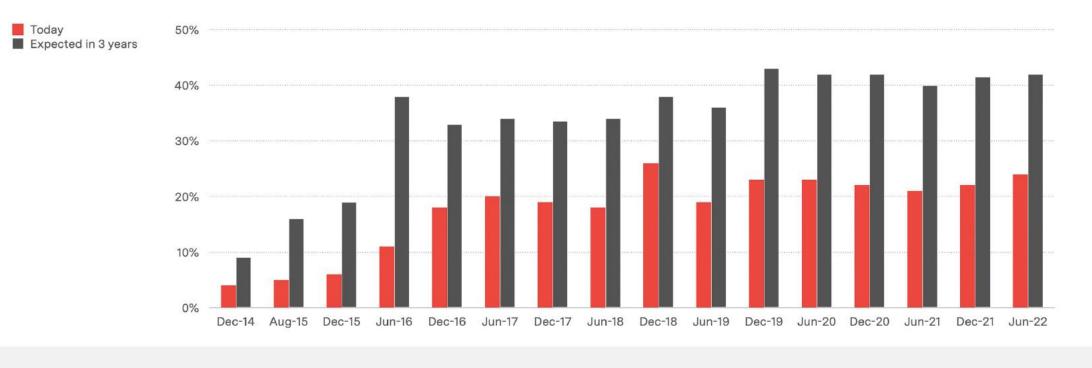


Source: DevOpsDays Austin 2015, Jamie Dimon, CEO@JPMC, 2015 letter to shareholders.

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





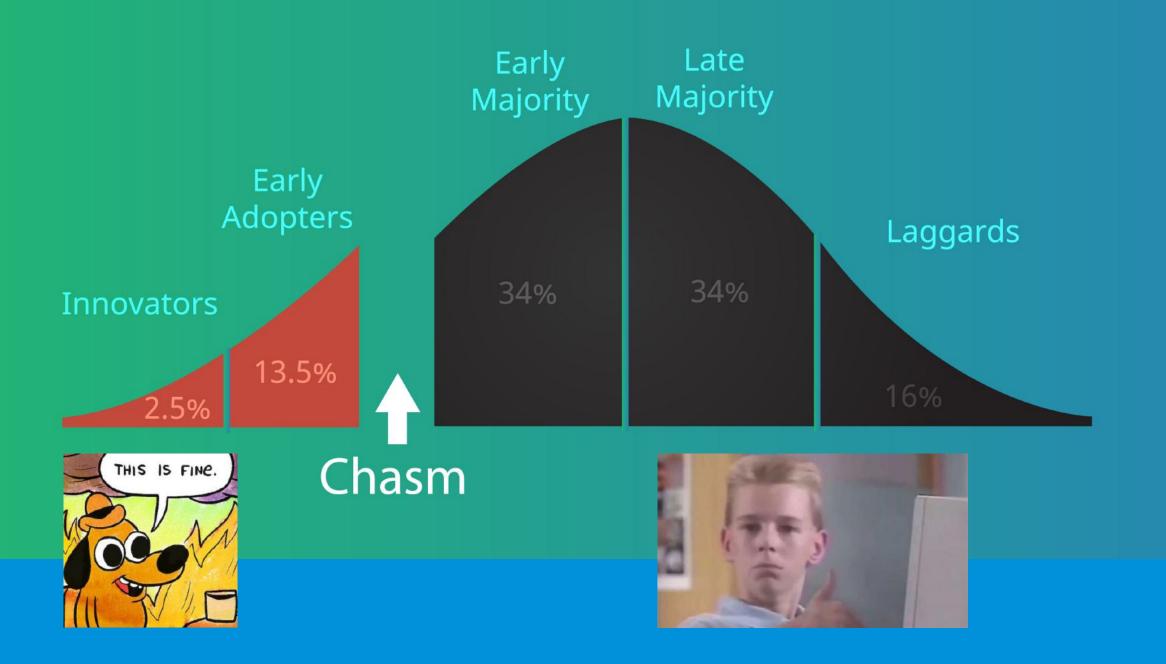
Source: Goldman Sachs CIO Survey

Benedict Evans -- February 2023 73



Be a late adopter

Or, be OK with taking a long time



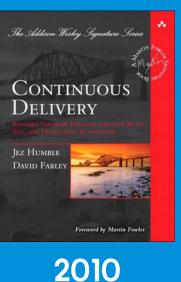
...there's a difference between being late adopter and being stubborn

Project octive release jobt 4 Bulle 2008 09 10101 35 01 000 05 00 Worths Finder	
la (di Yaw Higtory Qostraetis (lock 1949) 🚔 • 🍻 • 😰 🏠 🏠 🚺 Hig (schwichsauftworks constantional Build (schul) (schwichsauf schi). 4	L2 tour send @+
Server : ccweb01.thoughtworks.com	(C cruisecontrol.
Control Ruites	
cclive-release-jdk1.4 Failed	Latest Builds 🔯 🗃
tuid 1 time: 10 569 2008 01:35 GMT Duration: 19 minutes 22 Geconds Let: successful bolts: 2 days 22 tours 7 minutes 13 Seconds age	0 10 Sep 2000 01: 25 GMT -05: 00 9 Sep 2000 22:21
	GMT-05.00 9 Sep 2008 21.81 GMT-05.00
Test Suites	 O f Sep 2009 03:27 GNT -05:00 bail.6
nes sourcefronge pruse control Autribute Test Tests: 3. Feilures: 0. Errors: 0. Duration: 0.376 nes sourceforge cruse control suit bacquinformationsuitcertest	GMT -05:00 build.5
Tests: 7, Falives: 0, Bross: 0, Duration: 0,346 net: sourceforge-crustectarchiel & diduligner buffenkina gerTest Tests: 2, Falives: 0, Bross: 0, Duration: 0,039 net: sourceforge-crustectarchiel & diduced Test	C 33 Aug 2008 34 10 GMT -05 00 build.+
net: Sourcetrage, characterized and planet text Tests: 2, Palmes: 0, Britos 0, Caralanis: 0, 039 net: Sourcetrage, characterized au identifiest Tests: 3, Palmes: 0, Britos 0, Caralanis: 0, 024	9 2+ 3/1 2008 03: 31 GMT -05: 00 23 3/1 2008 20: LR
Tests & Fouries 0, errors 0, ourietoni cute+ net sourceting, crusecutori crusecutricicon/etitochetest Tests 8, Follures 0, Errors 0, Ourietoni 0,229 net sourceting, crusecutori CruseControlCon/epreConffest	GMT -05:00
A a	GMT -05.00 GMT -05.00 GMT -05.00 GMT -05.00
	9 23 Jul 2008 10 42

ACCHES A Inter/Acheske.after/hudsen/						1
Hudson					-	l
Hudson						
Hathon				8	ENABLE AUTO	
😁 New Job	12	All JAX-WS JAXS Tang	i java.net +			
X Contaire		dot	Last Success	Last Failure	Last	
Steland Config.	0	Common anistations	4 days (#16)	9 months (#2)	30 seconde	
Build Queue	0) ist	6 mantha	10 mentha	59 seconds	
imb-ri Q	0	dis-parase:	6 menths	R/A	1 mitute	
Build Executor Status		ALCORDA.	(#8)	22		
No. Status	6	ž.	28 days (#505)	1 month (#367)	7 minutes	
z tole	0	3 (weekty)	6 stays (#53)	13 dava (#52)	5 minutes	
3 Building jasanat-maxan-repository-deamon #026	0		4 hours	1 day (mill)		
4 Building japping #101	0	alaodish	(#104)	1 day (1988)	1 hour	
3 Eviding <u>alsoniph ellis</u> 6 Ide	0	hadson	4 minutes (#201)	N/A	1 minute	
	6	ateck commany	12 days (#19)	16 days (#1)	14 seconds	
		HADRY	3 days (#55)	7 hours	1 minute	
	C	tionesion undeter	d minister (#16146)	10 hours (#16125)	1 minute	
	0	invalvel ad processor	15 hours (#162)	N/A	D seconde	



2011



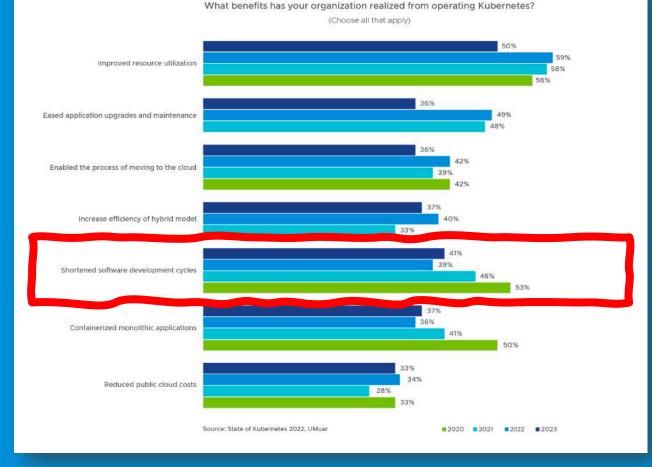


Starting and Scaling DevOps in the Enterprise Gary Gruver

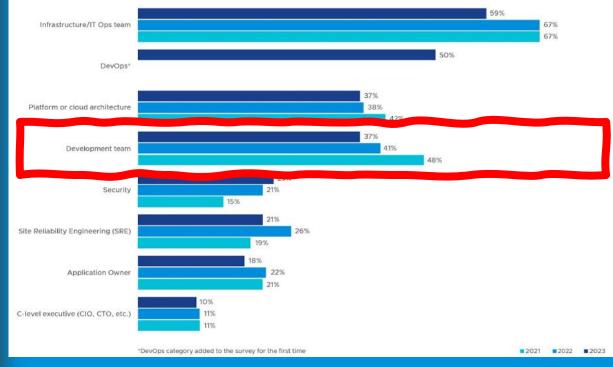
2016

2001

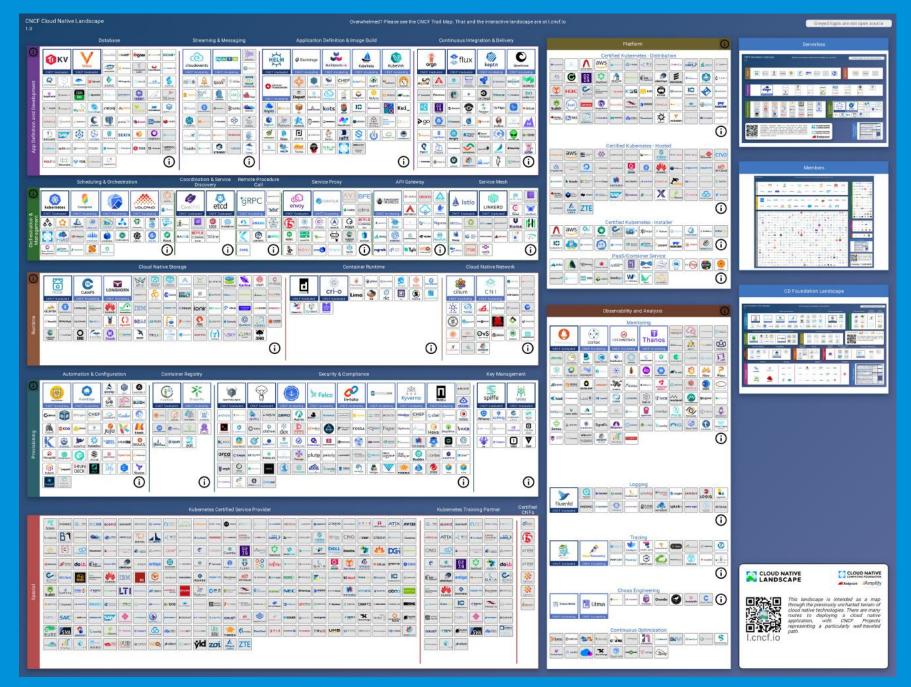
2005



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







Source: <u>CNCF Cloud Native Landscape</u>, retrieved Sep 1st, 2023.

<u>2014</u>

"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**

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

2017

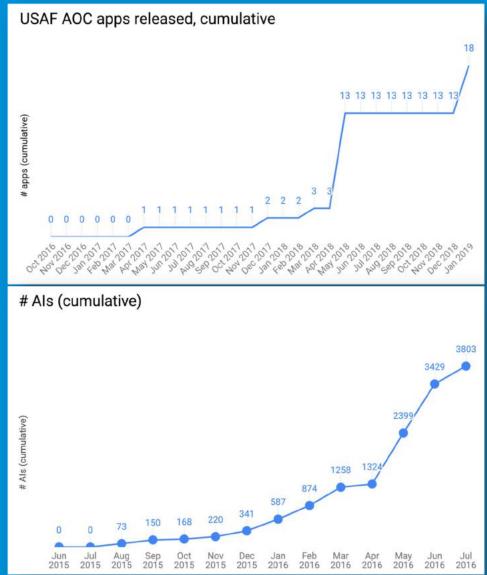


"The initial experience, that 'wall of yaml,' as we like to say, when you configure your first application can be a little bit daunting. And, I'm sorry about that. We never really intended folks to interact directly with that subsystem. It's more or less developed a life of its own over time."

Kelsey Hightower

Craig McLuckie





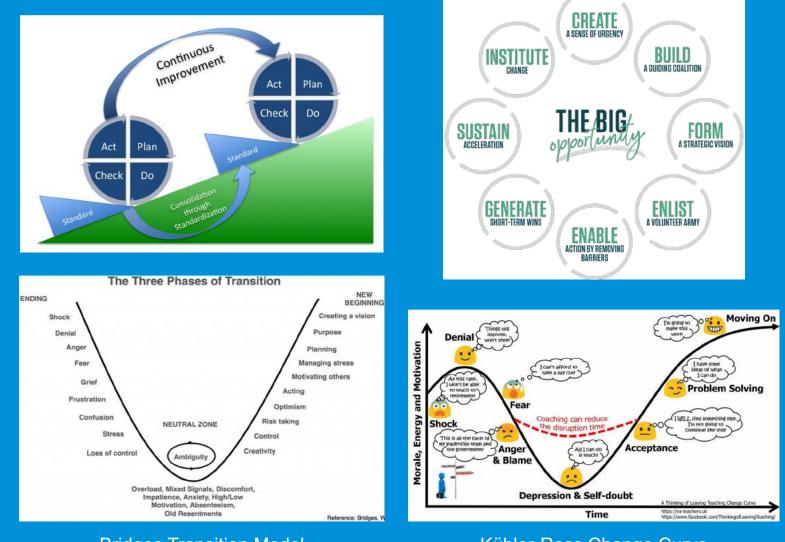
Sources <u>"From 0 to 1000 Apps: The First Year of Cloud Foundry at The Home Depot,"</u> Anthony McCulley, The Home Depot, Aug 2016; <u>"Cloud Native at The Home Depot, with Tony McCulley," Pivotal Conversations #45; USAF presentations and write-ups</u>.

#12

To change, slowly build up trust and word-of-mouth.

(Among many other things)

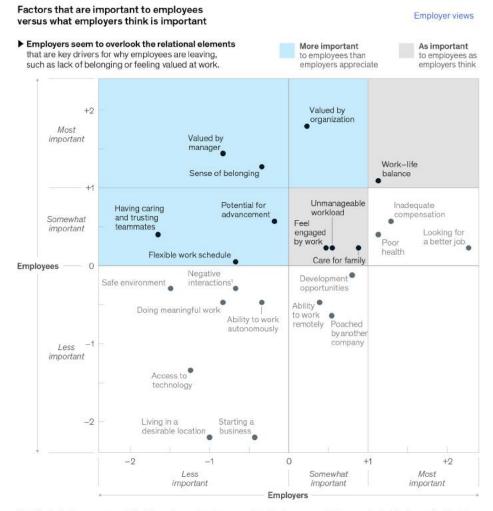




Bridges Transition Model

Kübler-Ross Change Curve

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



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) 'Includes clients, customers, patients, and students.

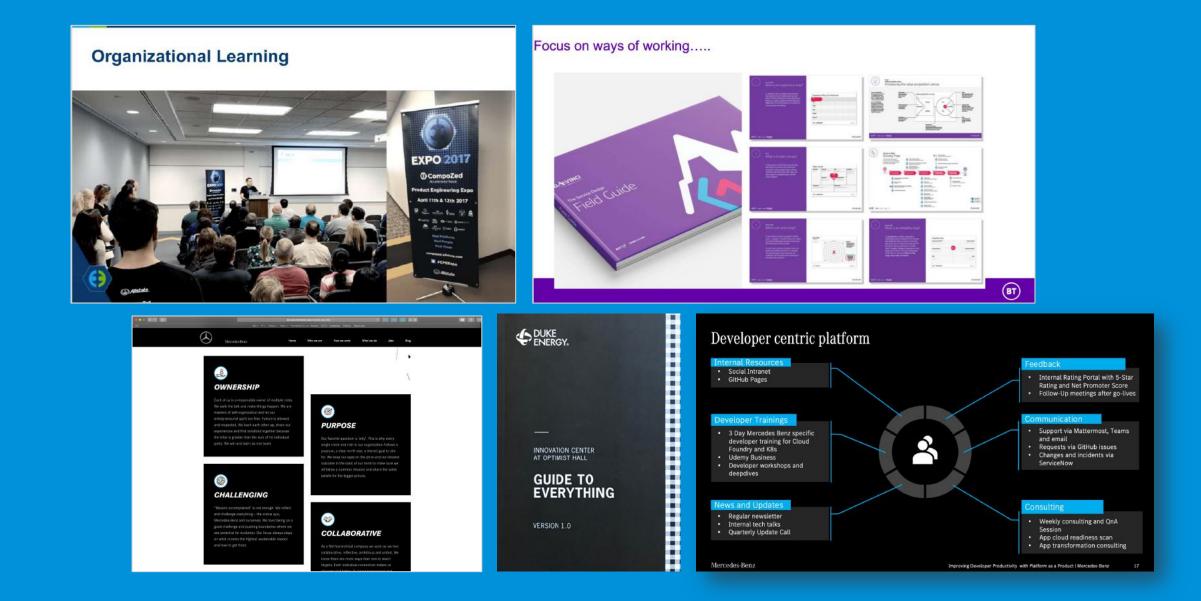
McKinsey & Company

14 CE 29 productivity UNIFICAR S OF LEARNING a takend work toget CONTRO SCHORSOF OUTPUT How Street Ball Ball Ball FIGURE 1: EXAMPLE METRICS SUSPERIOR & WELTERING EFFERENCE & FLOW Awith' LEVEL 80 INDIVIDUAL *Code *Number *Code review *Code review *Developer One satisfaction review of code score (quality or timing thoughtfulness] person *Retention¹ *Producvelocity reviews *PR merge times *Satisfaction completed tivity with code *Coding time *Quality of perception *# Commits meetings *Lack of reviews assigned *Lines of *Knowledge inter-*Perception of codet sharing. ruptions code reviews discoverability (quality of documentation) TEAM OR *Developer *Code *# Story *PR merge times *Code review GROUP satisfaction *Quality of review points timing People *Retention[†] velocity completed[†] meetingst *Handoffs that work *Story points *Knowledge together shipped sharing or discoverability (quality of documentation) SYSTEM *Code review *Frequency *Satisfaction *Knowledge *Code review End-towith velocity of deploysharing, timing end work engineering *Code review discoverability *Velocity/ ments through system (e.g., CII [acceptance (quality of flow a system CD pipeline) rate] documentation) through the (like a *Customer system develsatisfaction opment *Reliability pipeline) (uptime)

[†] Use these metrics with (even more) caution – they can proxy more things.

acmqueue | january-february 2021 14

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



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



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

DevOps is ike flossing...?

Thanks!

<u>https://newsletter.cote.io</u>



