Developer Productivity is Waste

Or, what to while DevOps is dead.

Coté – June 20th, 2024

Developer Productivity: The quick answers

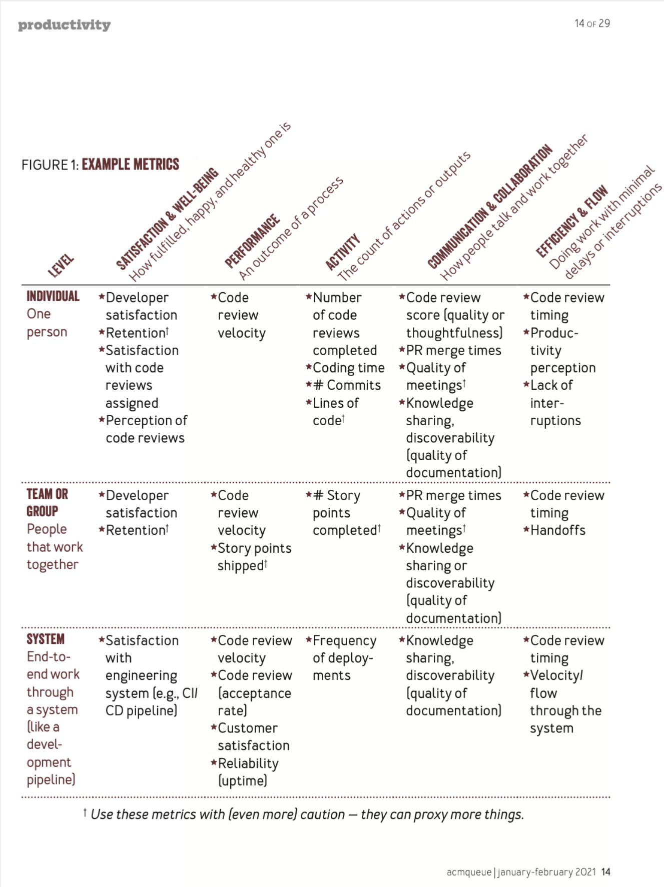
“It depends…”

Source: “DevEx: What Actually Drives Productivity,” Abi Noda, Margaret-Anne Storey, Nicole Forsgren, Microsoft Research, Michaela Greiler, May 2023. “Programming productivity,” Wikipedia, accessed June 12th, 2024.

"outcomes"
is a fancy word for:

€ $ £

Use metrics to find & monitor happiness, flow, features



productivity 14 of 29

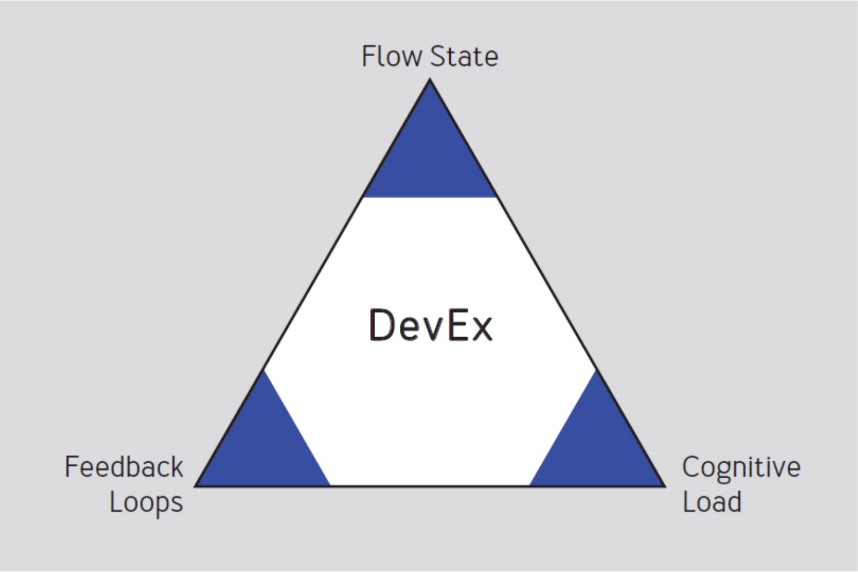
FIGURE 1: **EXAMPLE METRICS**

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.*

acmqueue | january-february 2021 14

FIGURE 1: **THREE CORE DIMENSIONS OF DEVELOPER EXPERIENCE**



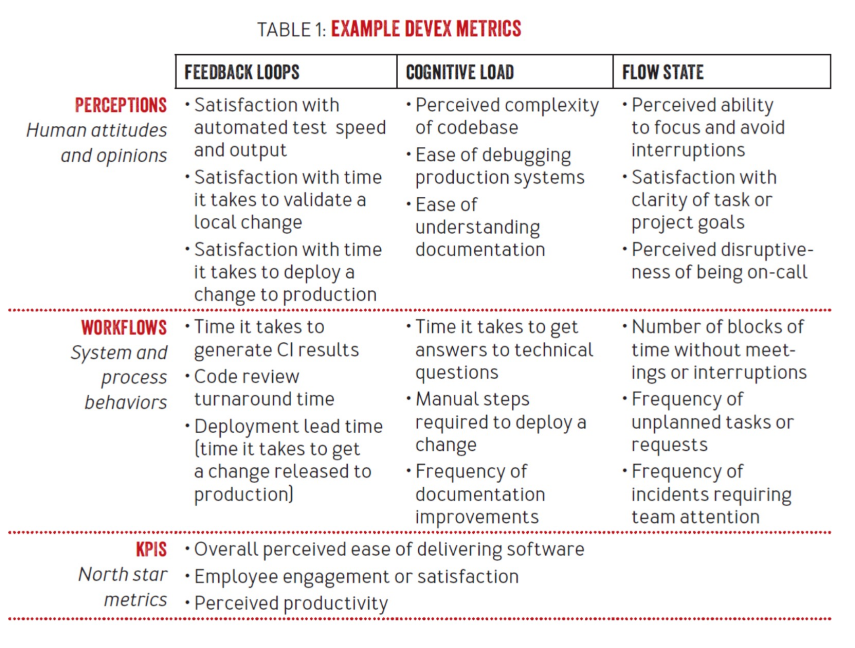
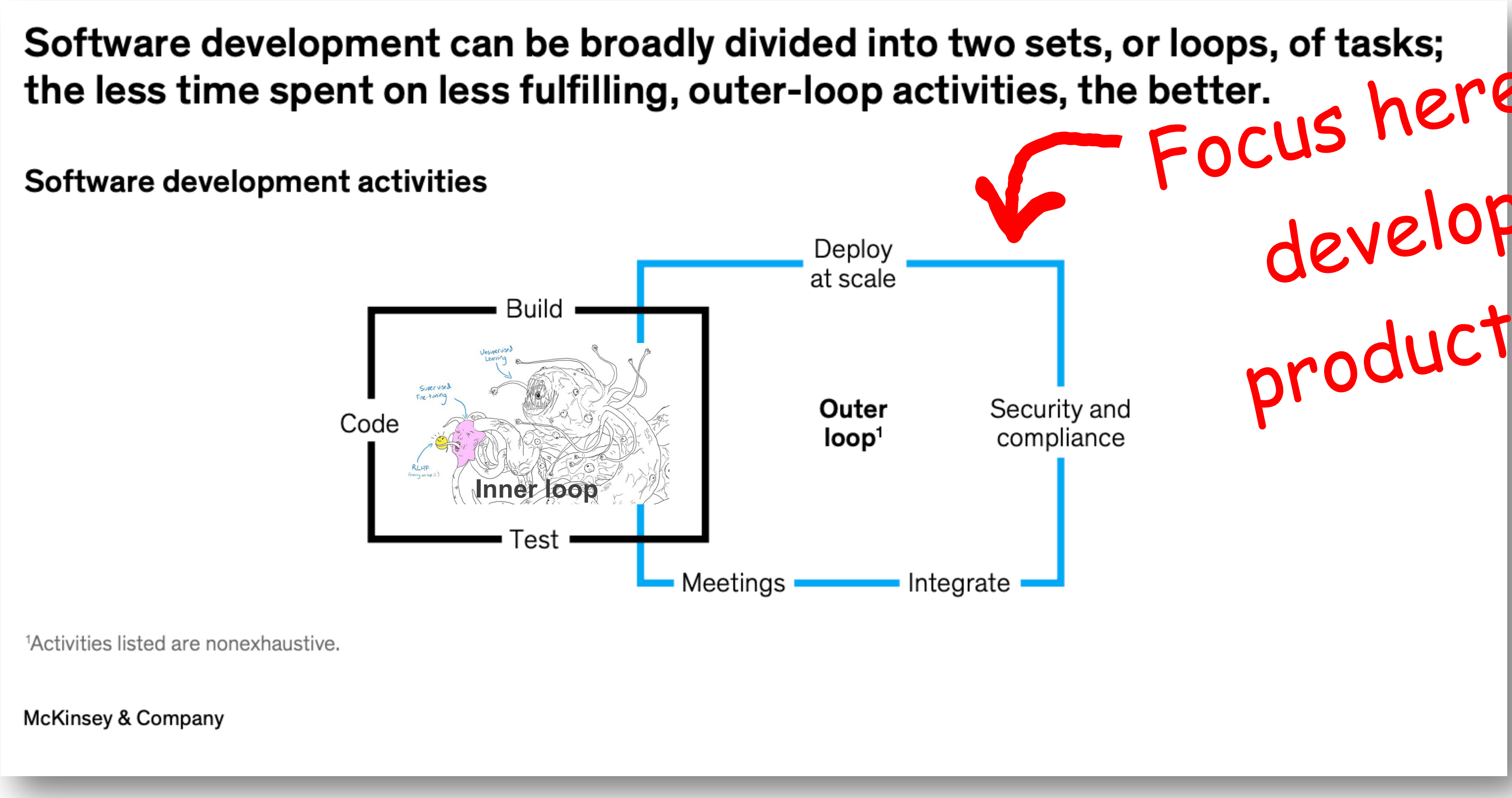


TABLE 1: **EXAMPLE DEVEX METRICS**

	FEEDBACK LOOPS	COGNITIVE LOAD	FLOW STATE
PERCEPTIONS *Human attitudes and opinions*	• 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	• Perceived complexity of codebase • Ease of debugging production systems • Ease of understanding documentation	• Perceived ability to focus and avoid interruptions • Satisfaction with clarity of task or project goals • Perceived disruptiveness of being on-call
WORKFLOWS *System and process behaviors*	• Time it takes to generate CI results • Code review turnaround time • Deployment lead time (time it takes to get a change released to production)	• Time it takes to get answers to technical questions • Manual steps required to deploy a change • Frequency of documentation improvements	• Number of blocks of time without meetings or interruptions • Frequency of unplanned tasks or requests • Frequency of incidents requiring team attention
KPIS *North star metrics*	• Overall perceived ease of delivering software • Employee engagement or satisfaction • Perceived productivity		

Source: "The SPACE of Developer Productivity," Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler, Jan-Feb, 2021; "DevEx: What Actually Drives Productivity," Abi Noda, Margaret-Anne Storey, Nicole Forsgren, Microsoft Research, Michaela Greiler, May 2023.



Software development can be broadly divided into two sets, or loops, of tasks; the less time spent on less fulfilling, outer-loop activities, the better.

Software development activities

Build

Code

Inner loop

Test

Deploy at scale

Outer loop[1]

Security and compliance

Meetings

Integrate

[1] Activities listed are nonexhaustive.

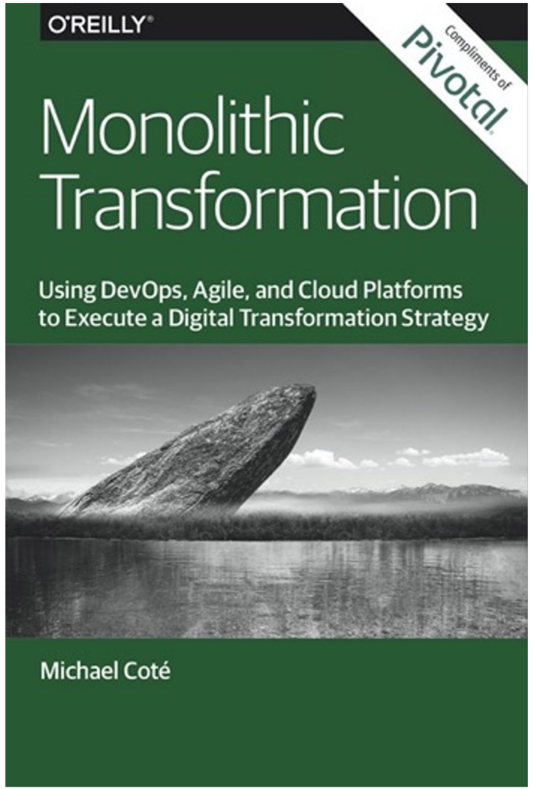
McKinsey & Company

Focus here for developer productivity

Source: "Yes, you can measure software developer productivity," Chandra Gnanasambandam, Martin Harrysson, Alharith Hussin, Jason Keovichit, Shivam Srivastava, McKinsey Aug(?) 2023.

Coté

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



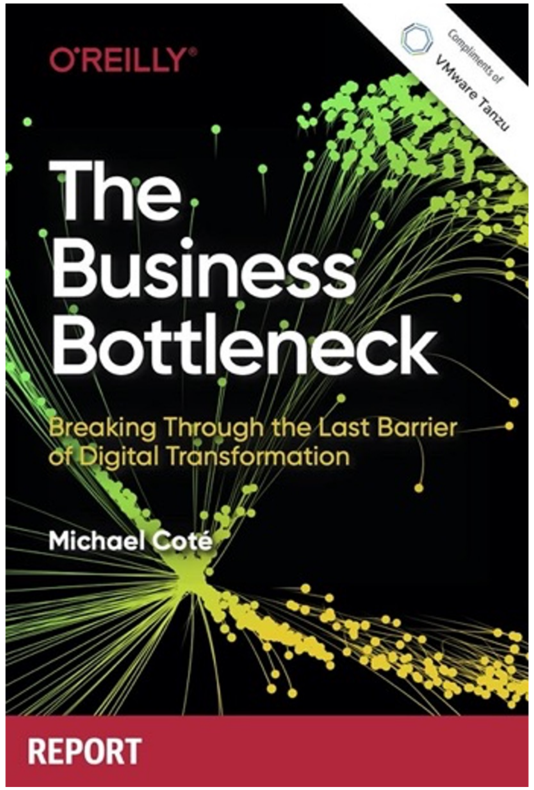
O'REILLY

Compliments of Pivotal

Monolithic Transformation

Using DevOps, Agile, and Cloud Platforms to Execute a Digital Transformation Strategy

Michael Coté



O'REILLY

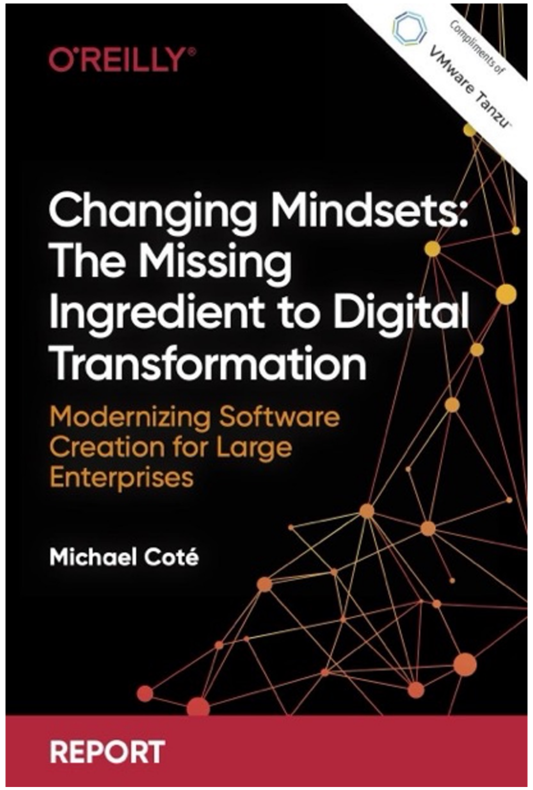
Compliments of VMware Tanzu

The Business Bottleneck

Breaking Through the Last Barrier of Digital Transformation

Michael Coté

REPORT



O'REILLY

Compliments of VMware Tanzu

Changing Mindsets: The Missing Ingredient to Digital Transformation

Modernizing Software Creation for Large Enterprises

Michael Coté

REPORT

Tanzu Talk

SOFTWARE DEFINED TALK

RedMonk DELL 451 Research Pivotal vmware Tanzu by Broadcom

Start with: Who's asking?

This guy.

Sources: Know Your Meme.

Competition👍

“Developer Velocity”

or,

Ship (more) features, more frequently.

Growth👍

Adding more developers

vs.

Increasing productivity per developer

Source: *Software Engineering at Google*, chapter 11, various, March 202

Costs👎

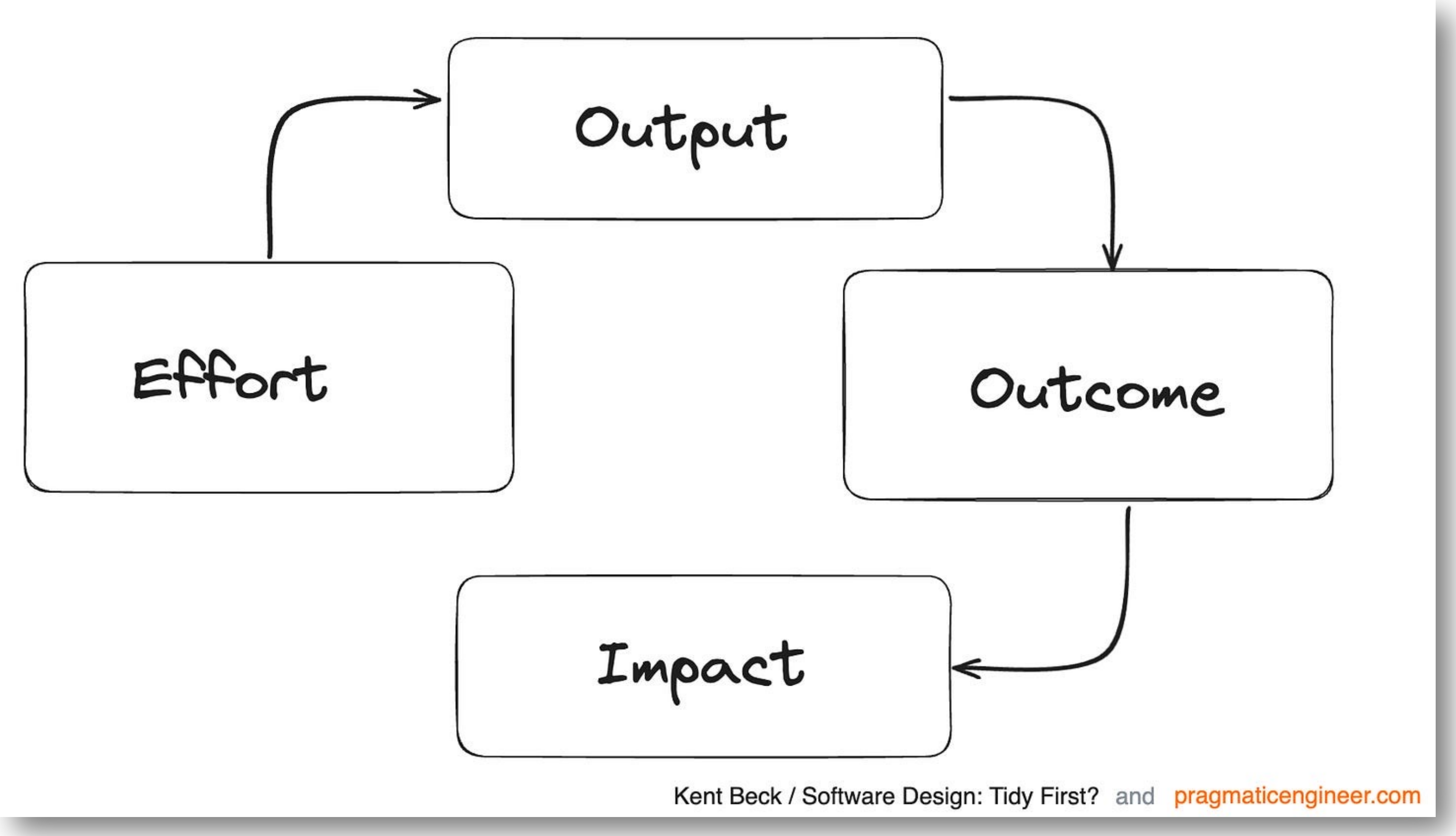
1. Are we paying too much?
2. Could we get by with paying less?
3. Who should I punish/fire?
4. (Who should I give more money to?)

(Also, sure,
craft optimization
& personal improvement)

What is “developer productivity”?

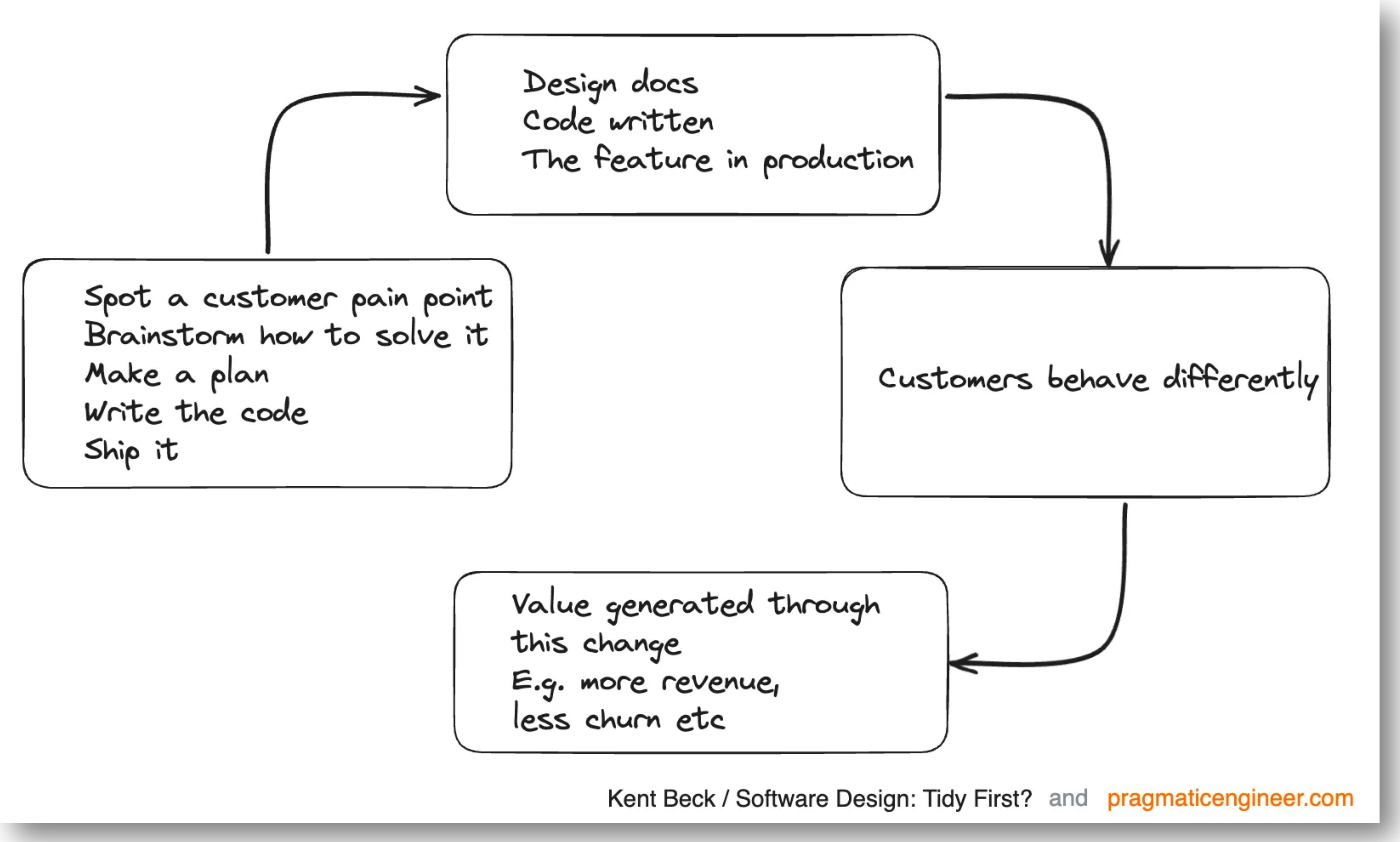
“It depends…” 🙄

Source: “DevEx: What Actually Drives Productivity,” Abi Noda, Margaret-Anne Storey, Nicole Forsgren, Microsoft Research, Michaela Greiler, May 2023; “Programming productivity,” Wikipedia, accessed June 12th, 2024.



Kent Beck / Software Design: Tidy First? and pragmaticengineer.com

Source: “Measuring developer productivity? A response to McKinsey,” Kent Beck and Gergely Orosz, August 2023.



Design docs
Code written
The feature in production

Customers behave differently

Value generated through this change
E.g. more revenue, less churn etc

Spot a customer pain point
Brainstorm how to solve it
Make a plan
Write the code
Ship it

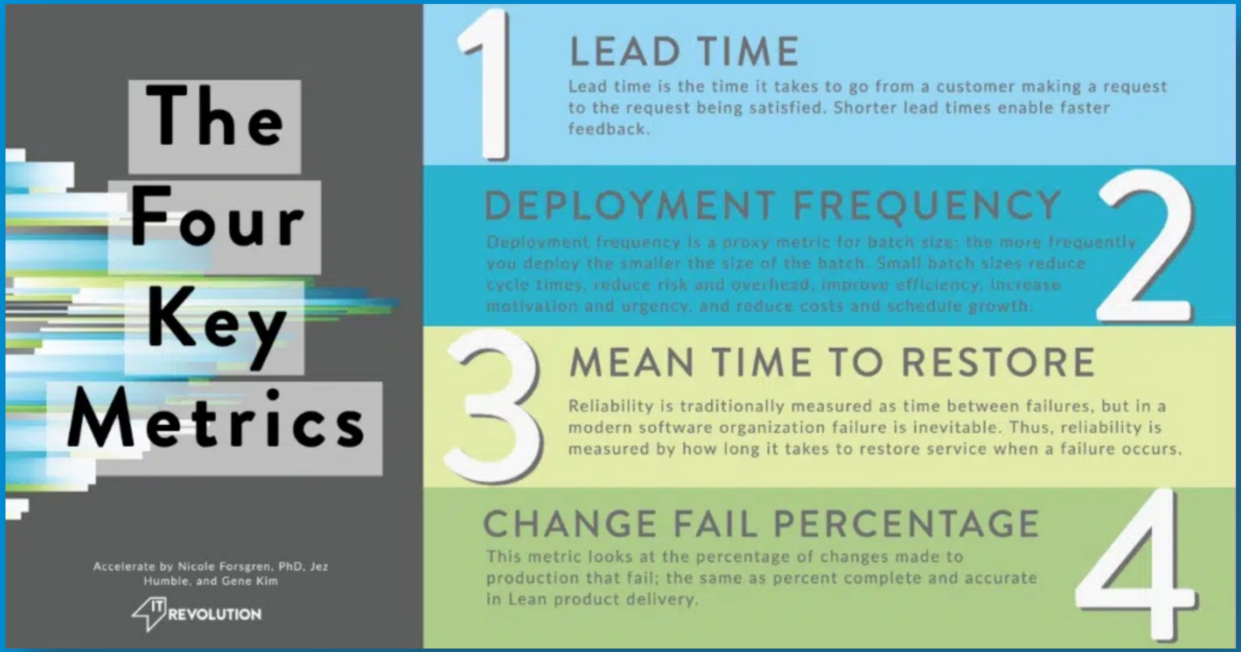
Kent Beck / Software Design: Tidy First? and pragmaticengineer.com

Source: "Measuring developer productivity? A response to McKinsey," Kent Beck and Gergely Orosz, August 2023.

At the Metrics Buffett

DORA

SPACE



The Four Key Metrics

Accelerate by Nicole Forsgren, PhD, Jez Humble, and Gene Kim

IT REVOLUTION

1 LEAD TIME

Lead time is the time it takes to go from a customer making a request to the request being satisfied. Shorter lead times enable faster feedback.

2 DEPLOYMENT FREQUENCY

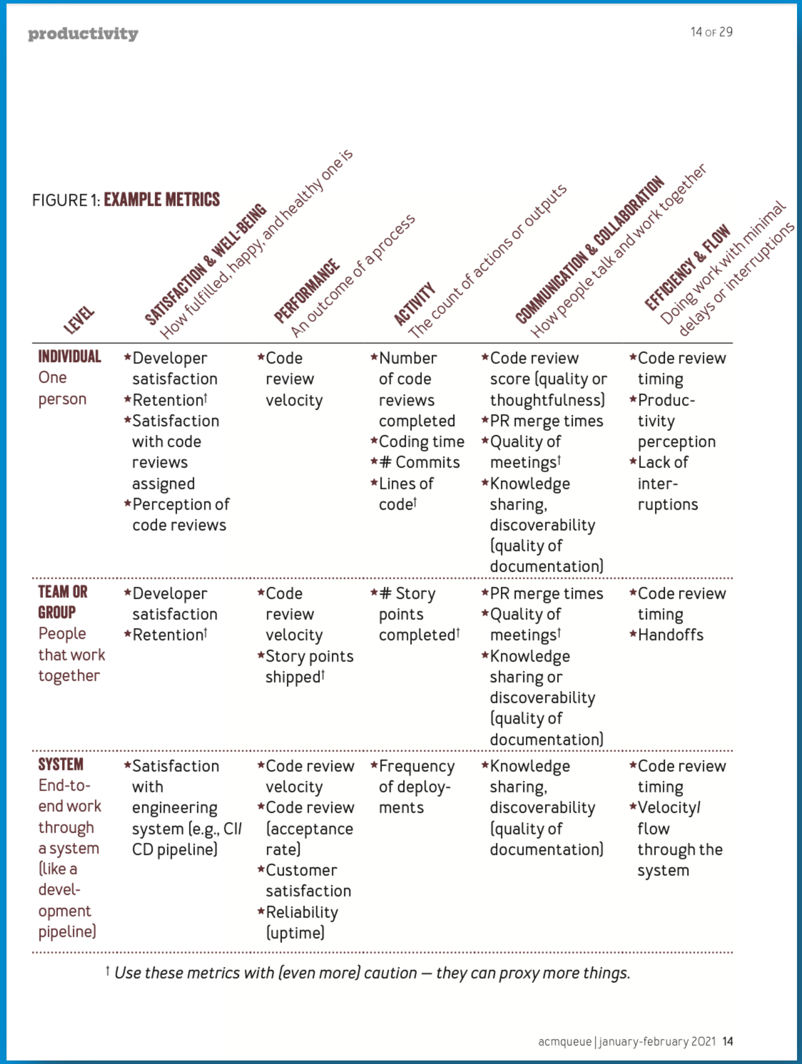
Deployment frequency is a proxy metric for batch size: the more frequently you deploy the smaller the size of the batch. Small batch sizes reduce cycle times, reduce risk and overhead, improve efficiency, increase motivation and urgency, and reduce costs and schedule growth.

3 MEAN TIME TO RESTORE

Reliability is traditionally measured as time between failures, but in a modern software organization failure is inevitable. Thus, reliability is measured by how long it takes to restore service when a failure occurs.

4 CHANGE FAIL PERCENTAGE

This metric looks at the percentage of changes made to production that fail; the same as percent complete and accurate in Lean product delivery.



productivity

14 of 29

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.*

acmqueue | january-february 2021 14

Source: "Measure Software Delivery Performance with Four Key Metrics," Nicole Forsgren ,Gene Kim, Jez Humble, Feb, 2021. "The SPACE of Developer Productivity," Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler, Jan-Feb, 2021. Also see: "Developer Thriving: The four factors that drive Software Developer Productivity across Industries," and, low usage of DORA and SPACE at tech companies from Abi Noda, Jan, 2024.

Happiness, flow, thriving, features

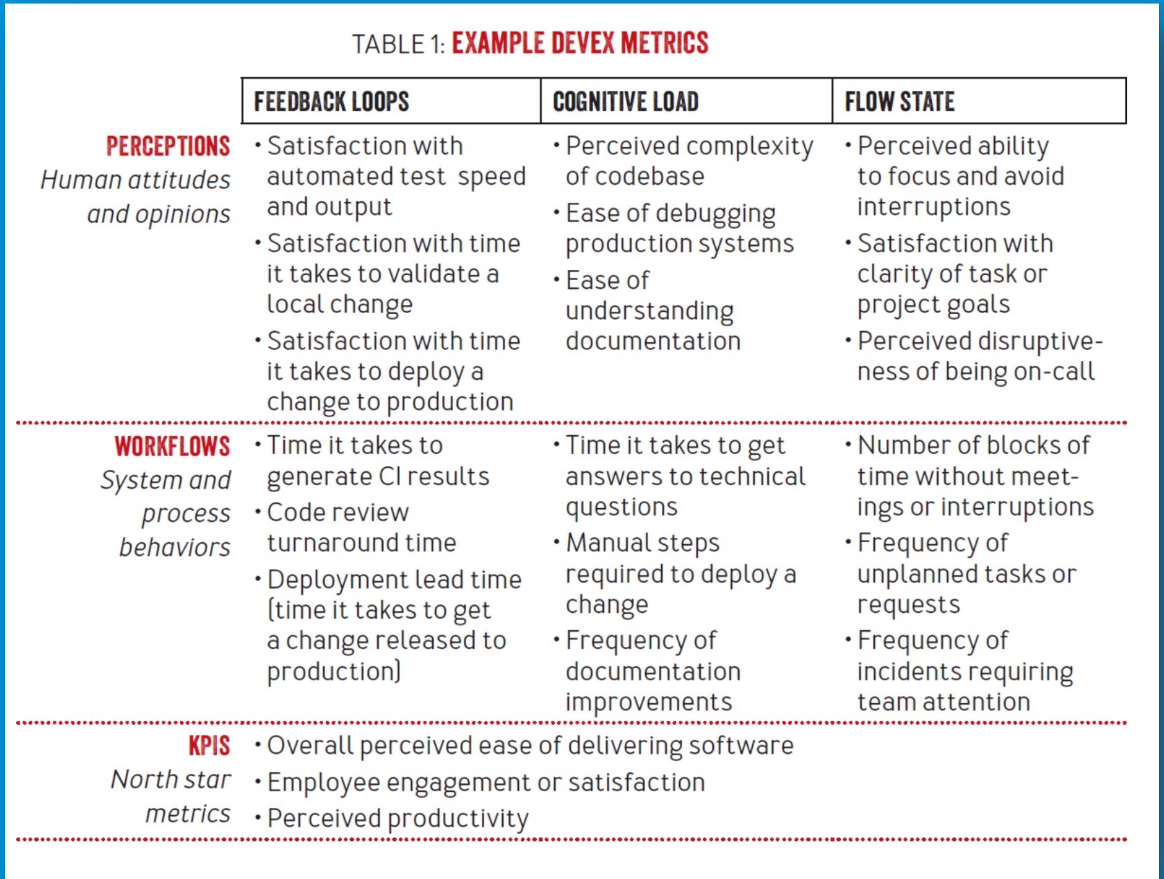


TABLE 1: **EXAMPLE DEVEX METRICS**

	FEEDBACK LOOPS	COGNITIVE LOAD	FLOW STATE
PERCEPTIONS *Human attitudes and opinions*	• 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	• Perceived complexity of codebase • Ease of debugging production systems • Ease of understanding documentation	• Perceived ability to focus and avoid interruptions • Satisfaction with clarity of task or project goals • Perceived disruptiveness of being on-call
WORKFLOWS *System and process behaviors*	• Time it takes to generate CI results • Code review turnaround time • Deployment lead time (time it takes to get a change released to production)	• Time it takes to get answers to technical questions • Manual steps required to deploy a change • Frequency of documentation improvements	• Number of blocks of time without meetings or interruptions • Frequency of unplanned tasks or requests • Frequency of incidents requiring team attention
KPIS *North star metrics*	• Overall perceived ease of delivering software • Employee engagement or satisfaction • Perceived productivity		

DevX framework

Causes of thriving	Because a developer is…
Agency	1) able to voice disagreement with team definitions of success 2) has a voice in how their contributions are measured
Motivation & Self-Efficacy	1) motivated when working on code at work 2) can see tangible progress most of the time 3) is working on the type of code work they want to work on 4) is confident that even when working in code is unexpectedly difficult, they will solve their problems
Learning Culture	1) learning new skills as a developer 2) able to share the things they learn at work
Support & Belonging	1) supported to grow, learn, and make mistakes by their team 2) agrees they are accepted for who they are by their team

Developer Thriving framework

Source: "DevEx: What Actually Drives Productivity," Abi Noda, Margaret-Anne Storey, Nicole Forsgren, Microsoft Research, Michaela Greiler, May 2023. See earlier study: "The SPACE of Developer Productivity: There's more to it than you think," Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler, Jan-Feb, 2021; Developer Thriving framework and studies, Cat Hicks et. al., March, 2023;

Developer Productivity Tools

Find the Developer Toil, Confusion, Blockers

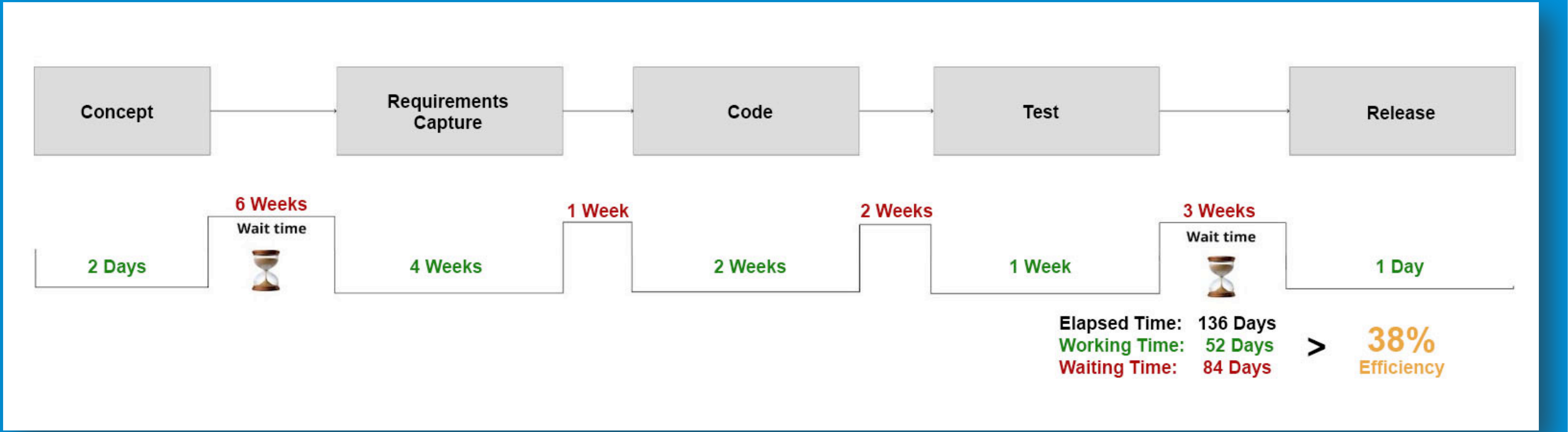
Find the Developer Toil, Confusion, Blockers

- 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.

Put CI/CD in place

Waste is outside the box



Concept

Requirements Capture

Code

Test

Release

2 Days

6 Weeks

Wait time

4 Weeks

1 Week

2 Weeks

2 Weeks

1 Week

3 Weeks

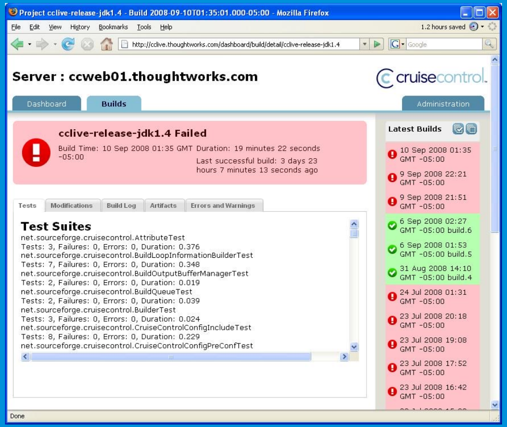
Wait time

1 Day

Elapsed Time: 136 Days
Working Time: 52 Days
Waiting Time: 84 Days

> 38%
Efficiency

Sources: ”The Power of a Path-to-Production Workshop,” Bryan Ross, September 2023.


Project cclive-release-jdk1.4 - Build 2008-09-10T01:35:01.000-05:00 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
1.2 hours saved
http://cclive.thoughtworks.com/dashboard/build/detail/cclive-release-jdk1.4
Google
cruisecontrol
Dashboard
Builds
Administration
Tests
Modifications
Build Log
Artifacts
Errors and Warnings
Done

Server : ccweb01.thoughtworks.com

cclive-release-jdk1.4 Failed

Build Time: 10 Sep 2008 01:35 GMT -05:00 Duration: 19 minutes 22 seconds

Last successful build: 3 days 23 hours 7 minutes 13 seconds ago

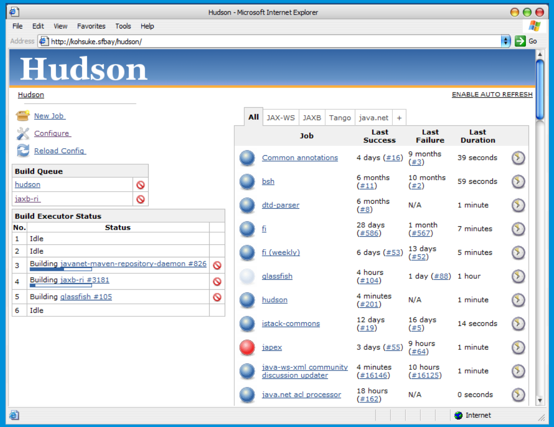
Test Suites

```
net.sourceforge.cruisecontrol.AttributeTest
Tests: 3, Failures: 0, Errors: 0, Duration: 0.376
net.sourceforge.cruisecontrol.BuildLoopInformationBuilderTest
Tests: 7, Failures: 0, Errors: 0, Duration: 0.348
net.sourceforge.cruisecontrol.BuildOutputBufferManagerTest
Tests: 2, Failures: 0, Errors: 0, Duration: 0.019
net.sourceforge.cruisecontrol.BuildQueueTest
Tests: 2, Failures: 0, Errors: 0, Duration: 0.039
net.sourceforge.cruisecontrol.BuilderTest
Tests: 3, Failures: 0, Errors: 0, Duration: 0.024
net.sourceforge.cruisecontrol.CruiseControlConfigIncludeTest
Tests: 8, Failures: 0, Errors: 0, Duration: 0.229
net.sourceforge.cruisecontrol.CruiseControlConfigPreConfTest
```

Latest Builds

- 10 Sep 2008 01:35 GMT -05:00
- 9 Sep 2008 22:21 GMT -05:00
- 9 Sep 2008 21:51 GMT -05:00
- 6 Sep 2008 02:27 GMT -05:00 build.6
- 6 Sep 2008 01:53 GMT -05:00 build.5
- 31 Aug 2008 14:10 GMT -05:00 build.4
- 24 Jul 2008 01:31 GMT -05:00
- 23 Jul 2008 20:18 GMT -05:00
- 23 Jul 2008 19:08 GMT -05:00
- 23 Jul 2008 17:52 GMT -05:00
- 23 Jul 2008 16:42 GMT -05:00

2001


Hudson - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://kohsuke.sfbay/hudson/
Go
Hudson
Hudson
ENABLE AUTO REFRESH
New Job
Configure
Reload Config
All
JAX-WS
JAXB
Tango
java.net
+
Internet

Build Queue

hudson
jaxb-ri

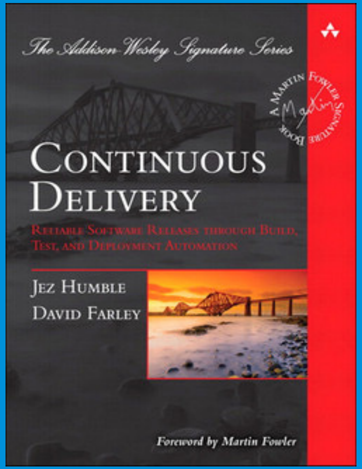
Build Executor Status

No.	Status
1	Idle
2	Idle
3	Building javanet-maven-repository-daemon #826
4	Building jaxb-ri #3181
5	Building glassfish #105
6	Idle

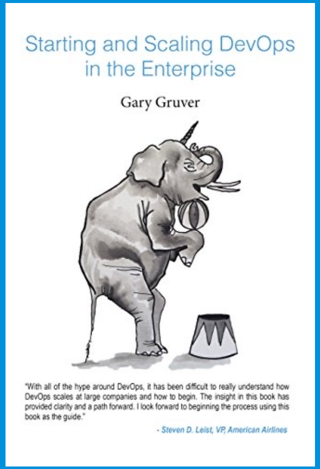
Job	Last Success	Last Failure	Last Duration
Common annotations	4 days (#16)	9 months (#3)	39 seconds
bsh	6 months (#11)	10 months (#2)	59 seconds
dtd-parser	6 months (#8)	N/A	1 minute
fi	28 days (#586)	1 month (#567)	7 minutes
fi (weekly)	6 days (#53)	13 days (#52)	5 minutes
glassfish	4 hours (#104)	1 day (#88)	1 hour
hudson	4 minutes (#201)	N/A	1 minute
istack-commons	12 days (#19)	16 days (#5)	14 seconds
japex	3 days (#55)	9 hours (#64)	1 minute
java-ws-xml community discussion updater	4 minutes (#16146)	10 hours (#16125)	1 minute
java.net acl processor	18 hours (#162)	N/A	0 seconds

2005

2011


The Addison Wesley Signature Series
A Martin Fowler Signature Book
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
Foreword by Martin Fowler

2010

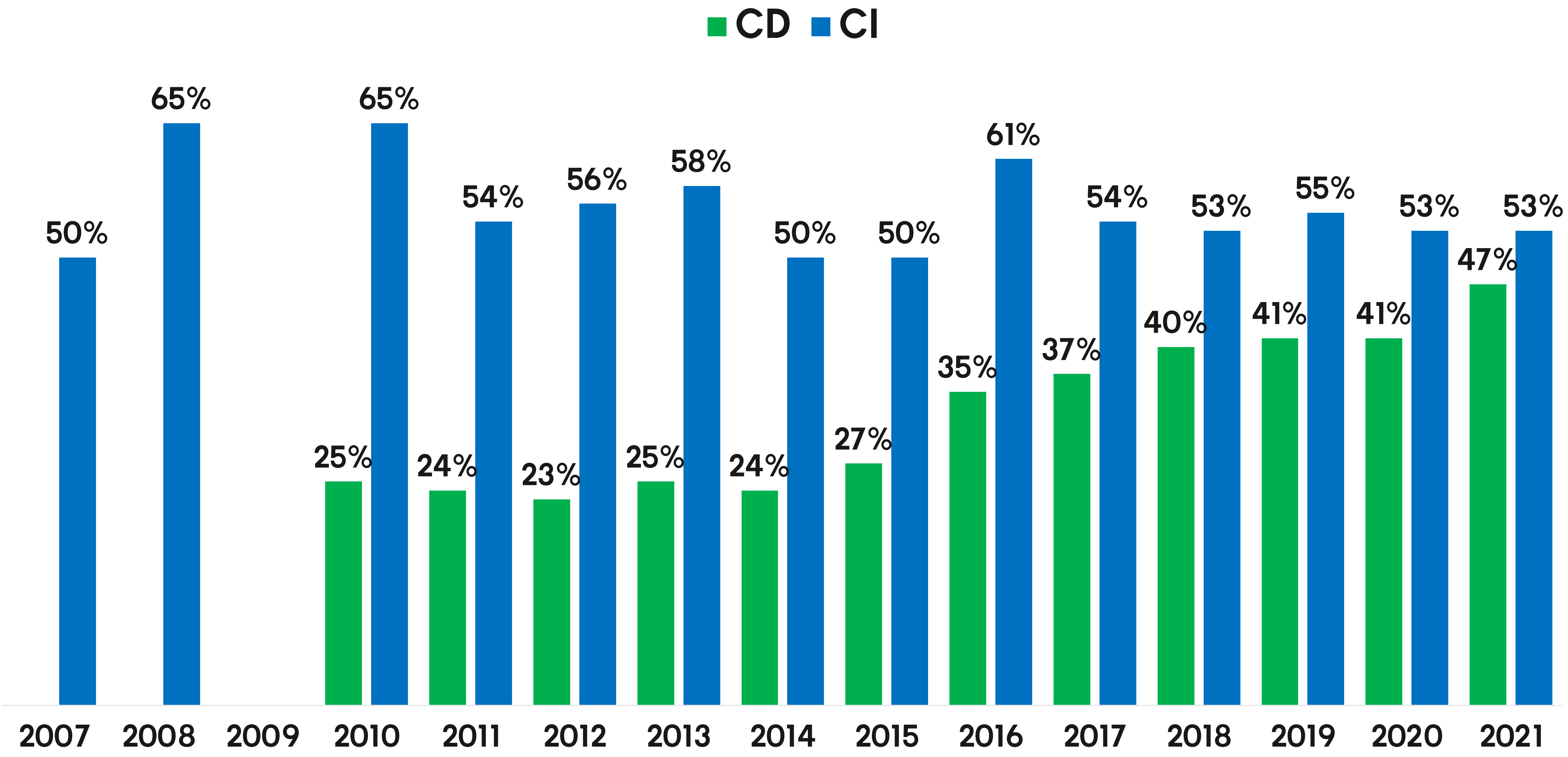

Starting and Scaling DevOps in the Enterprise
Gary Gruver
"With all of the hype around DevOps, it has been difficult to really understand how DevOps scales at large companies and how to begin. The insight in this book has provided clarity and a path forward. I look forward to beginning the process using this book as the guide."
- Steven D. Leist, VP, American Airlines

2016

"Printer firmware? Hold my beer."

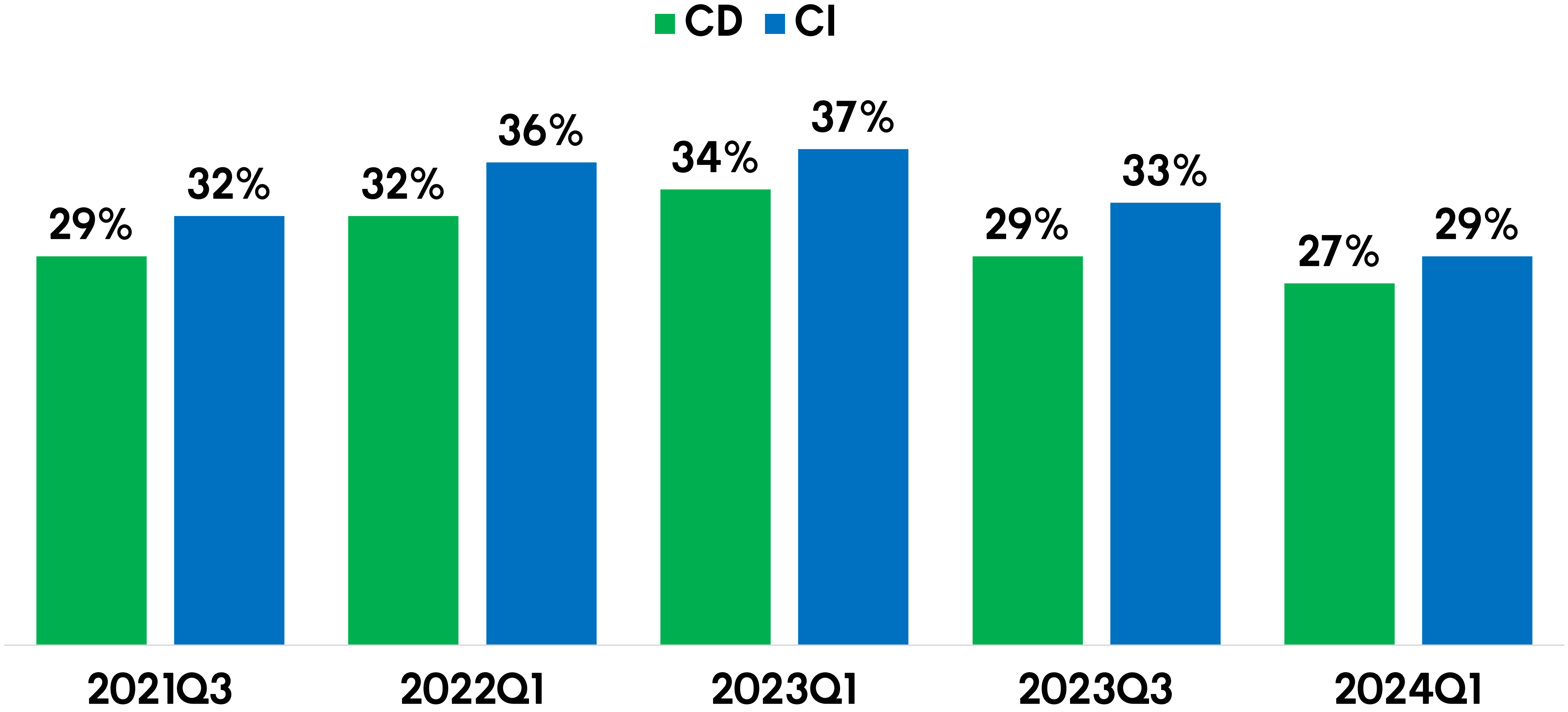
Note: originally attributed to Grady Booch in 1991, also XP principal in 1998. Sources: book listings, Sourceforge(!), Wikipedia on Sep 1st, 2023.

CI and CD usage, 2007 to 2021



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

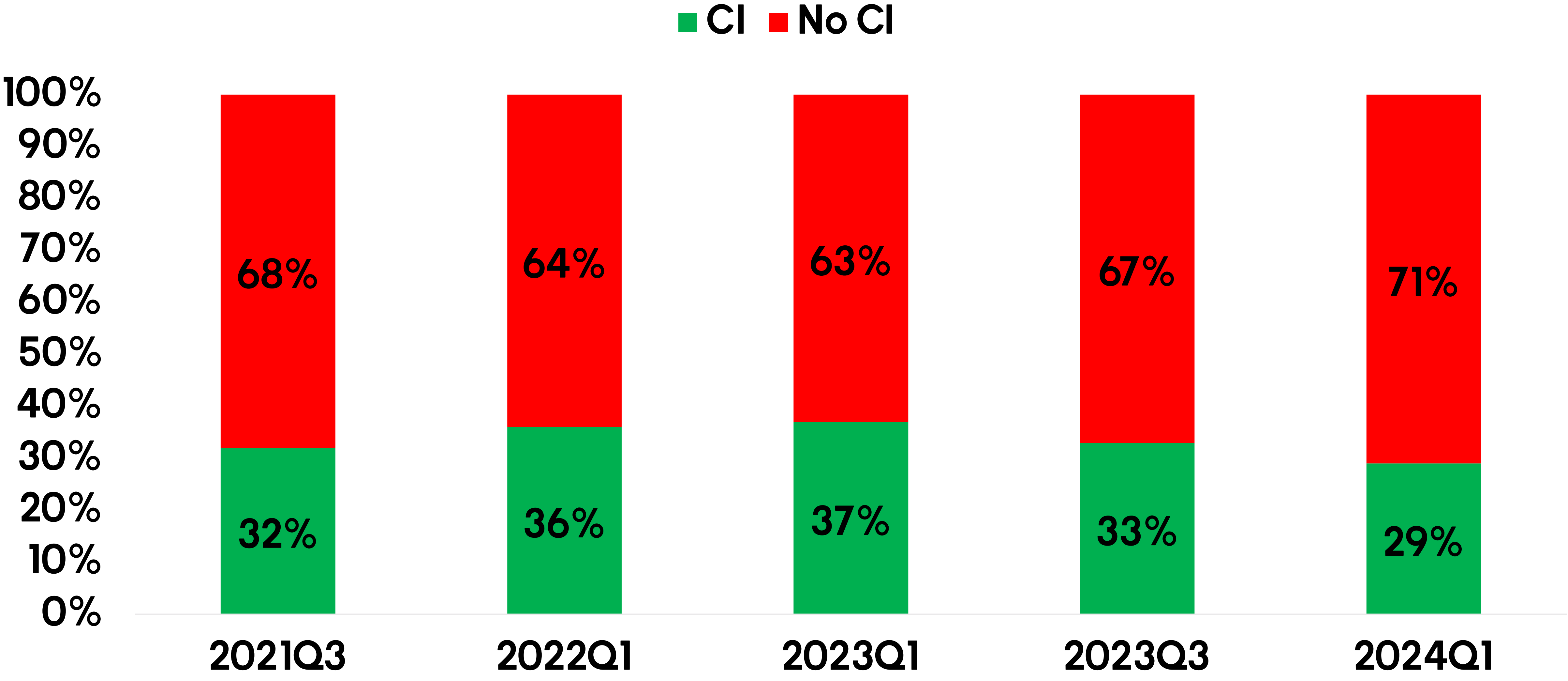
CI and CD Usage, 2021 to 2024



Question: Which of the following technologies have you used as part of your development activities in the last 12 months? Source: CD Foundation Surveys (Slashdata).

Source: "State of CI/CD Report 2024: The Evolution of Software Delivery Performance," CD Foundation and SlashData, April, 2024, and previous studies.

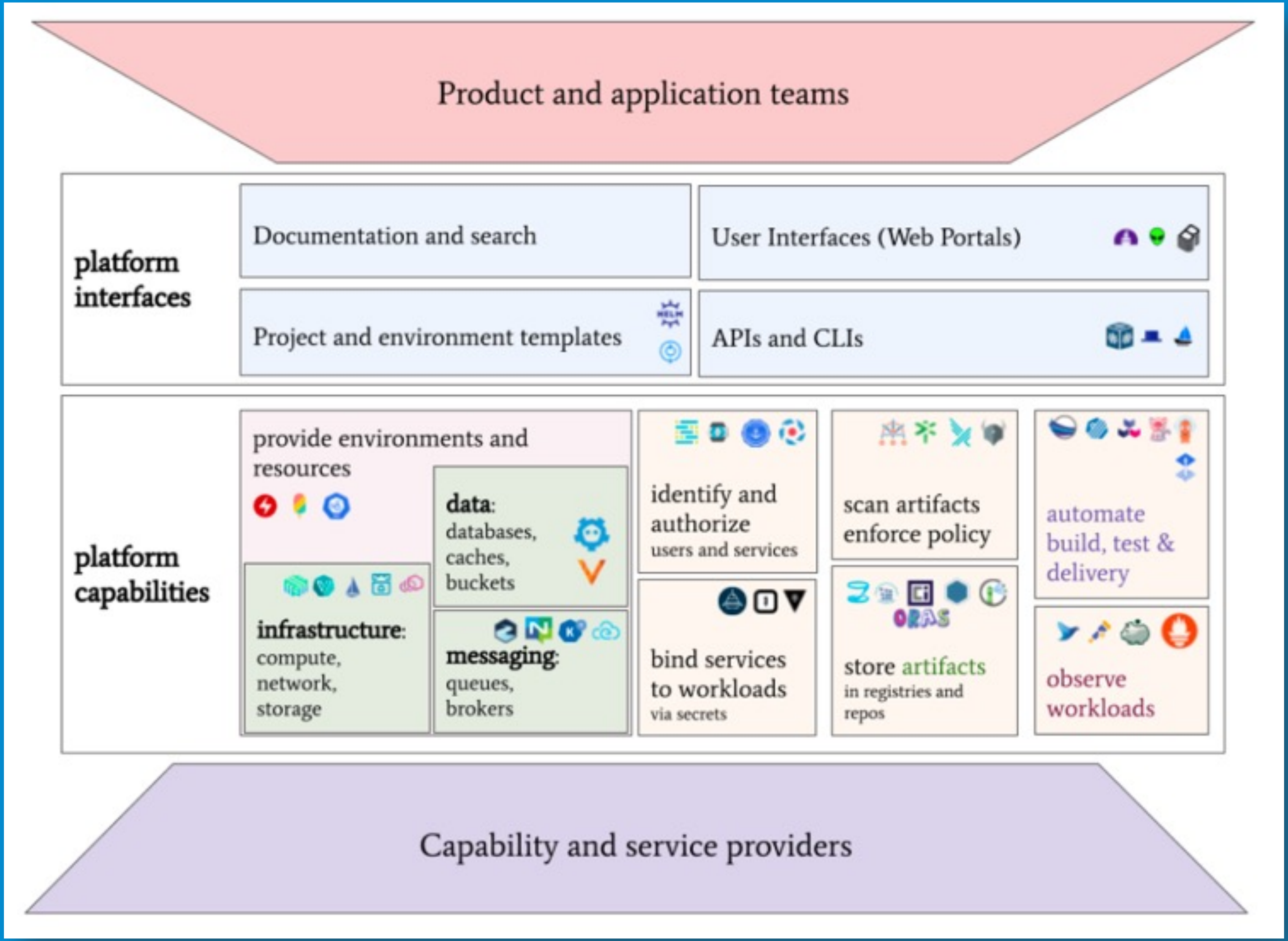
CI Usage, 2021 to 2024



Question: Which of the following technologies have you used as part of your development activities in the last 12 months? Source: CD Foundation Surveys (Slashdata).

Source: "State of CI/CD Report 2024: The Evolution of Software Delivery Performance," CD Foundation and SlashData, April, 2024, and previous studies.

Stop building your own platforms, or, at least, have less of them



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

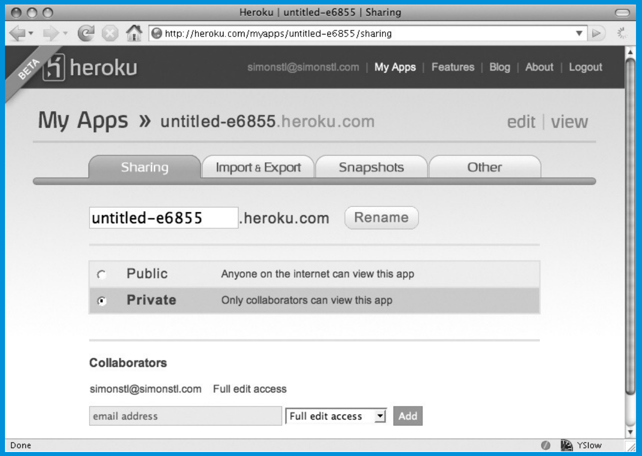
observe workloads

Capability and service providers

Source: "CNCF Platforms White Paper," March 2023.

The Eternal Recurrence of (Platforms, PaaS, DevOps, Cloud Native, Golden Paths, Platform Engineering, ...)

(╯°□°) ╯□ ┻━━┻)   (╯°□°) ╯□ ┻━━┻)


Heroku | untitled-e6855 | Sharing

http://heroku.com/myapps/untitled-e6855/sharing

BETA
heroku
simonstl@simonstl.com | My Apps | Features | Blog | About | Logout

My Apps » untitled-e6855.heroku.com

untitled-e6855 .heroku.com

○ Public Anyone on the internet can view this app

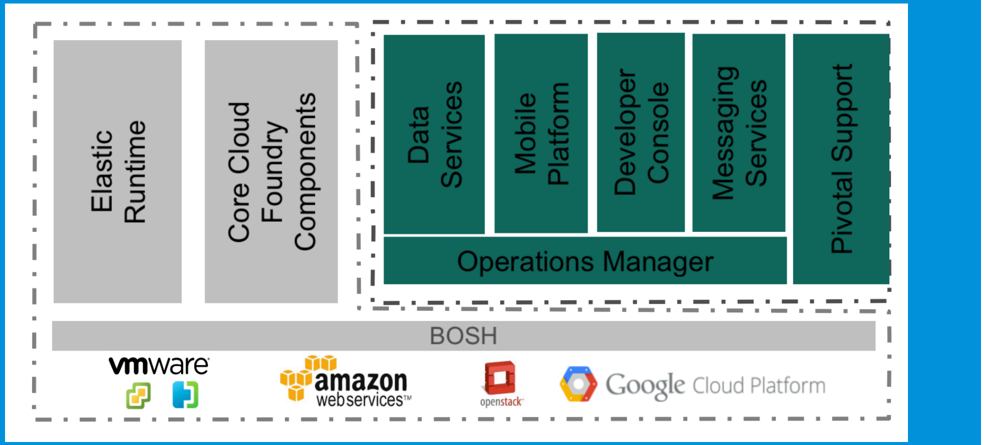
● Private Only collaborators can view this app

Collaborators

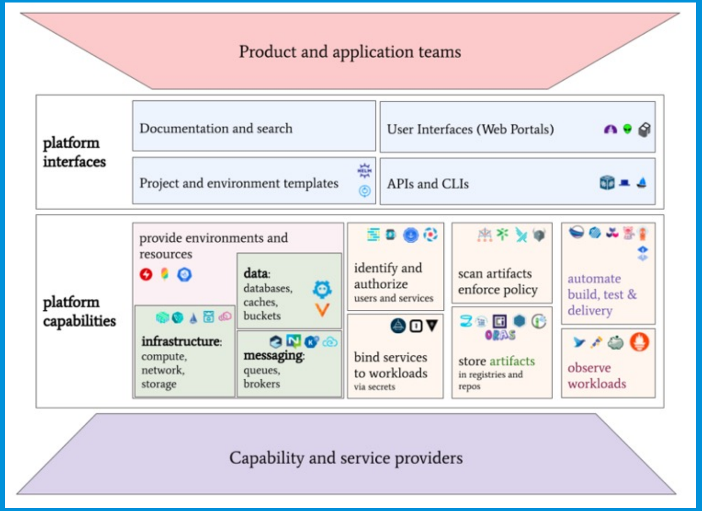
simonstl@simonstl.com Full edit access

email address Full edit access

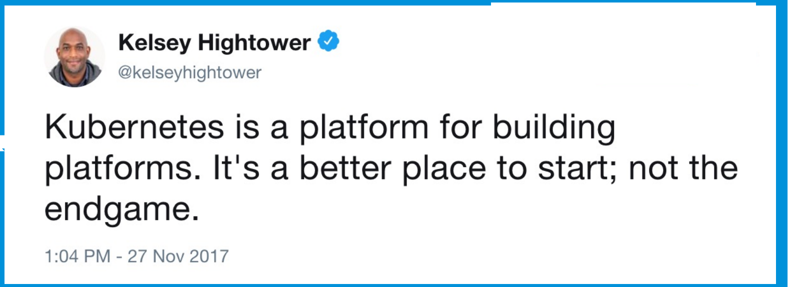
2007



2011 to 2015


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
infrastructure: compute, network, storage
data: databases, caches, buckets
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

2023 & Beyond

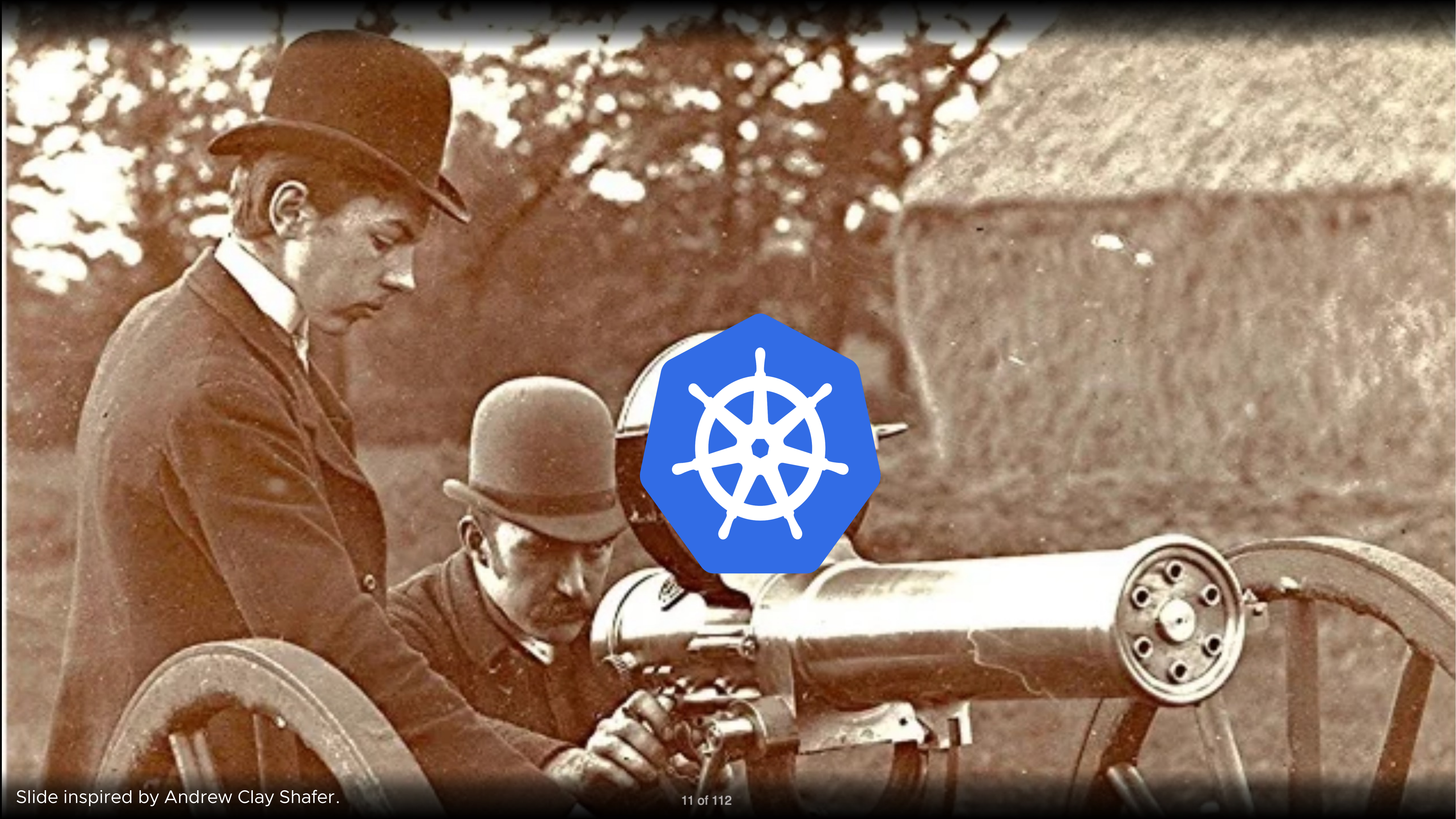

Kelsey Hightower
@kelseyhightower

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

1:04 PM - 27 Nov 2017

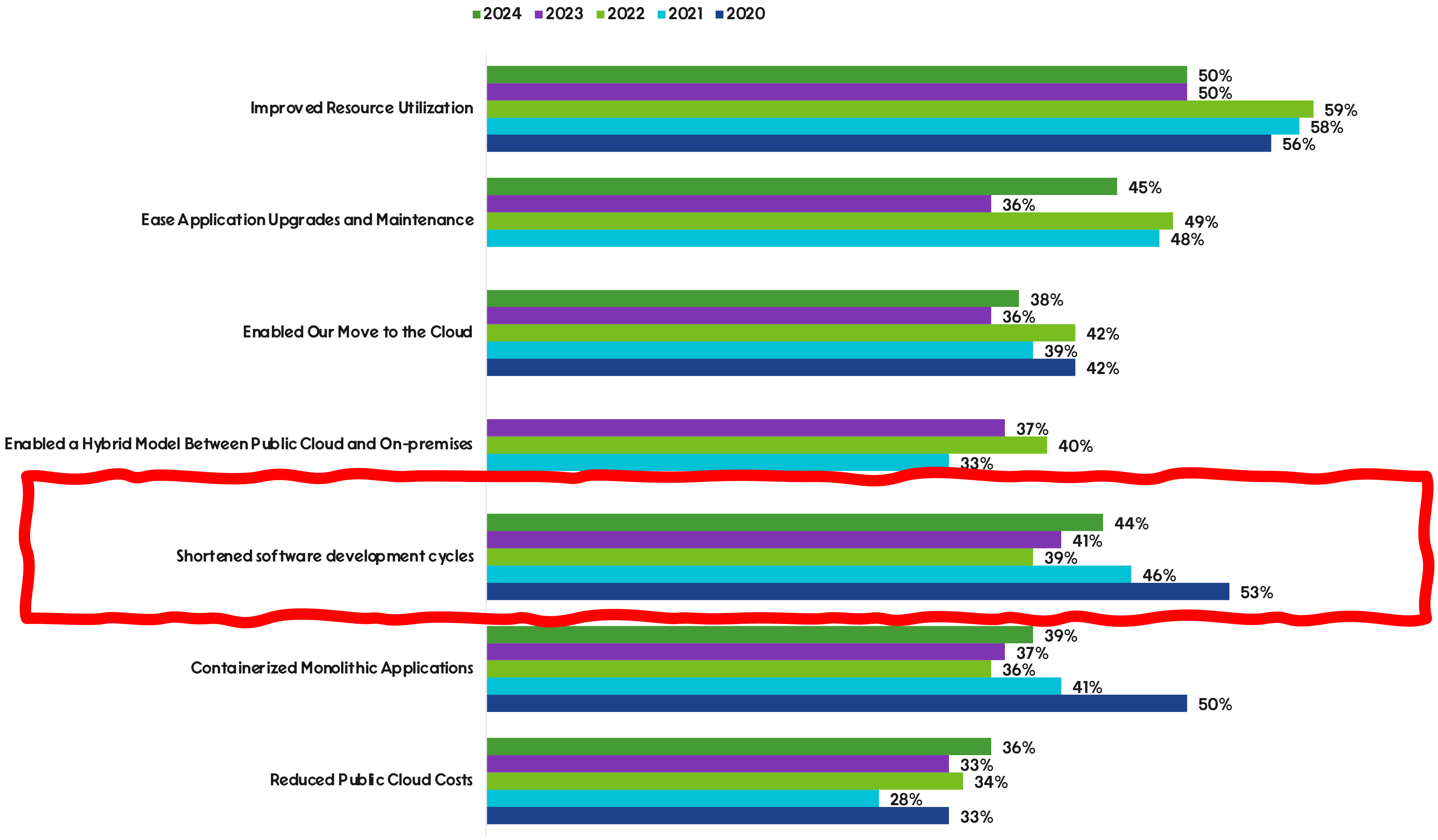
Not pictured:

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



Slide inspired by Andrew Clay Shafer.

What benefits has your organization realized from operating Kubernetes?



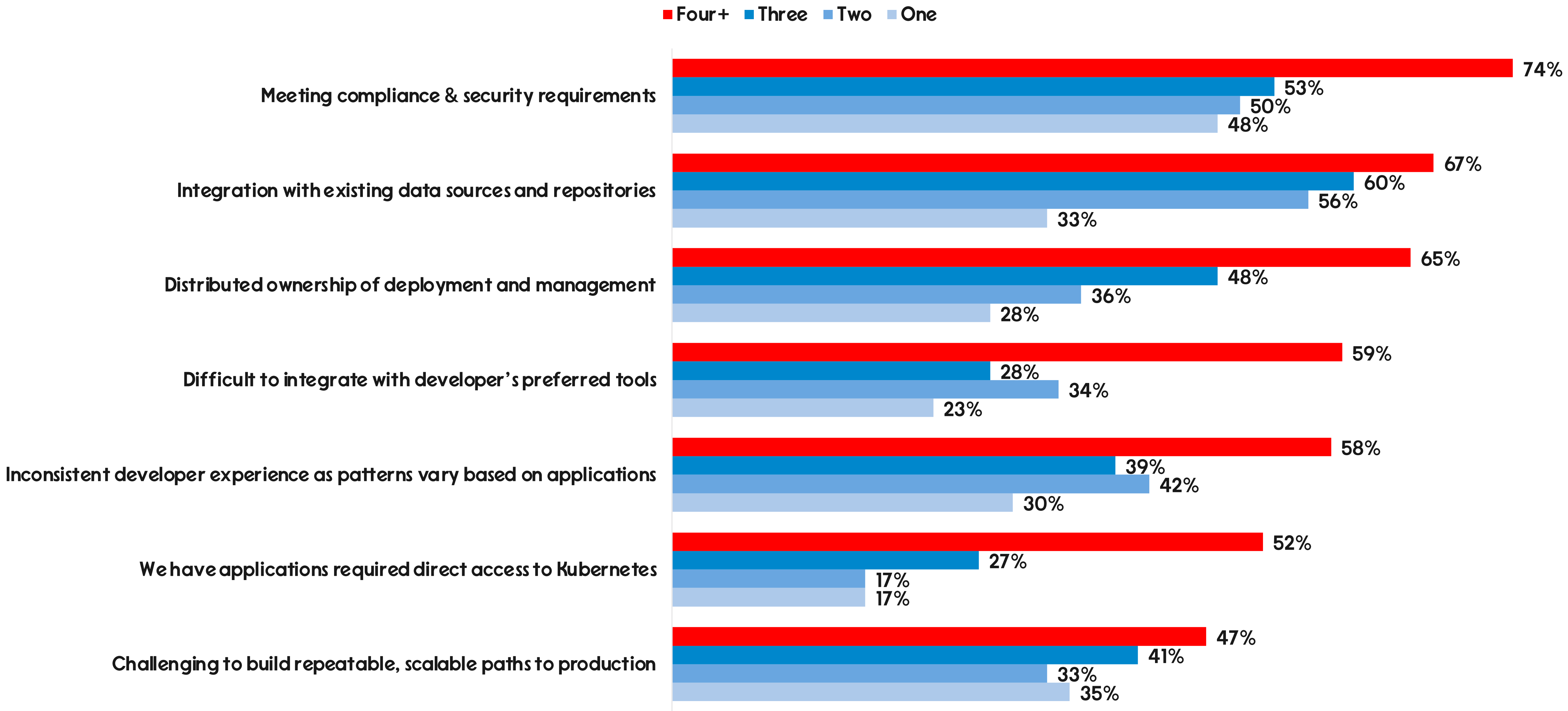
Source: State of Cloud Native Platform 2024, various State of Kubernetes. More: "Exploring the State of Cloud Native App Platforms and VMware Tanzu," July, 2024.

“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.”

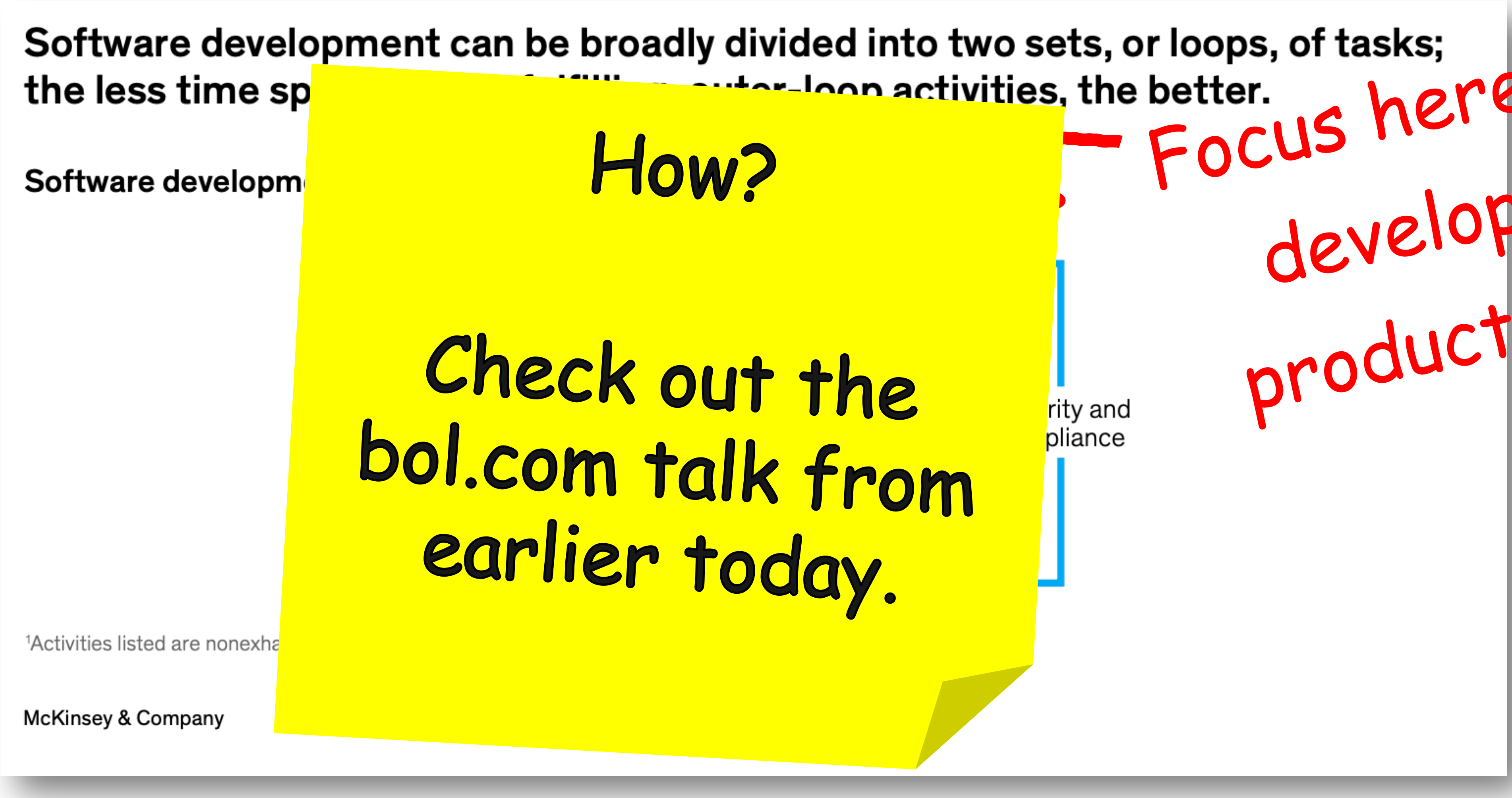
Craig McLuckie, SpringOne 2021

Challenges building and managing platforms

(by # of platforms in use)



Source: "The State of Cloud Native Application Platforms 2024," VMware Tanzu by Broadcom, June, 2023. 760 respondents at orgs. 1,000+ employees.



Software development can be broadly divided into two sets, or loops, of tasks; the less time sp

outer-loop activities, the better.

Software developm

rity and
pliance

[1]Activities listed are nonexha

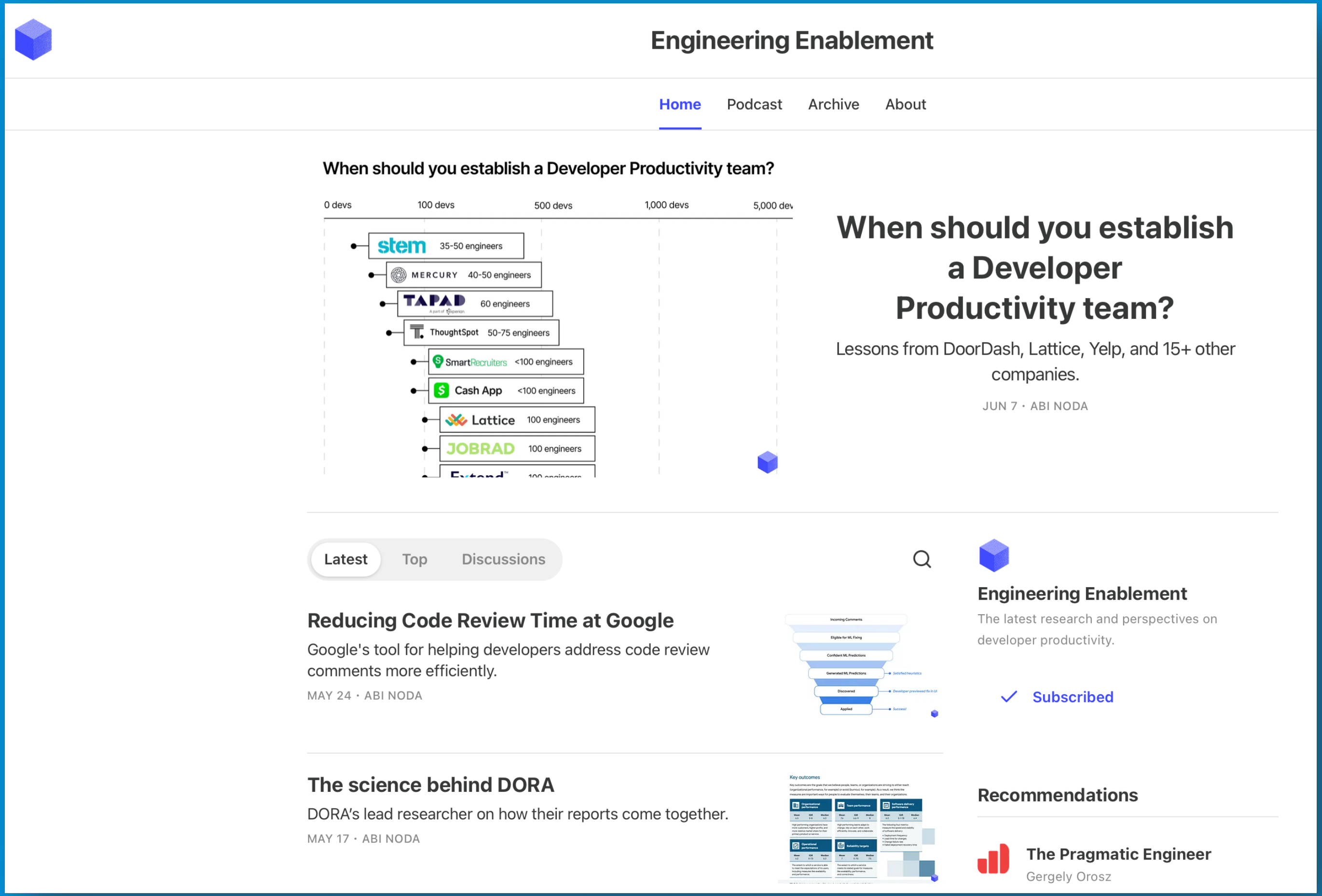
McKinsey & Company

How?

Check out the bol.com talk from earlier today.

Focus here for developer productivity

Source: "Yes, you can measure software developer productivity," Chandra Gnanasambandam, Martin Harrysson, Alharith Hussin, Jason Keovichit, Shivam Srivastava, McKinsey Aug(?) 2023.



Engineering Enablement

Home Podcast Archive About

When should you establish a Developer Productivity team?

Lessons from DoorDash, Lattice, Yelp, and 15+ other companies.

JUN 7 · ABI NODA

Latest Top Discussions

Reducing Code Review Time at Google

Google's tool for helping developers address code review comments more efficiently.

MAY 24 · ABI NODA

The science behind DORA

DORA's lead researcher on how their reports come together.

MAY 17 · ABI NODA

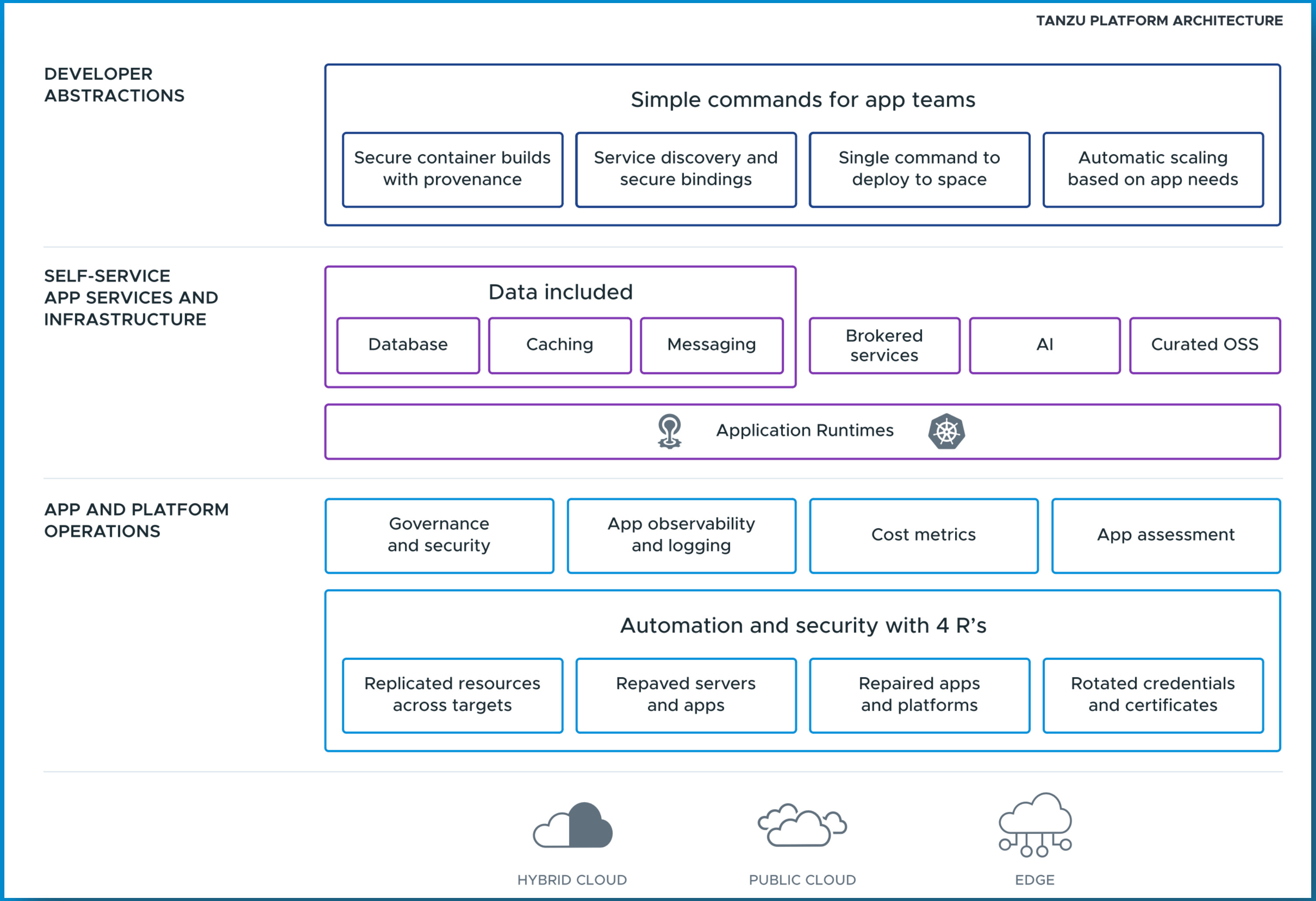
Engineering Enablement

The latest research and perspectives on developer productivity.

✓ Subscribed

Recommendations

The Pragmatic Engineer
Gergely Orosz



TANZU PLATFORM ARCHITECTURE

DEVELOPER ABSTRACTIONS

Simple commands for app teams

- Secure container builds with provenance
- Service discovery and secure bindings
- Single command to deploy to space
- Automatic scaling based on app needs

SELF-SERVICE APP SERVICES AND INFRASTRUCTURE

Data included

- Database
- Caching
- Messaging

- Brokered services
- AI
- Curated OSS

Application Runtimes

APP AND PLATFORM OPERATIONS

- Governance and security
- App observability and logging
- Cost metrics
- App assessment

Automation and security with 4 R’s

- Replicated resources across targets
- Repaved servers and apps
- Repaired apps and platforms
- Rotated credentials and certificates

Thank You!

🏗 https://tanzu.vmware.com/platform

📩 https://newsletter.cote.io/

Slides!