



Two talks:

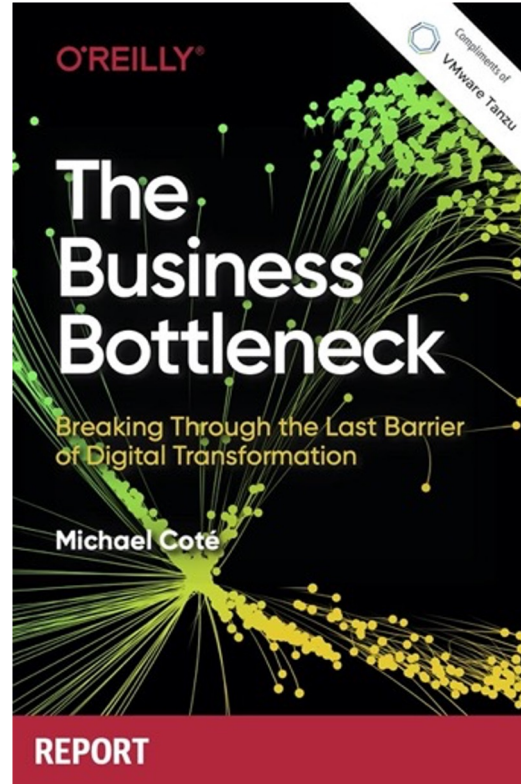
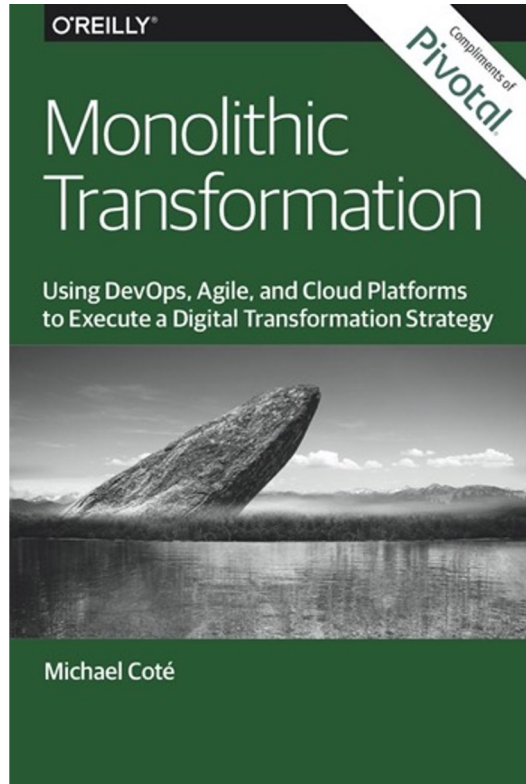
1. “State of DevOps”

2. “We Fear Change”



Coté

<https://newsletter.cote.io/> | cote@broadcom.com



Three views of DevOps in 2024

What DevOps means to people.

1.

DevOps is

A centralized developer tools group

Market Definition/Description

This Magic Quadrant for DevOps Platforms is the first version of this Magic Quadrant. It replaces the [Market Guide for Value Stream Delivery Platforms](#).

Gartner defines DevOps platforms as those that provide fully integrated capabilities to enable continuous delivery of software using Agile and DevOps practices. These capabilities run the gamut of the software development life cycle (SDLC) and include product planning, version control, continuous integration, test automation, continuous deployment, release orchestration, automating security and compliance policies, monitoring, and observability. DevOps platforms support team collaboration, secure software development and measurement of software delivery metrics.

DevOps platforms simplify creation, maintenance and management of the components

2.

DevOps 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 inquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

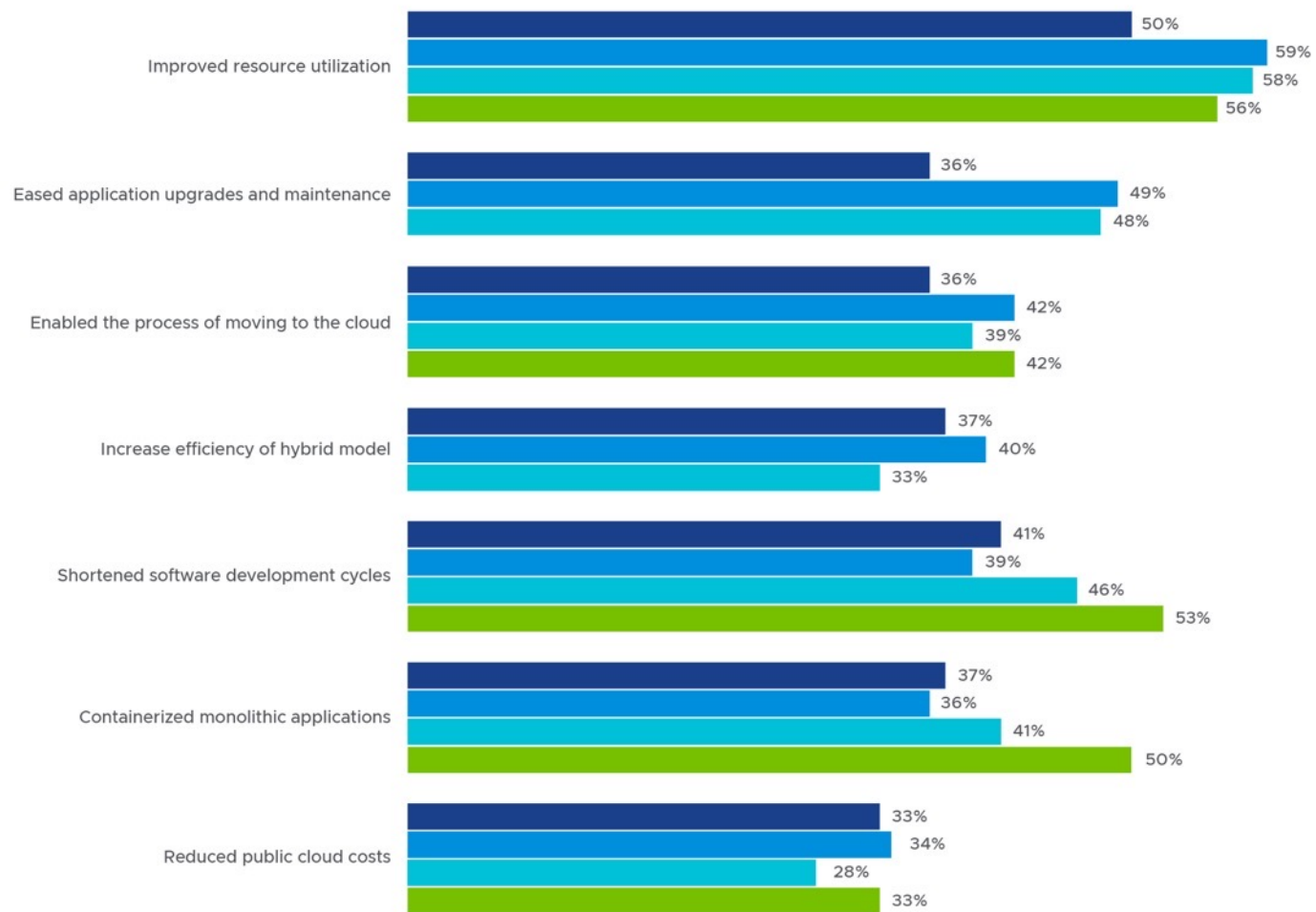
3.

DevOps is

Simplifying yaml for developers
(i.e., “Kubernetes”)

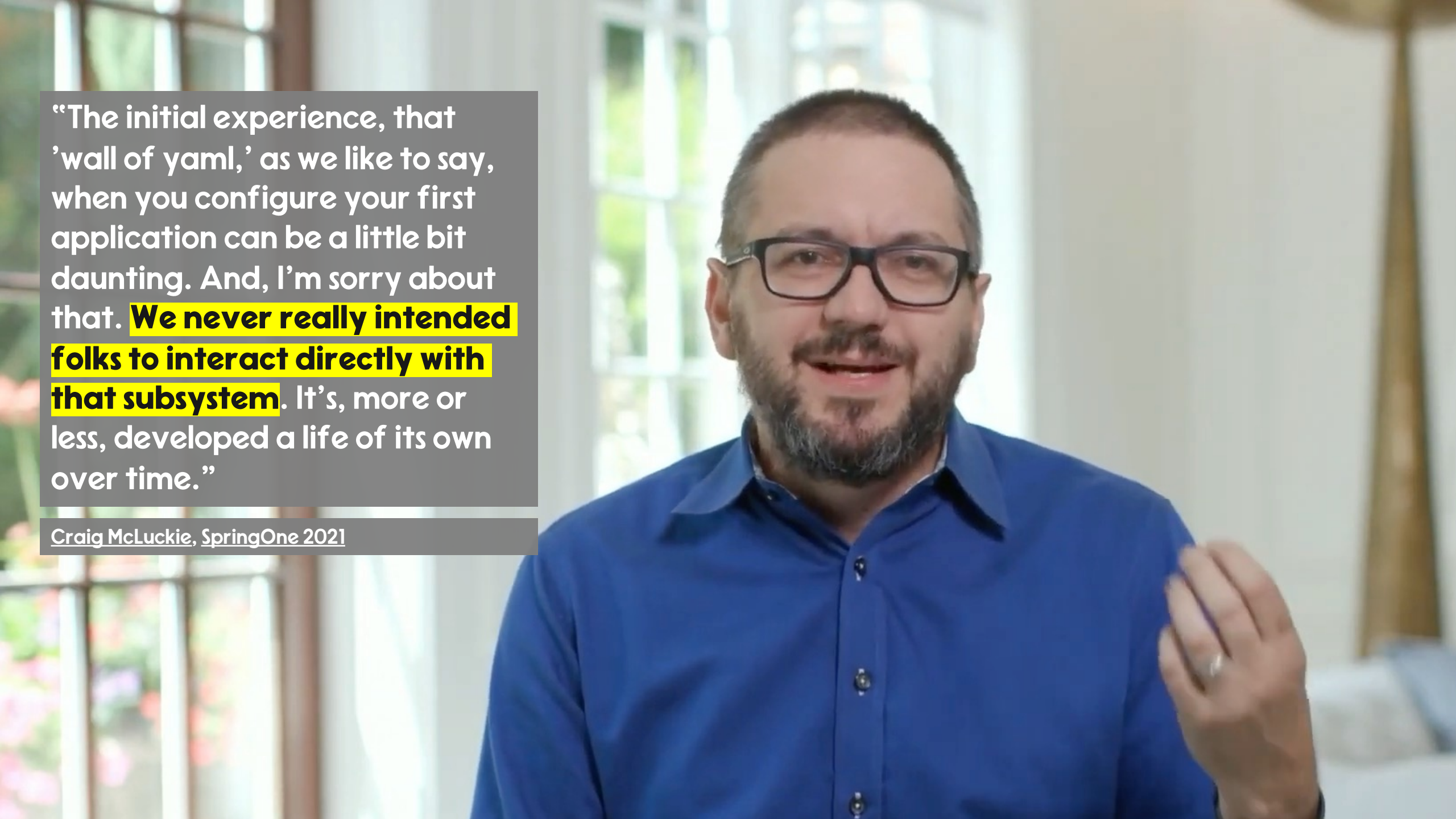
What benefits has your organization realized from operating Kubernetes?

(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

4.

DevOps is

Building & running platforms



DEVOPS / PLATFORM ENGINEERING / SOFTWARE DEVELOPMENT / TECH LIFE

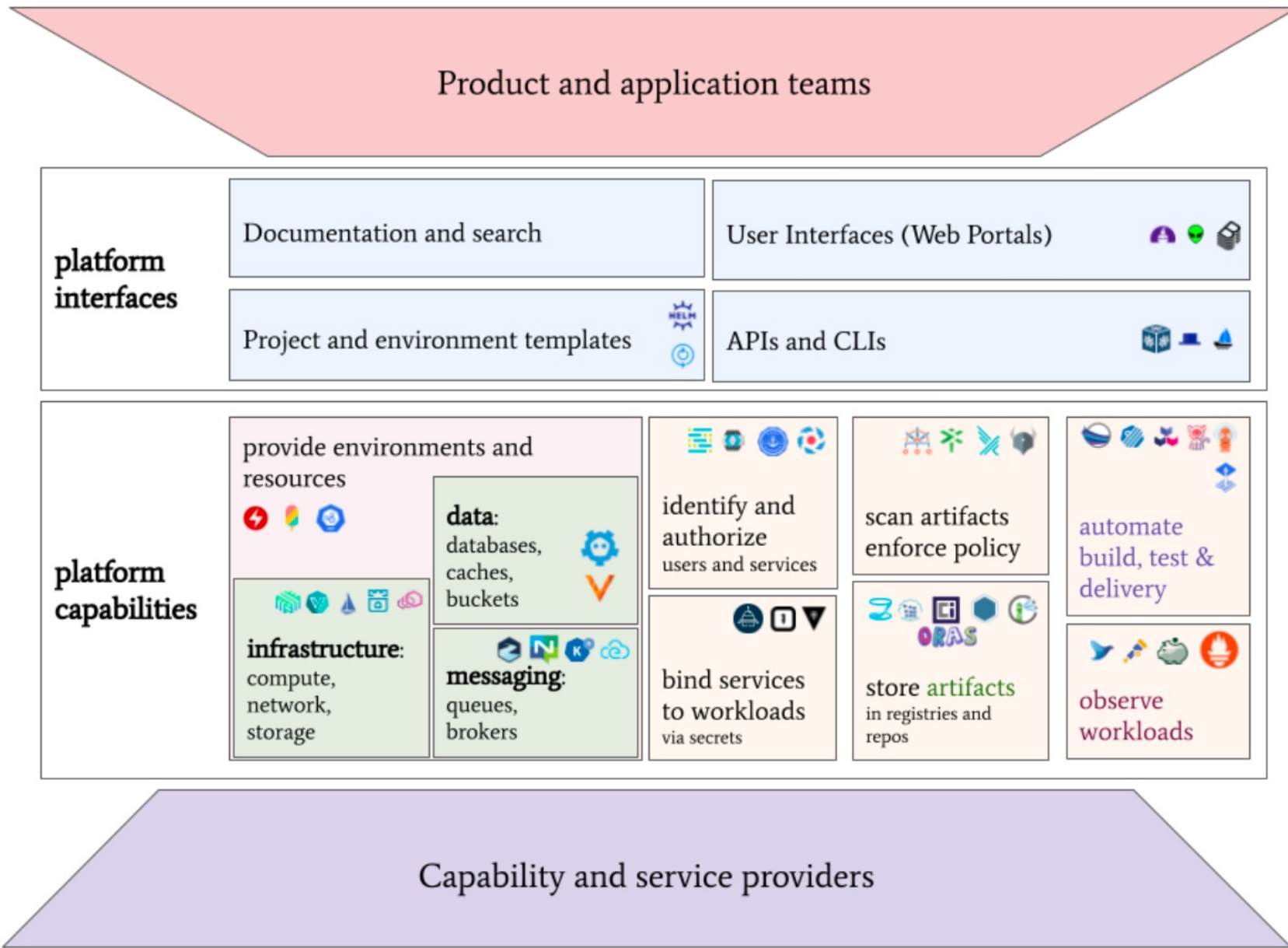
DevOps Is Dead. Embrace Platform Engineering

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.



Product and application teams

platform interfaces

Documentation and search

User Interfaces (Web Portals)



Project and environment templates



APIs and CLIs



platform capabilities

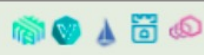
provide environments and resources



data:
databases,
caches,
buckets



infrastructure:
compute,
network,
storage



messaging:
queues,
brokers



identify and
authorize
users and services



bind services
to workloads
via secrets



scan artifacts
enforce policy



store artifacts
in registries and
repos



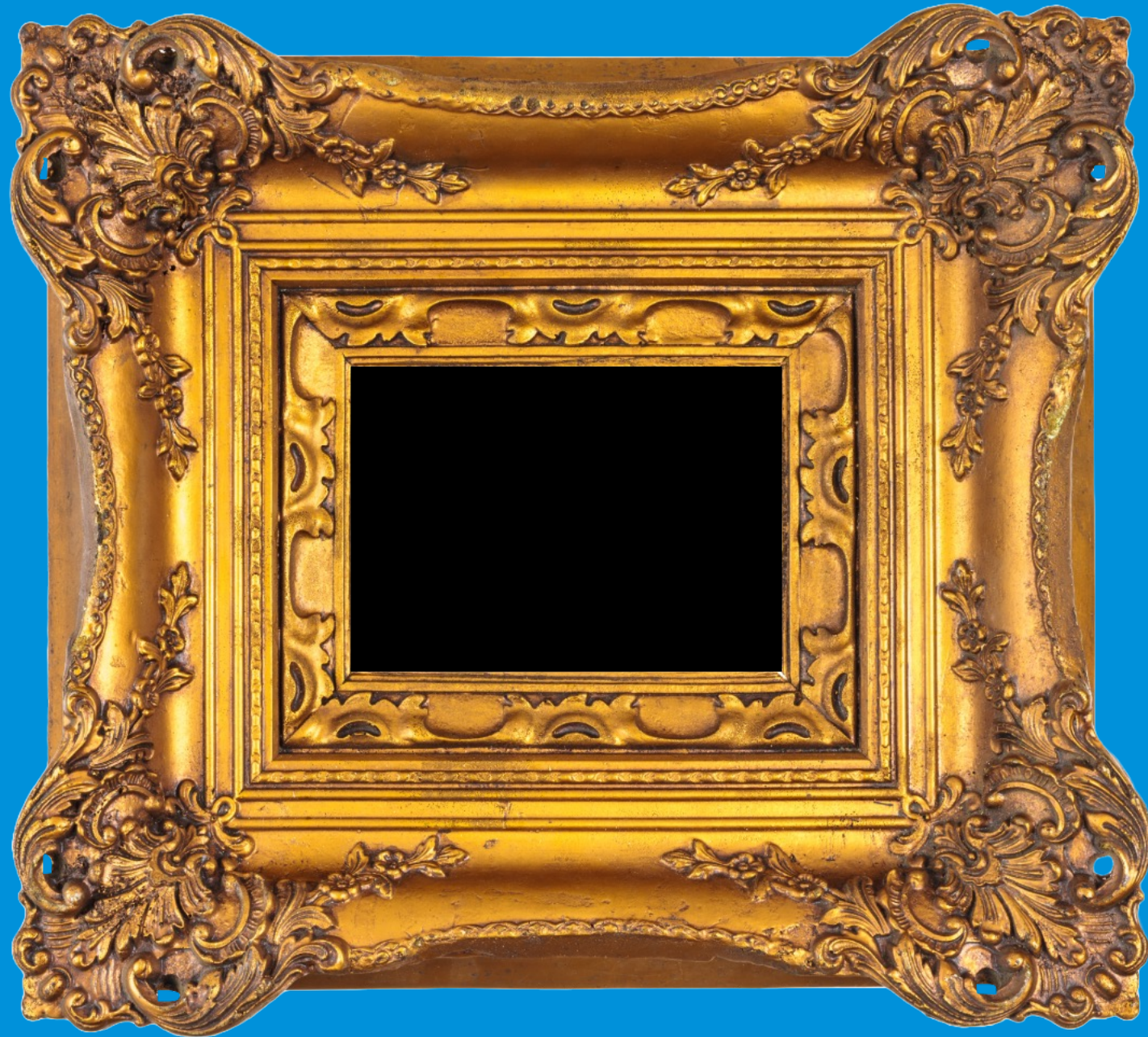
automate
build, test &
delivery



observe
workloads



Capability and service providers





We fear change.

Understanding why people resist using your platform

Coté - February 6thth, 2024

We all know that

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

Actually,

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

Evolution of the *Star Trek* Analogy

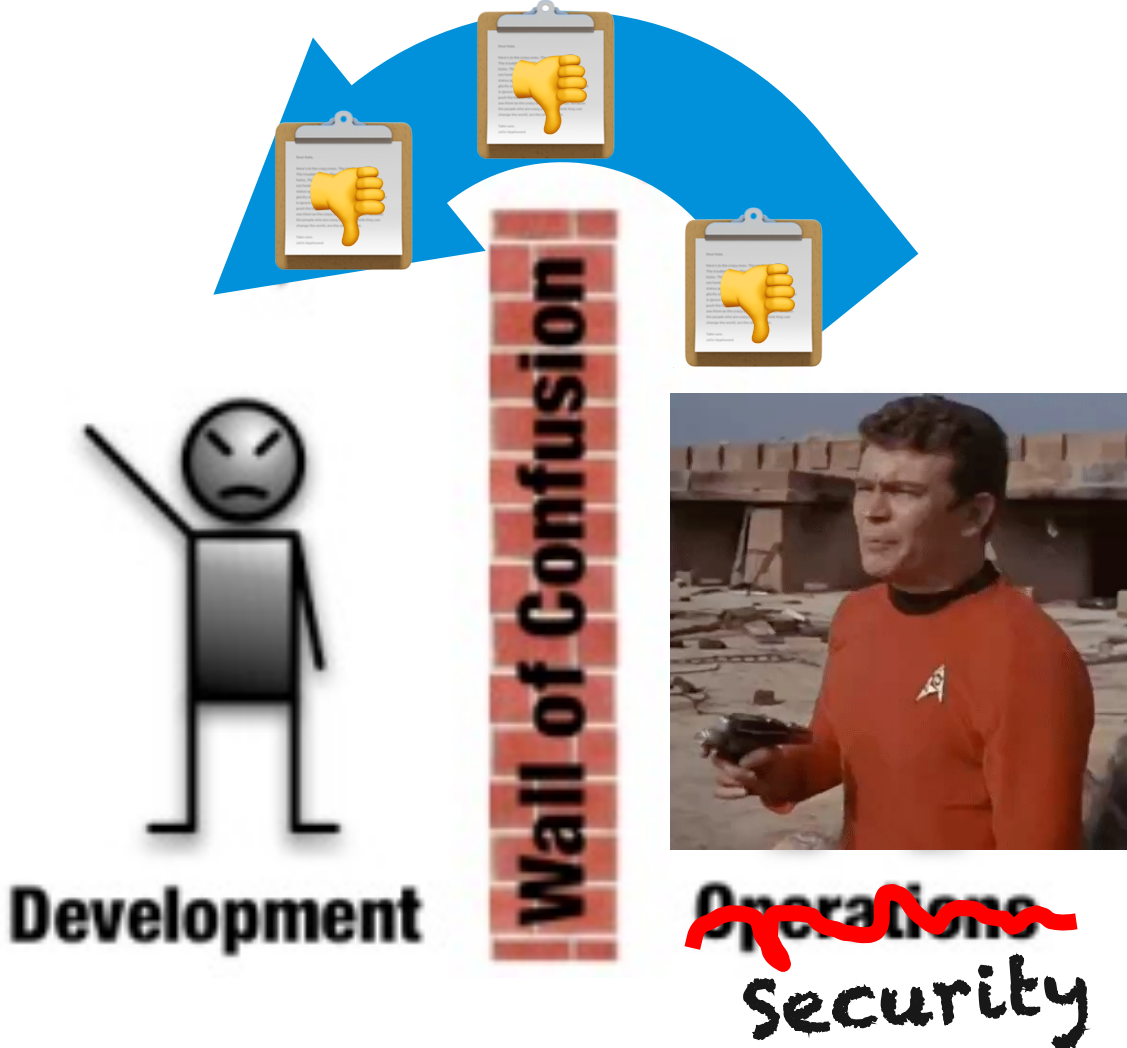


Little bit weird
Sits closer to the boss
Thinks too hard



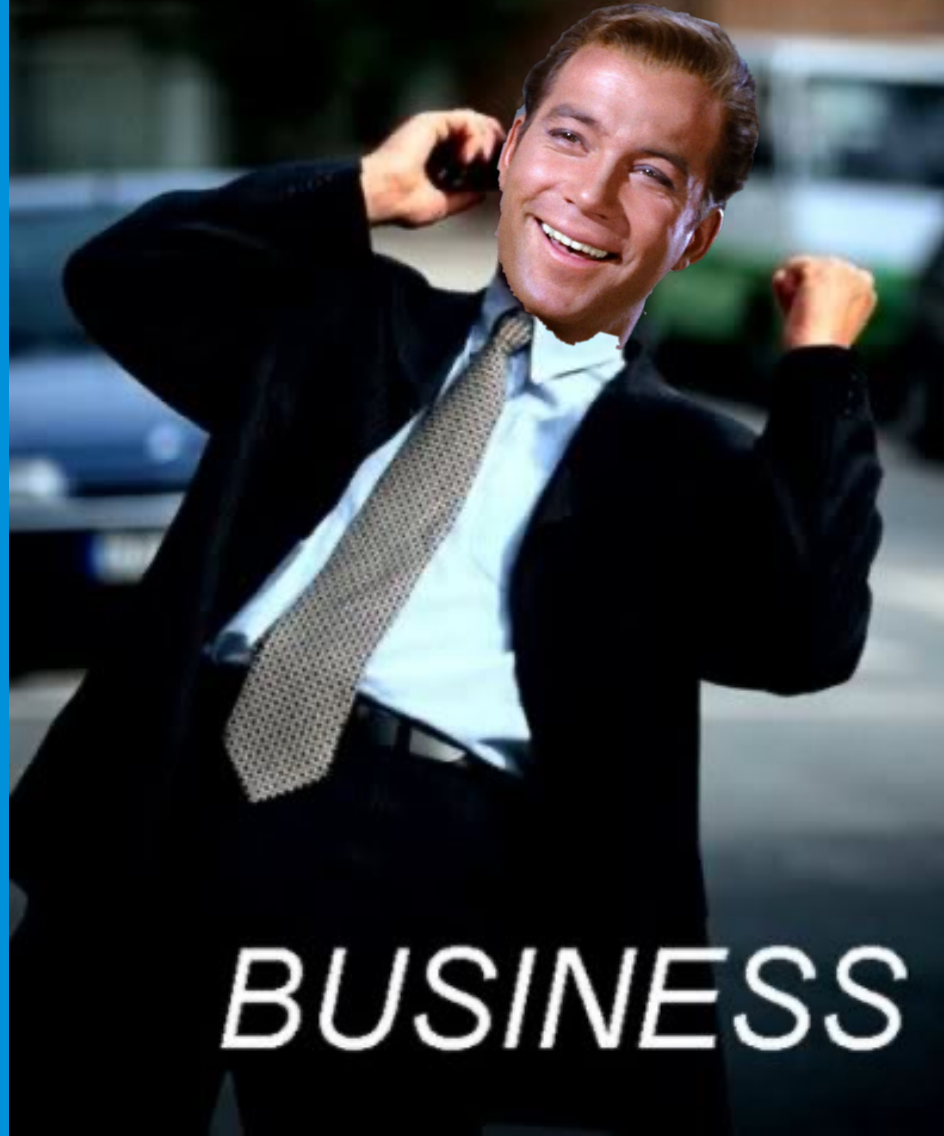
Pulls levers & turns knobs
Easily excited
Yells a lot in emergencies

DevSecOps



Source: Original concept/graphic from Andrew Shafer and Lee Thompson, circa 2009. *Star Trek*, circa 1968.

HA HA!



BUSINESS

The Wonderful World of Management



“This is a 1 1/2 CIO Job.”



Sources: “[Fortune 500 C-Suite Snapshot: Profiles in Functional Leadership](#),” SpencerStuart, 2023 (analysis done as of June 30, 2023).

Management & workers often have different incentives & motivations

Adding a focus on opportunities to software developer productivity metrics can offer clearer paths to improvement.

Focus areas by level

● DORA¹ metrics ● SPACE² metrics ● Opportunity-focused metrics

	Outcomes focus <i>Are you delivering products satisfactorily?</i>	Optimization focus³ <i>Are you delivering products in an optimized way?</i>	Opportunities focus <i>Are there specific opportunities to improve how you deliver products, and what are they worth?</i>
System level	<ul style="list-style-type: none"> ● Deployment frequency ● Customer satisfaction ● Reliability (uptime) 	<ul style="list-style-type: none"> ● Code-review timing ● Velocity/flow through the system 	<ul style="list-style-type: none"> ● Satisfaction with engineering system ● Inner/outer loop time spent
Team level	<ul style="list-style-type: none"> ● Lead time for changes ● Change failure rate ● Time to restore service ● Code-review velocity 	<ul style="list-style-type: none"> ● Story points completed ● Handoffs 	<ul style="list-style-type: none"> ● Quality of documentation ● Developer Velocity Index benchmark⁴ ● Contribution analysis
Individual level	<ul style="list-style-type: none"> ● Developer satisfaction ● Retention 	<ul style="list-style-type: none"> ● Interruptions 	<ul style="list-style-type: none"> ● Contribution analysis ● Talent capability score

¹Google's DevOps research and assessment team, which developed these outcome metrics.

²Satisfaction and well-being, performance, activity, communication and collaboration, and efficiency and flow; GitHub and Microsoft Research developed these metrics, which aim to look at developer well-being as a measurement at the individual level.

³Nonexhaustive.

⁴Benchmarks an organization's technology, working practices, and organizational enablement; see Shivam Srivastava, Kartik Trehan, Dilip Wagle, and Jane Wang, "Developer Velocity: How software excellence fuels business performance," McKinsey, Apr 20, 2020.

McKinsey & Company



A thriving organization focuses on satisfaction, flow, ease, happiness





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

TABLE 1: **EXAMPLE DEVEX METRICS**

	FEEDBACK LOOPS	COGNITIVE LOAD	FLOW STATE
PERCEPTIONS <i>Human attitudes and opinions</i>	<ul style="list-style-type: none"> Satisfaction with automated test speed and output Satisfaction with time it takes to validate a local change Satisfaction with time it takes to deploy a change to production 	<ul style="list-style-type: none"> Perceived complexity of codebase Ease of debugging production systems Ease of understanding documentation 	<ul style="list-style-type: none"> Perceived ability to focus and avoid interruptions Satisfaction with clarity of task or project goals Perceived disruptiveness of being on-call
WORKFLOWS <i>System and process behaviors</i>	<ul style="list-style-type: none"> Time it takes to generate CI results Code review turnaround time Deployment lead time (time it takes to get a change released to production) 	<ul style="list-style-type: none"> Time it takes to get answers to technical questions Manual steps required to deploy a change Frequency of documentation improvements 	<ul style="list-style-type: none"> Number of blocks of time without meetings or interruptions Frequency of unplanned tasks or requests Frequency of incidents requiring team attention
KPIS <i>North star metrics</i>	<ul style="list-style-type: none"> Overall perceived ease of delivering software Employee engagement or satisfaction Perceived productivity 		


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

“We believe that we need to increase our independence to make our business more resilient to global economic uncertainty and to follow our strategic vision in a more complex world.”

Siew Choo Soh, DBS Bank



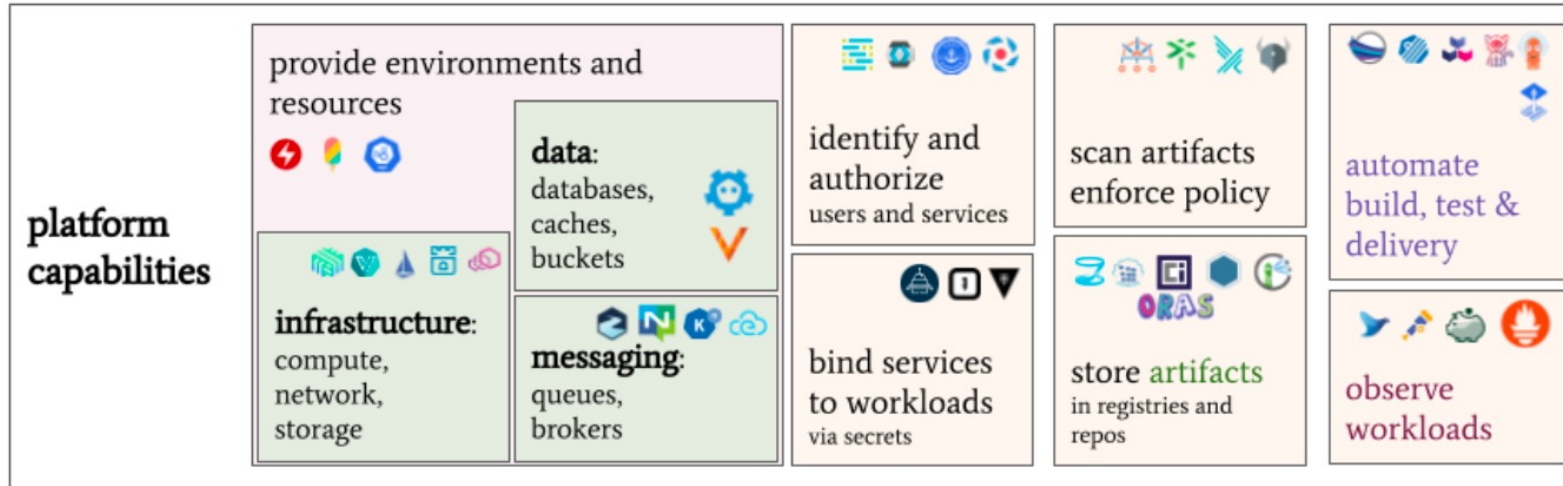
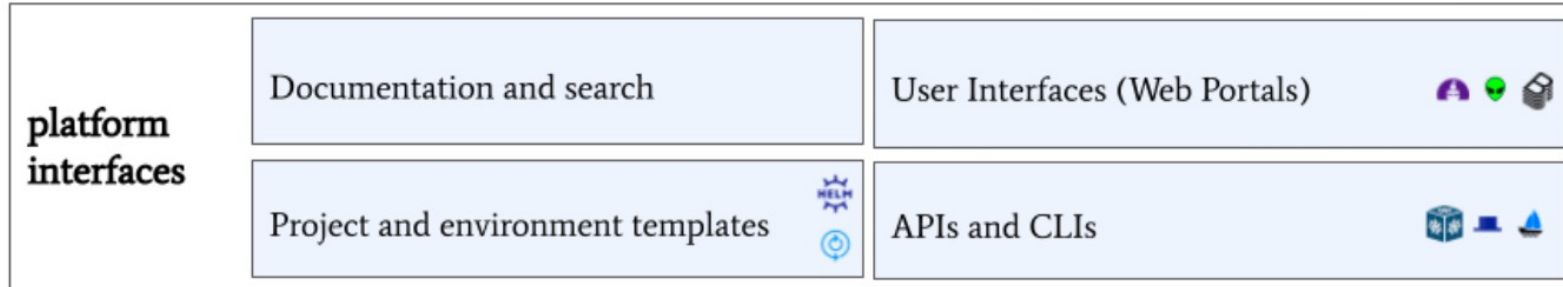
Containers *will not fix* your broken culture (and other hard truths)

BRIDGET KROMHOUT

**COMPLEX
SOCIO-TECHNICAL
SYSTEMS
ARE HARD;
FILM AT 11**

We focus so often on technical anti-patterns, neglecting similar problems inside our social structures. Spoiler alert: the solutions to many difficulties that seem technical can be found by examining our interactions with others. Let's talk about five things you'll want to know when working with those pesky creatures known as humans.

Product and application teams



Capability and service providers

“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.

TECHNICAL IMPROVEMENTS

Daily deploys

+30% developer productivity

+78% operational efficiency

60% reduction in incidents

Repaving prod months->weeks->daily

BUSINESS IMPROVEMENTS

65% shift to in-app ordering

+46% enrollment rates

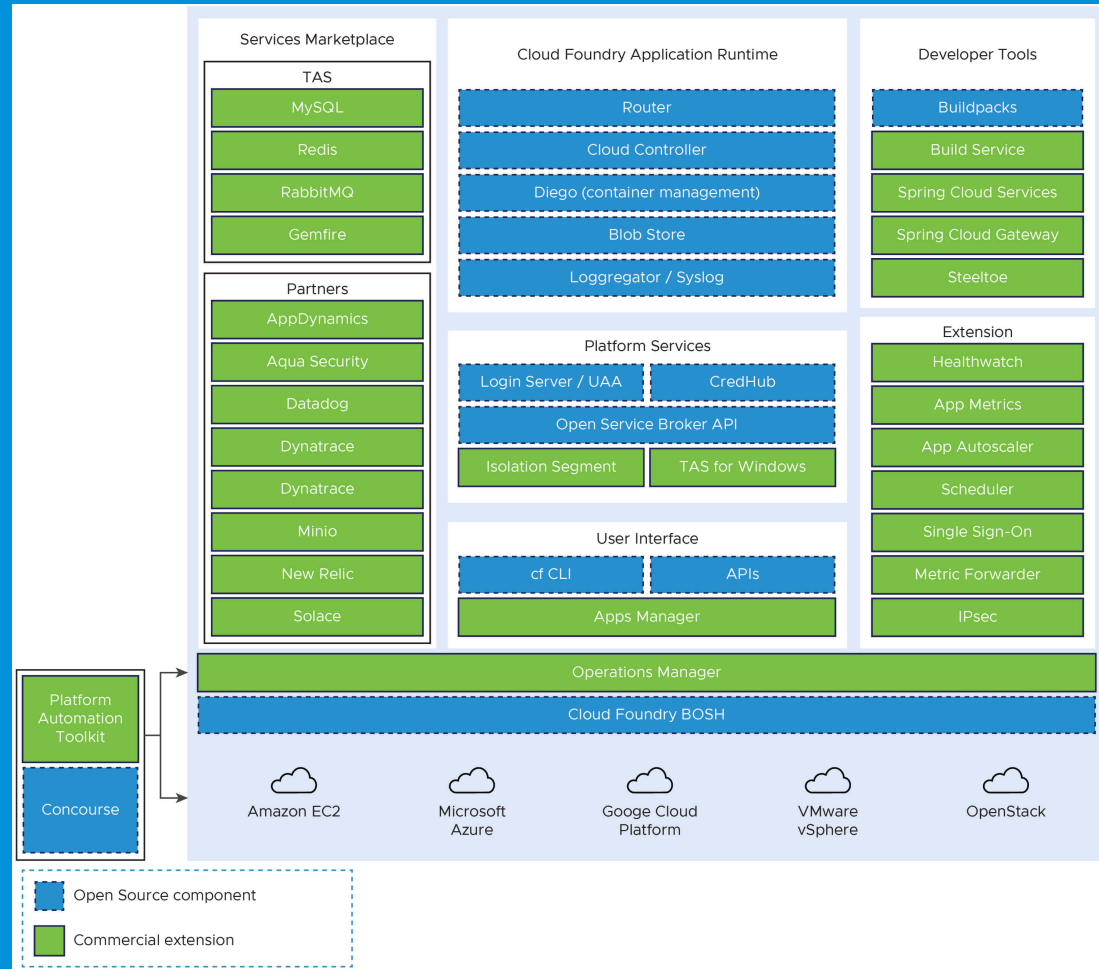
3 1/2 weeks to retool loan program

6 months to launch a new business

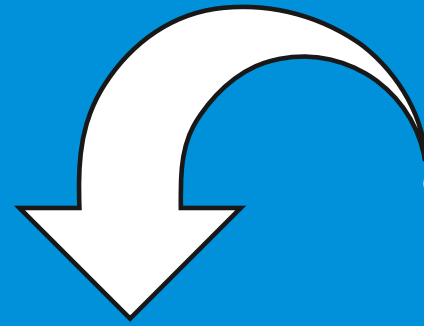
142% ROI on platform investment

A fine selection of pre-shaved yaks

<https://tanzu.vmware.com>



Thanks!



Slides & stuff



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



<https://cote.io/platform>



cote@broadcom.com

