



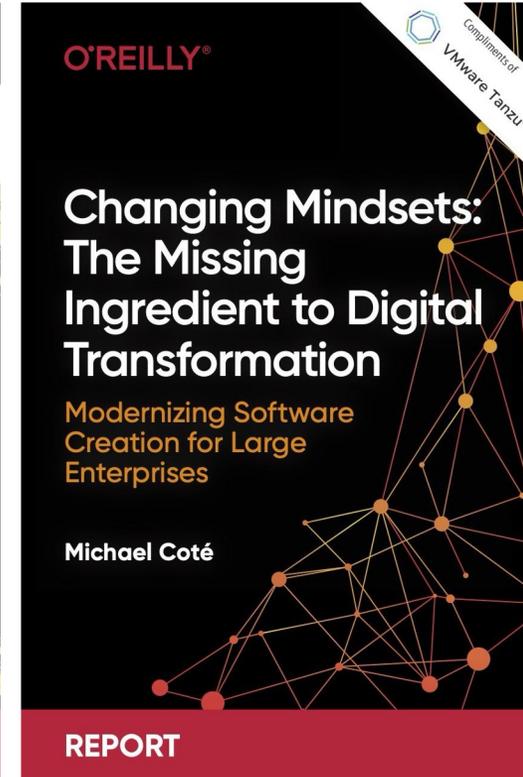
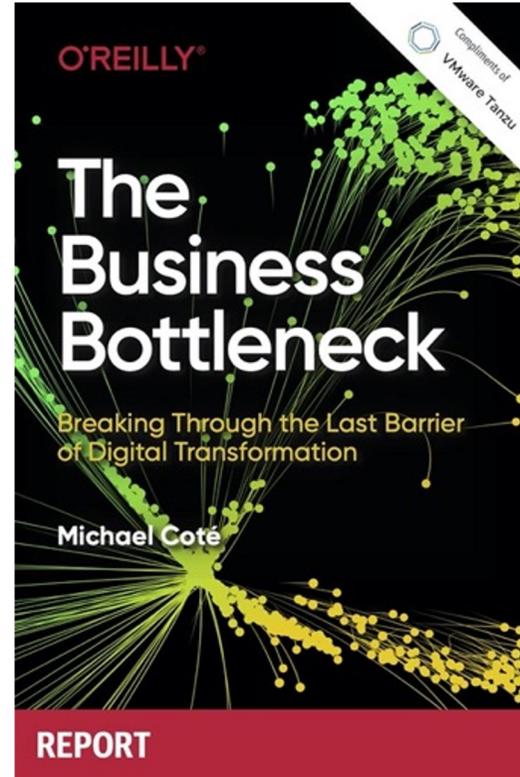
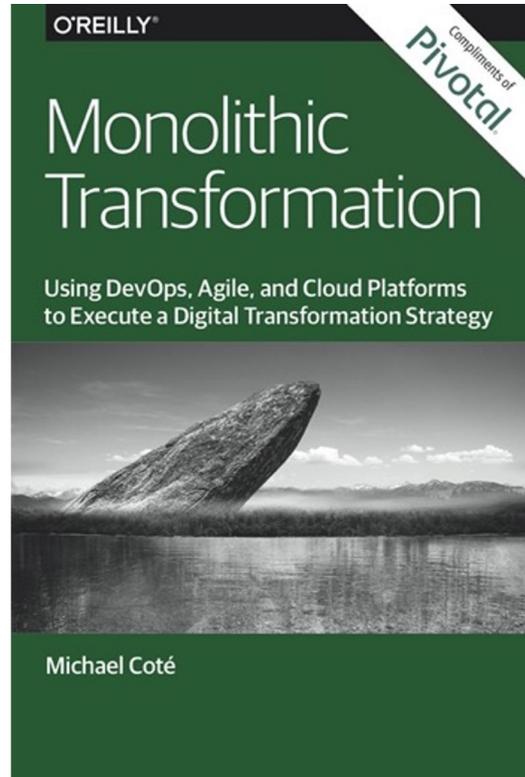
**We fear change.**

**Understanding why people resist using your platform**

Coté - February 6th<sup>th</sup>, 2024

# Coté

<https://newsletter.cote.io/> | [cote@broadcom.com](mailto:cote@broadcom.com)

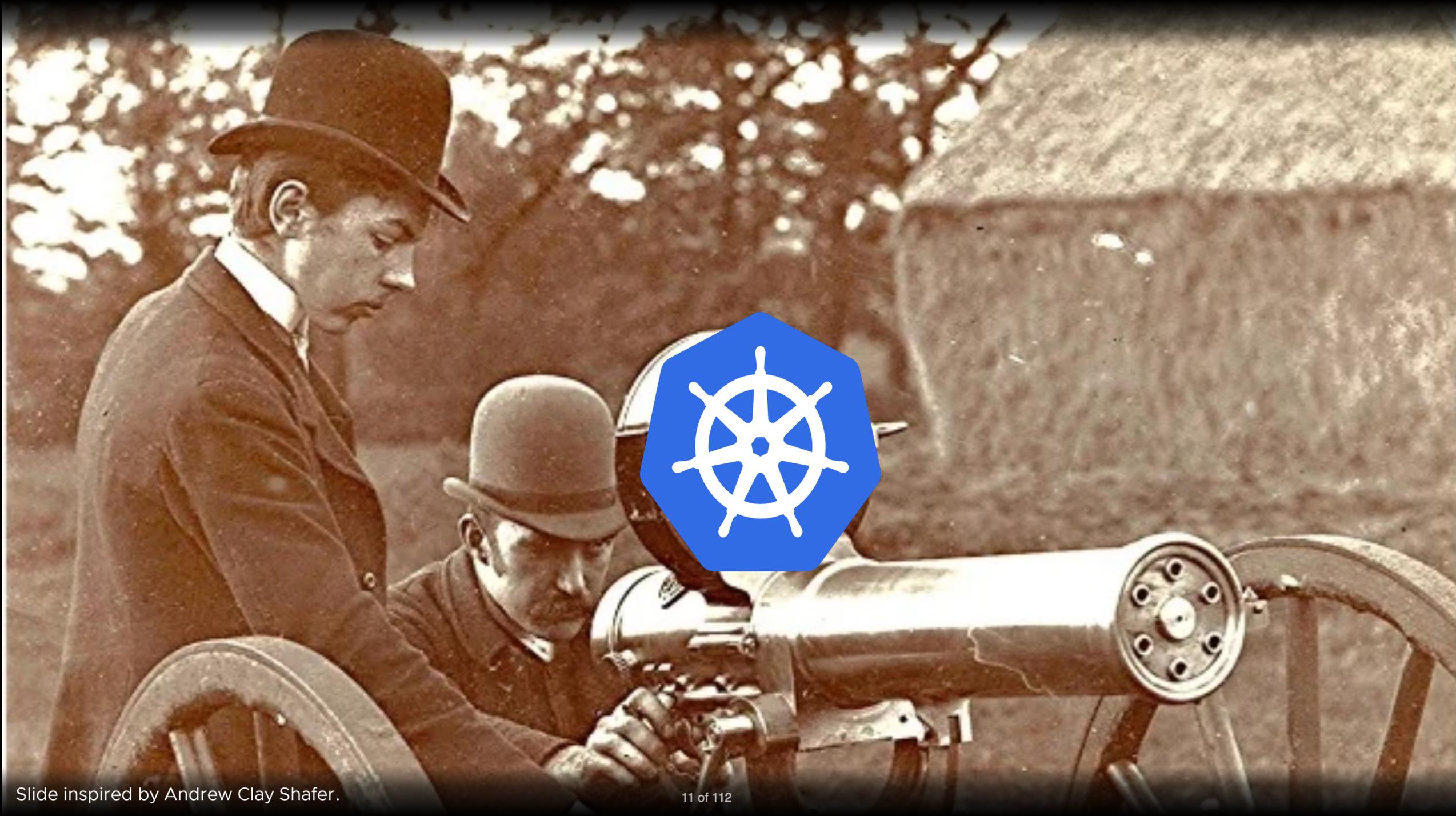


...and,  
I'm a recovering  
thought-leader

Yes,



But





**Error:  
No Thoughts  
Found**

We all know that

**Changing organizations fails 70% of  
the time.**

Actually,

**We have no idea how frequently  
changing organizations succeeds or  
fails.**

We all know that

**Technology is easy.**

**People are hard.**

Actually,

**Technology is very hard!**

**(People are hard too.)**

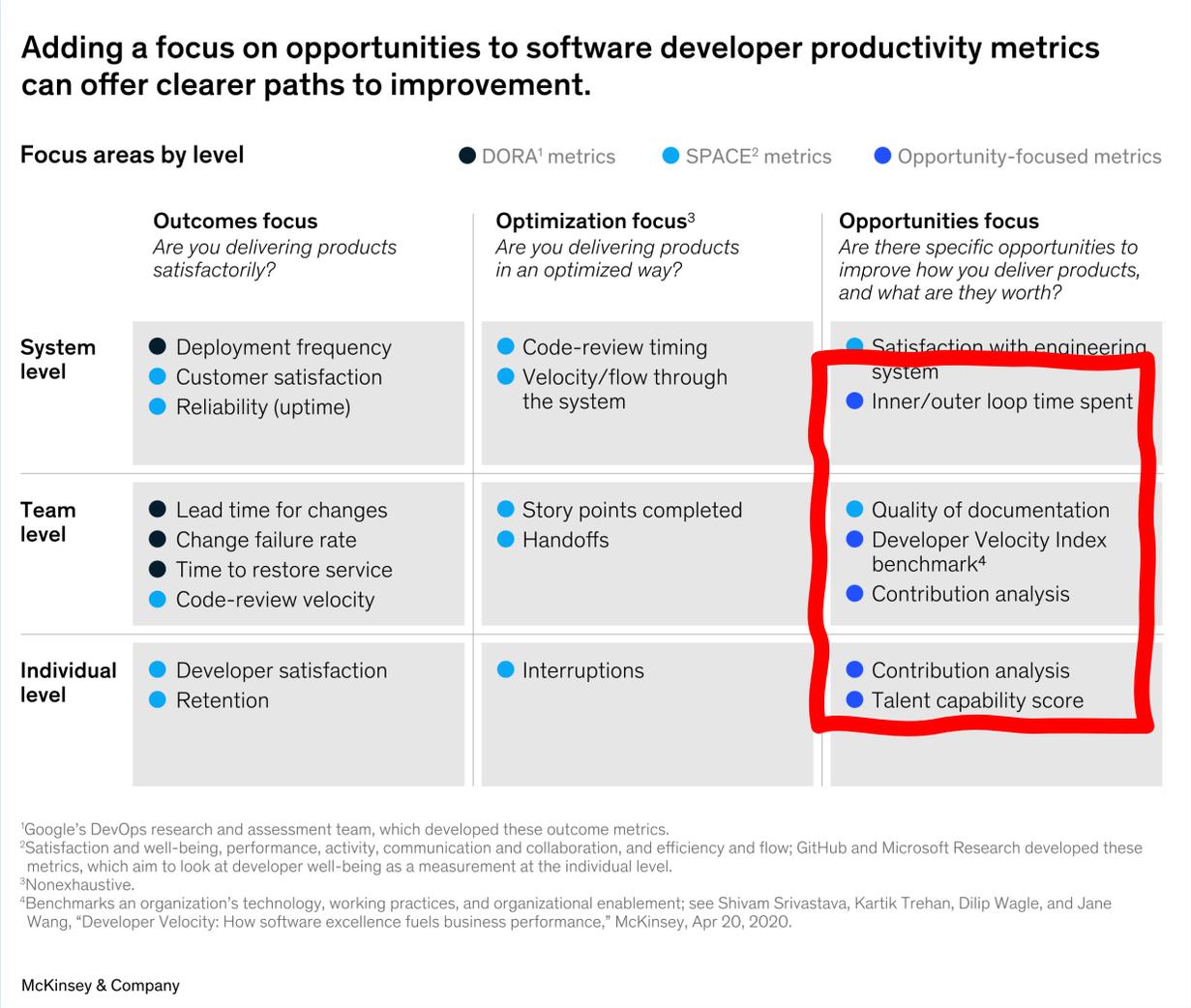
**My Theory**

**Management needs to change,  
and then people will change.**

# Part One: The Wonderful World of Management



# Management & workers often have different incentives & motivations



Sources: ["Great Attrition' or 'Great Attraction'? The choice is yours,"](#) Aaron De Smet, Bonnie Dowling, Marino Mugayar-Baldocchi, Bill Schaninger, McKinsey, Sep 2021; ["Yes, you can measure software developer productivity,"](#) Chandra Gnanasambandam, Martin Harrysson, Alharith Hussin, Jason Keovichit, and Shivam Srivastava, McKinsey, August, 2023. See also [further commentary from Côté.](#)

# A thriving organization focuses on satisfaction, not productivity

Causes of thriving	Because a developer is...
<b>Agency</b>	<ol style="list-style-type: none"> <li>able to voice disagreement with team definitions of success</li> <li>has a voice in how their contributions are measured</li> </ol>
<b>Motivation &amp; Self-Efficacy</b>	<ol style="list-style-type: none"> <li>motivated when working on code at work</li> <li>can see tangible progress most of the time</li> <li>is working on the type of code work they want to work on</li> <li>is confident that even when working in code is unexpectedly difficult, they will solve their problems</li> </ol>
<b>Learning Culture</b>	<ol style="list-style-type: none"> <li>learning new skills as a developer</li> <li>able to share the things they learn at work</li> </ol>
<b>Support &amp; Belonging</b>	<ol style="list-style-type: none"> <li>supported to grow, learn, and make mistakes by their team</li> <li>agrees they are accepted for who they are by their team</li> </ol>

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
<b>INDIVIDUAL</b> One person	<ul style="list-style-type: none"> <li>Developer satisfaction</li> <li>Retention<sup>†</sup></li> <li>Satisfaction with code reviews assigned</li> <li>Perception of code reviews</li> </ul>	<ul style="list-style-type: none"> <li>Code review velocity</li> </ul>	<ul style="list-style-type: none"> <li>Number of code reviews completed</li> <li>Coding time</li> <li># Commits</li> <li>Lines of code<sup>†</sup></li> </ul>	<ul style="list-style-type: none"> <li>Code review score (quality or thoughtfulness)</li> <li>PR merge times</li> <li>Quality of meetings<sup>†</sup></li> <li>Knowledge sharing, discoverability (quality of documentation)</li> </ul>	<ul style="list-style-type: none"> <li>Code review timing</li> <li>Productivity perception</li> <li>Lack of interruptions</li> </ul>
<b>TEAM OR GROUP</b> People that work together	<ul style="list-style-type: none"> <li>Developer satisfaction</li> <li>Retention<sup>†</sup></li> </ul>	<ul style="list-style-type: none"> <li>Code review velocity</li> <li>Story points shipped<sup>†</sup></li> </ul>	<ul style="list-style-type: none"> <li># Story points completed<sup>†</sup></li> </ul>	<ul style="list-style-type: none"> <li>PR merge times</li> <li>Quality of meetings<sup>†</sup></li> <li>Knowledge sharing or discoverability (quality of documentation)</li> </ul>	<ul style="list-style-type: none"> <li>Code review timing</li> <li>Handoffs</li> </ul>
<b>SYSTEM</b> End-to-end work through a system (like a development pipeline)	<ul style="list-style-type: none"> <li>Satisfaction with engineering system (e.g., CI/CD pipeline)</li> </ul>	<ul style="list-style-type: none"> <li>Code review velocity</li> <li>Code review (acceptance rate)</li> <li>Customer satisfaction</li> <li>Reliability (uptime)</li> </ul>	<ul style="list-style-type: none"> <li>Frequency of deployments</li> </ul>	<ul style="list-style-type: none"> <li>Knowledge sharing, discoverability (quality of documentation)</li> </ul>	<ul style="list-style-type: none"> <li>Code review timing</li> <li>Velocity/flow through the system</li> </ul>

<sup>†</sup> Use these metrics with (even more) caution – they can proxy more things.

# Management vs. workers often have different urgency & motivation to change

Exec's View	Work the Same	Transform!
Compensation	\$	\$\$\$\$
Risk	HIGH	HIGH
Outcome		

Staff's View	Work the Same	Transform!
Compensation	\$	
Risk		HIGH
Outcome		

# The people who do the work (should) decide how to change the work

**Leaders at the Genba**



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

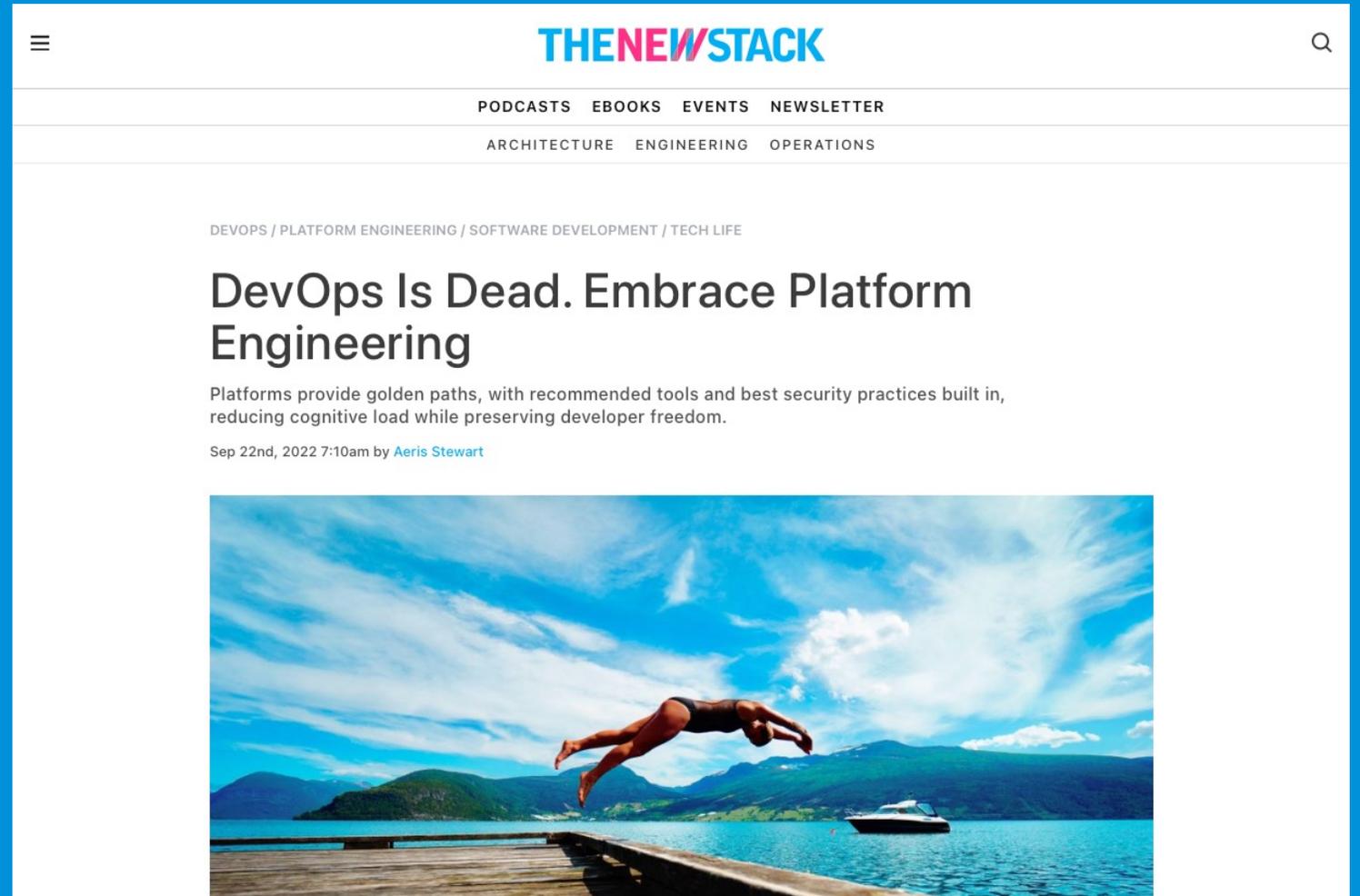
Copyright The Flow Consortium Inc and Nigel Thurlow 2017 onwards. All rights reserved. Confidential - Distribution, copying or commercial use prohibited without permission.

FLOW SYSTEM

5 DevOpsDays DFW 2022

403

# Part Two: Platforms



The screenshot shows the top portion of a web page. At the top left is a hamburger menu icon. The logo 'THE NEW STACK' is centered at the top in blue and red. To the right is a search icon. Below the logo is a navigation bar with links for 'PODCASTS', 'EBOOKS', 'EVENTS', and 'NEWSLETTER'. A second navigation bar contains 'ARCHITECTURE', 'ENGINEERING', and 'OPERATIONS'. The main content area features a breadcrumb trail: 'DEVOPS / PLATFORM ENGINEERING / SOFTWARE DEVELOPMENT / TECH LIFE'. The article title is 'DevOps Is Dead. Embrace Platform Engineering'. Below the title is a short paragraph: 'Platforms provide golden paths, with recommended tools and best security practices built in, reducing cognitive load while preserving developer freedom.' The author and date are listed as 'Sep 22nd, 2022 7:10am by Aeris Stewart'. At the bottom of the screenshot is a large image of a person performing a backflip over a wooden pier into a lake, with mountains in the background.

THE NEW STACK

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](#)



# Build it and they don't come



“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

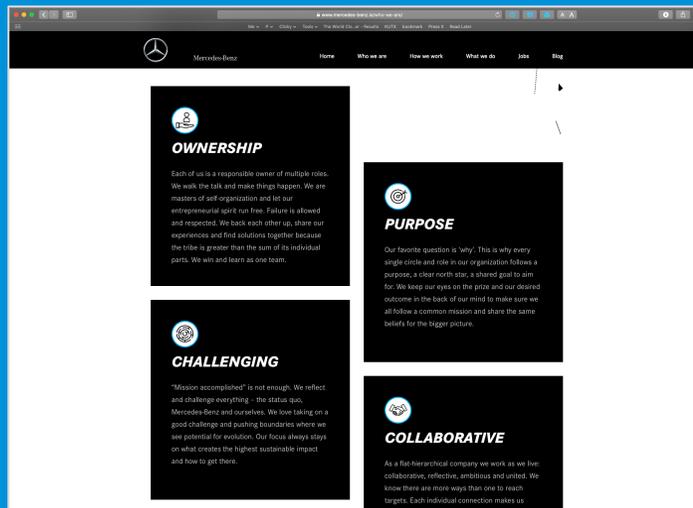
- What are we making?
- We have a strong vision for our product, and we're doing important work together every day to fulfill that vision.
- I have the context I need to confidently make changes while I'm working.
- I am proud of the work I have delivered so far for our product.
- I am learning things 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.

# Platform marketing, advocacy, consulting

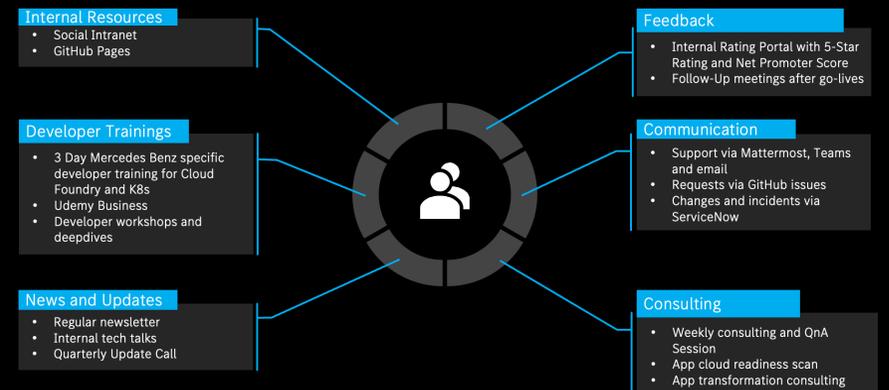
## Organizational Learning



## Focus on ways of working.....



## Developer centric platform



Mercedes-Benz

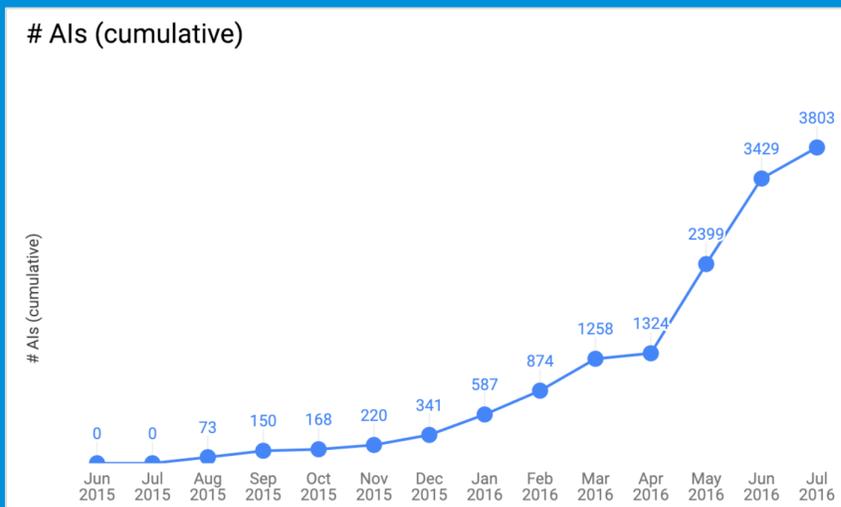
Improving Developer Productivity with Platform as a Product | Mercedes-Benz

17

# Scaling Phase – Pairing & Seeding to build trust & training



1. Create platform team.
2. Pick one or two apps, *real* apps.
3. Develop the apps & platform together.
4. Do this for three months.
5. Pick some more apps, to taste.
6. Seed app people to new teams.
7. GOTO 3.



# Lessons learned from supporting 1,500+ application at JP Morgan Chase

## A Successful Developer Experience (1/2)

1. Customer Focus: Treat internal developers like clients
2. Build, nourish and embrace a community around your platforms
3. Focus on end-to-end & deliver an integrated experience
4. Culture is critical
5. Cloud Blueprints
6. Cloud Parties
7. Self-service everything

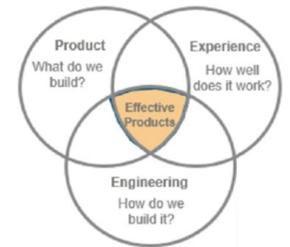


Build a customer-centre culture.  
"15 Proven Techniques to Improve Customer Experience (CX)"  
Blog by Snigdha Patel on the revechat.com platform

J P M o r g a n C h a s e

## A Successful Developer Experience (2/2)

8. Clear responsibility model, boundaries and platform contract
9. Operationally stable, reliable, and has well-defined SLOs
10. Inherently secure
11. Streamline tooling for CI/CD
12. Enable innovation through managing risk
13. Automate, automate, automate!
14. Short time to Hello World!
15. Partner for success

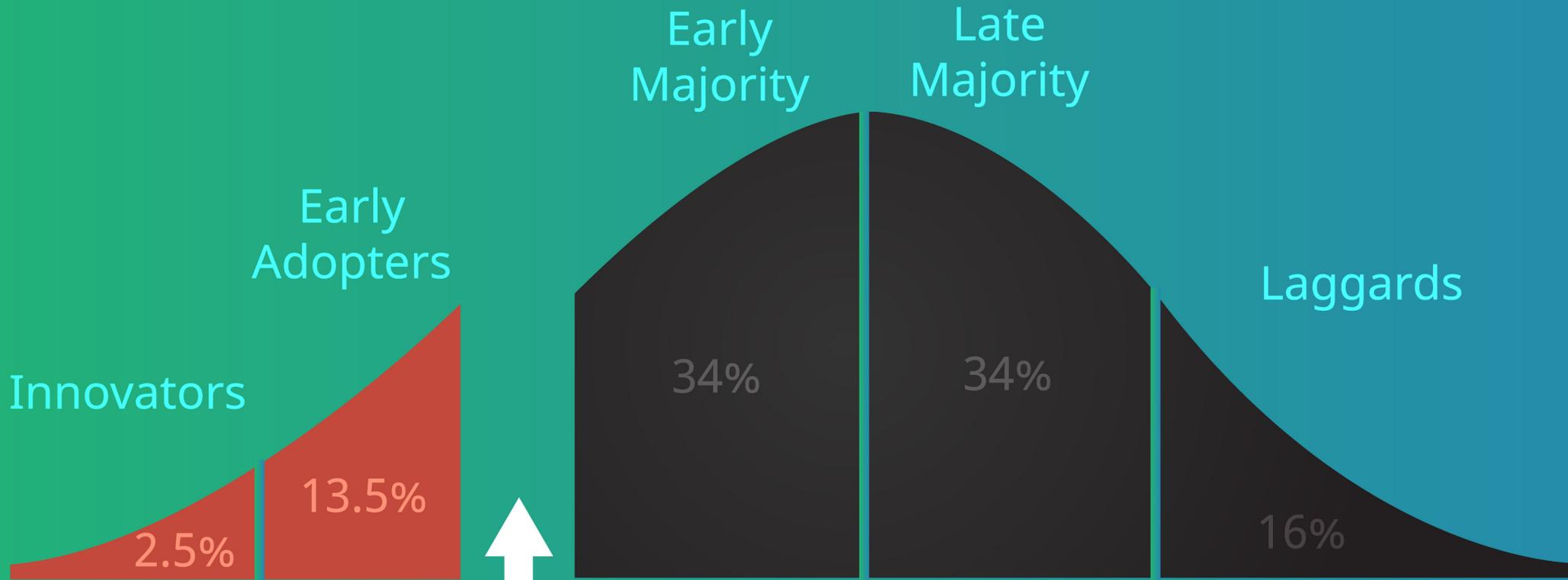


J P M o r g a n C h a s e

# Part Thre: A Plea



I'm getting too old for this shit.



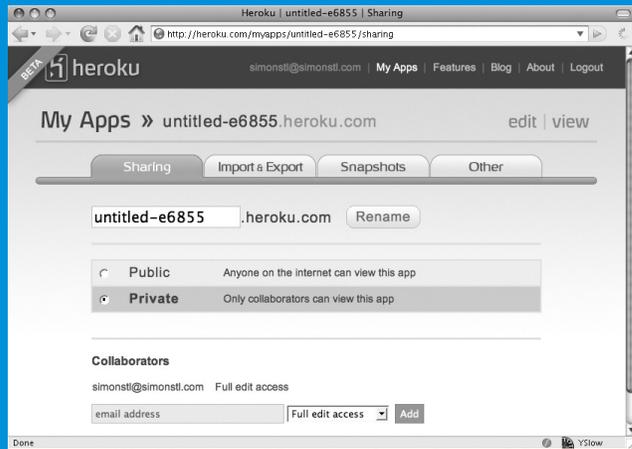
Chasm



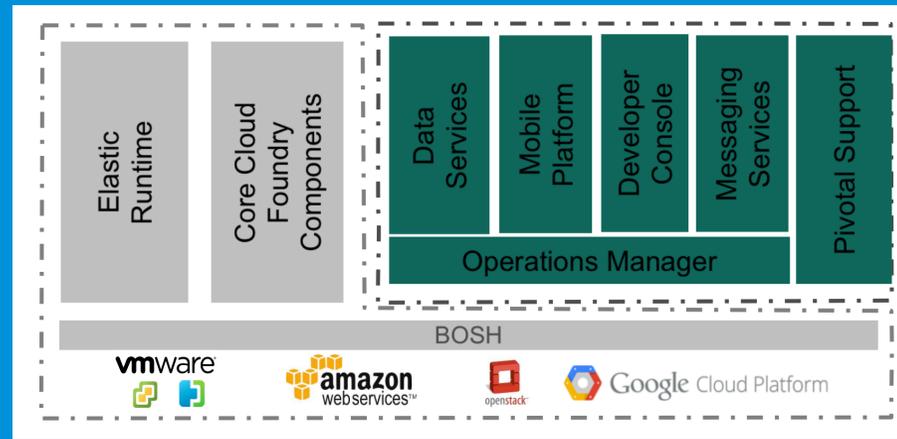
# The Eternal Recurrence of (Platforms, PaaS, DevOps, Cloud Native)

(٠□٠) 丩 □ ⊥ ⊥ )

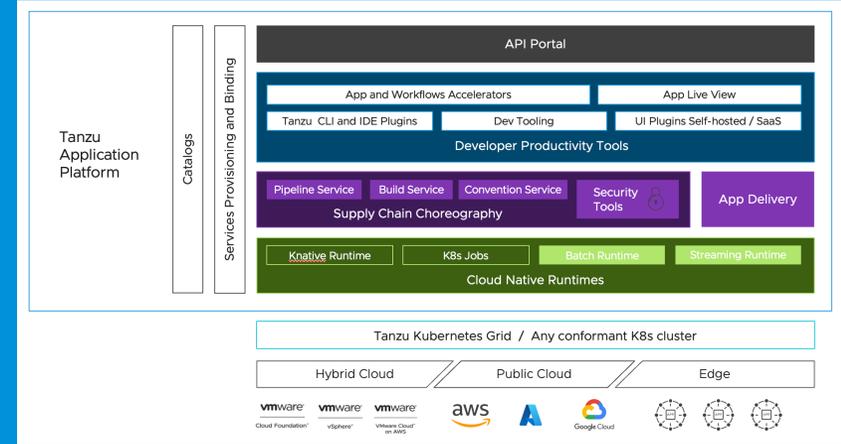
(٠□٠) 丩 □ ⊥ ⊥ )



2007



2015



2023 & Beyond



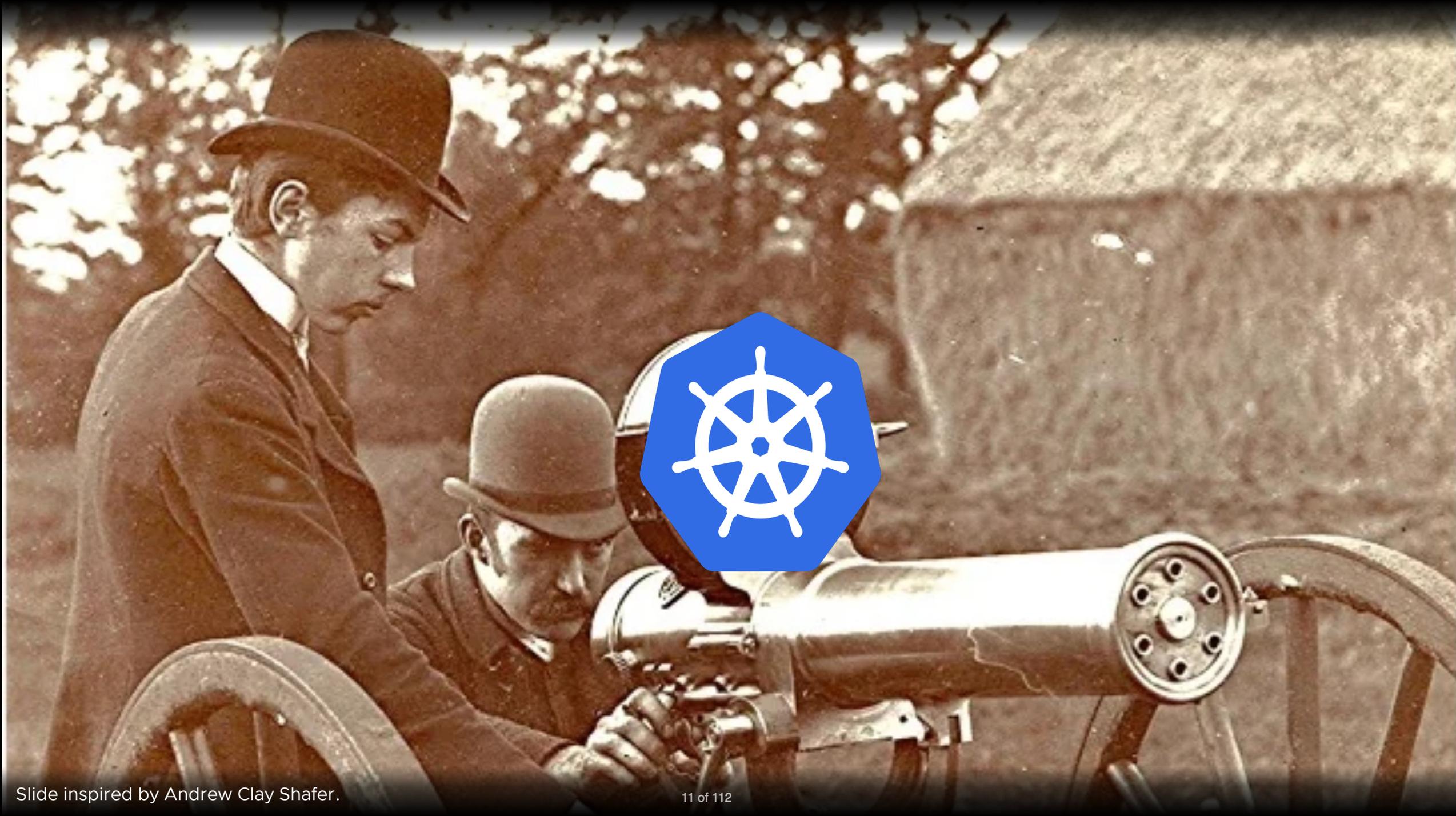
**Kelsey Hightower**   
@kelseyhightower

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

1:04 PM - 27 Nov 2017

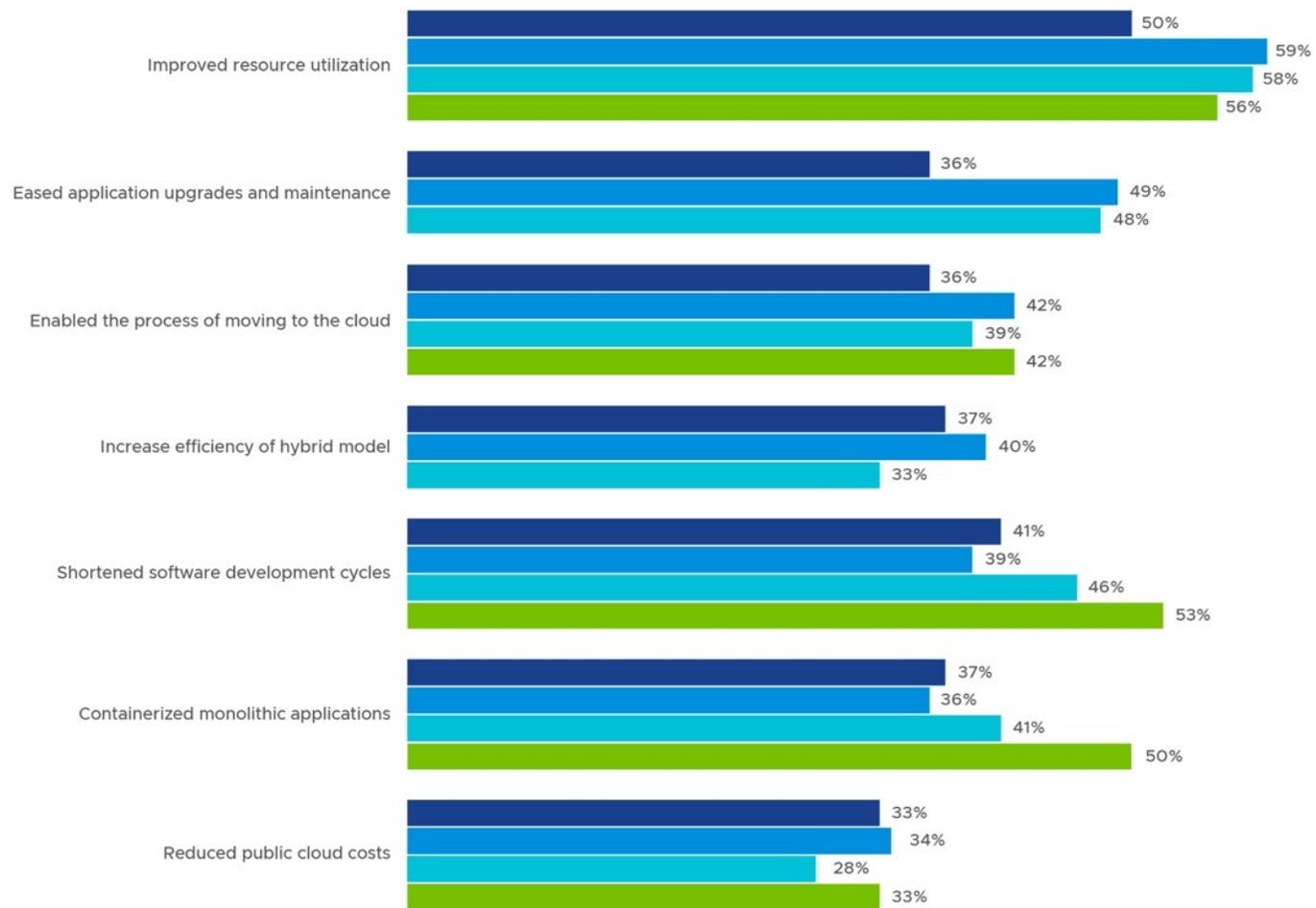
Not pictured:

OO, Small Talk, RUP, CORBA, J2EE/.Net, SOA & WS-\*, RAD, Low Code, Public Clouds



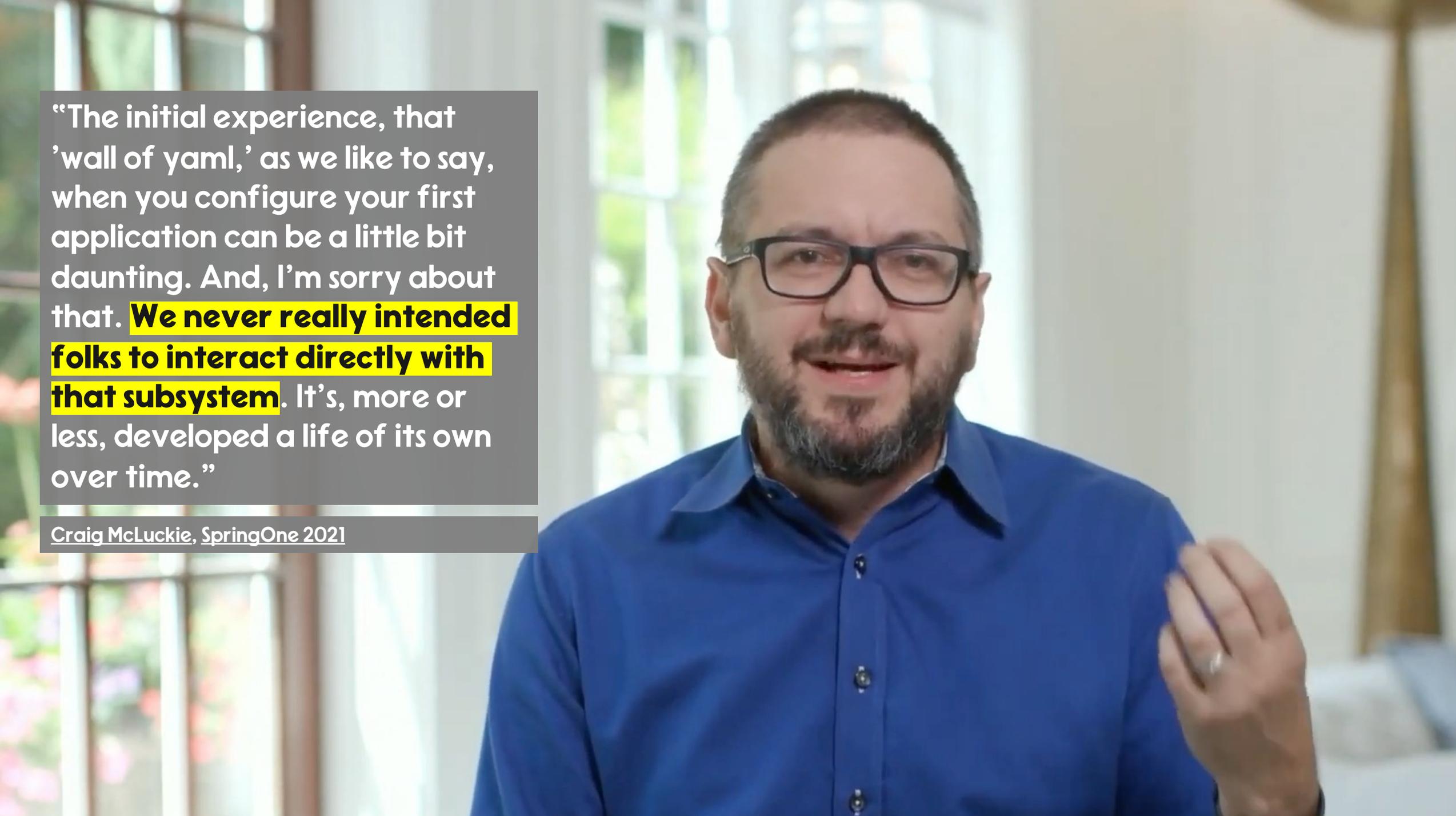
### What benefits has your organization realized from operating Kubernetes?

(Choose all that apply)



Source: State of Kubernetes 2022, UMuar

■ 2020 ■ 2021 ■ 2022 ■ 2023



“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.”

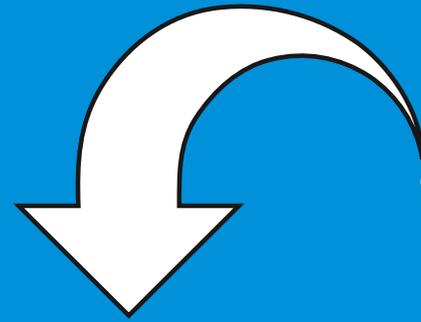
Craig McLuckie, SpringOne 2021

“We believe that we need to reimagine banking to make banking simple, seamless, as well as invisible to allow our customers to live a borderless life without the bank's complexity.”

*Siew Choo Soh, DBS Bank*



# Thanks!



*Slides & stuff*

 <https://newsletter.cote.io>

 <https://cote.io/platform/>

 [cote@broadcom.com](mailto:cote@broadcom.com)

