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.

€ $ £

Use metrics to find & monitor happiness, flow, features

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 *Produc-tivity perception *Lack of inter-ruptions
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 devel-opment pipeline]	*Satisfaction with engineering system (e.g., CI/ CD pipeline]	*Code review velocity *Code review (acceptance rate] *Customer satisfaction *Reliability (uptime]	*Frequency of deploy-ments	*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.*

FIGURE 1: THREE CORE DIMENSIONS OF DEVELOPER EXPERIENCE

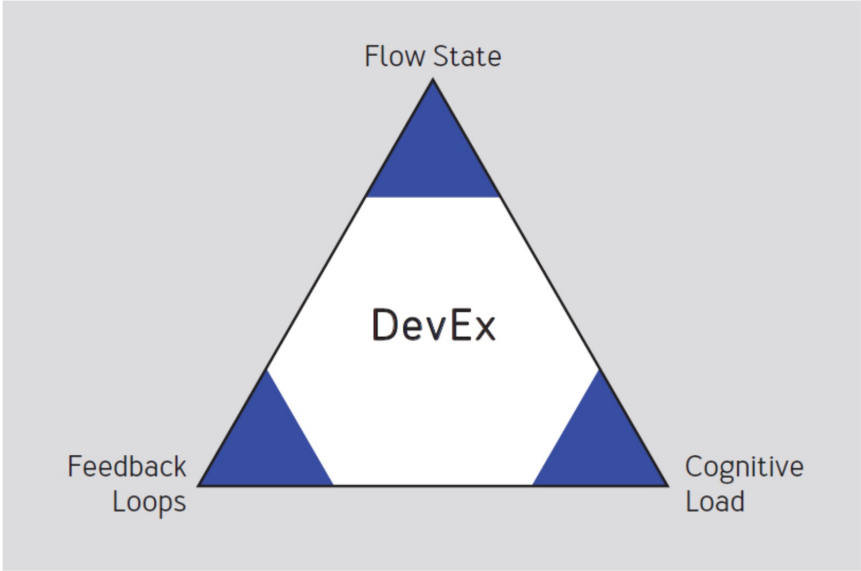


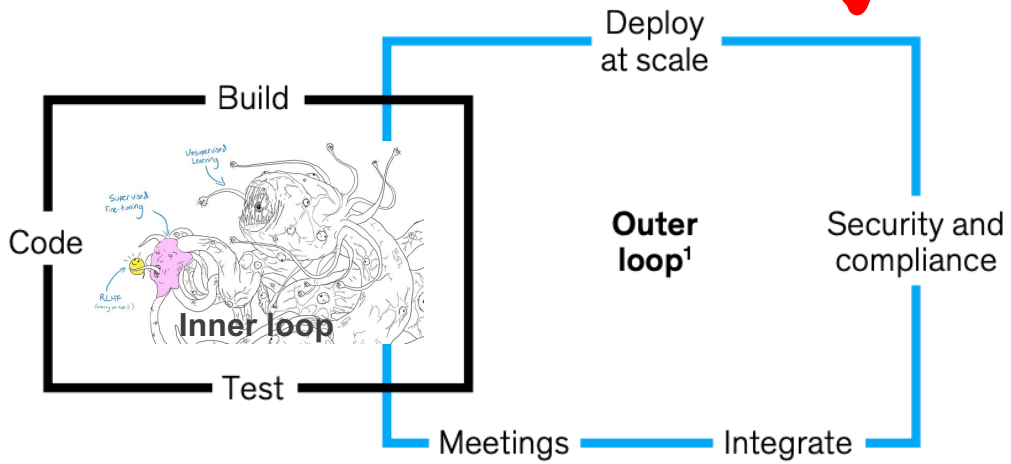
TABLE 1: EXAMPLE DEVEX METRICS

	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 disruptive-ness 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 meet-ings 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



[1]Activities listed are nonexhaustive.

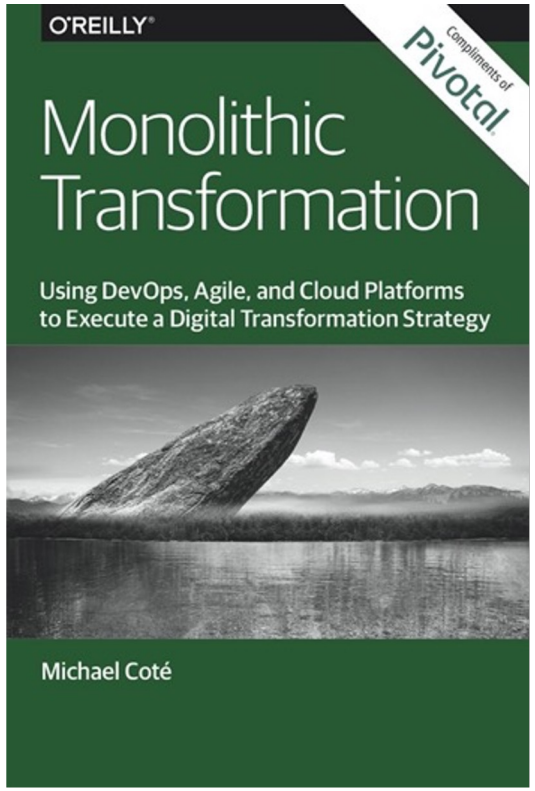
McKinsey & Company

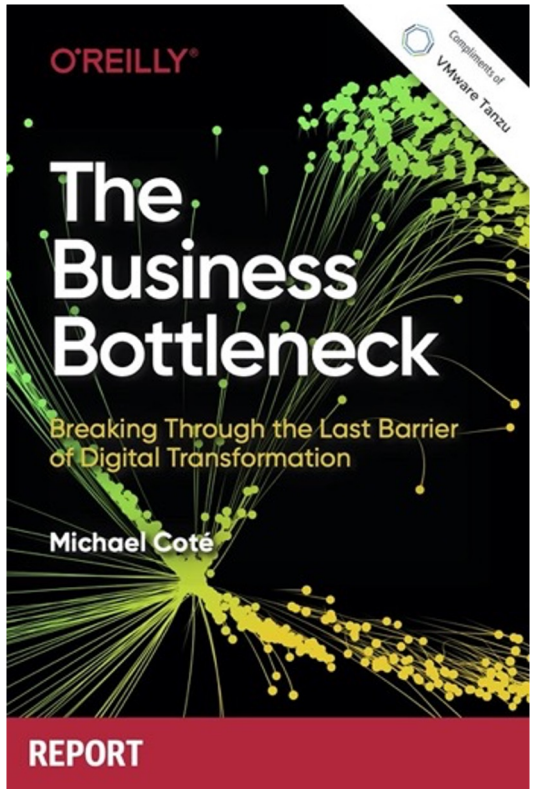
Focus here for developer productivity

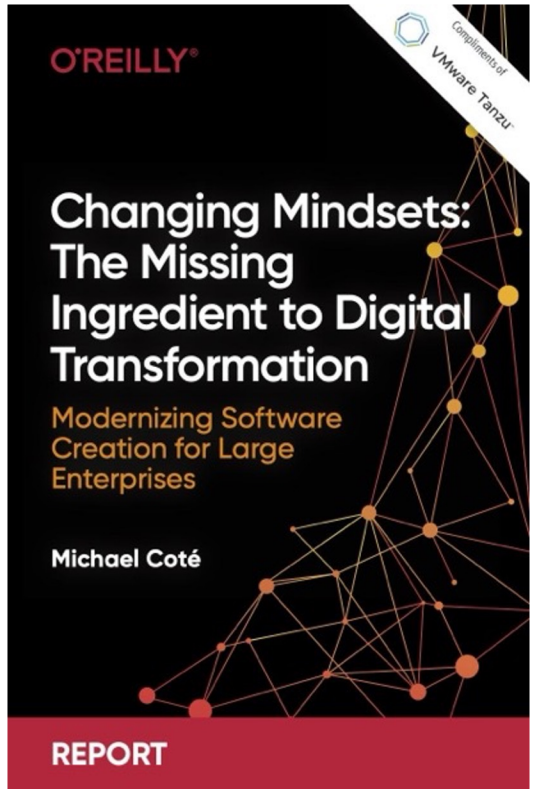
Source: "Yes, you can measure software developer productivity," Chandra Gnanasambandam, Martin Harrysson, Alharith Hussin, Jason Keovichit, Shivam Srivastava, McKinsey Aug(?) 2023.

Coté

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







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👎

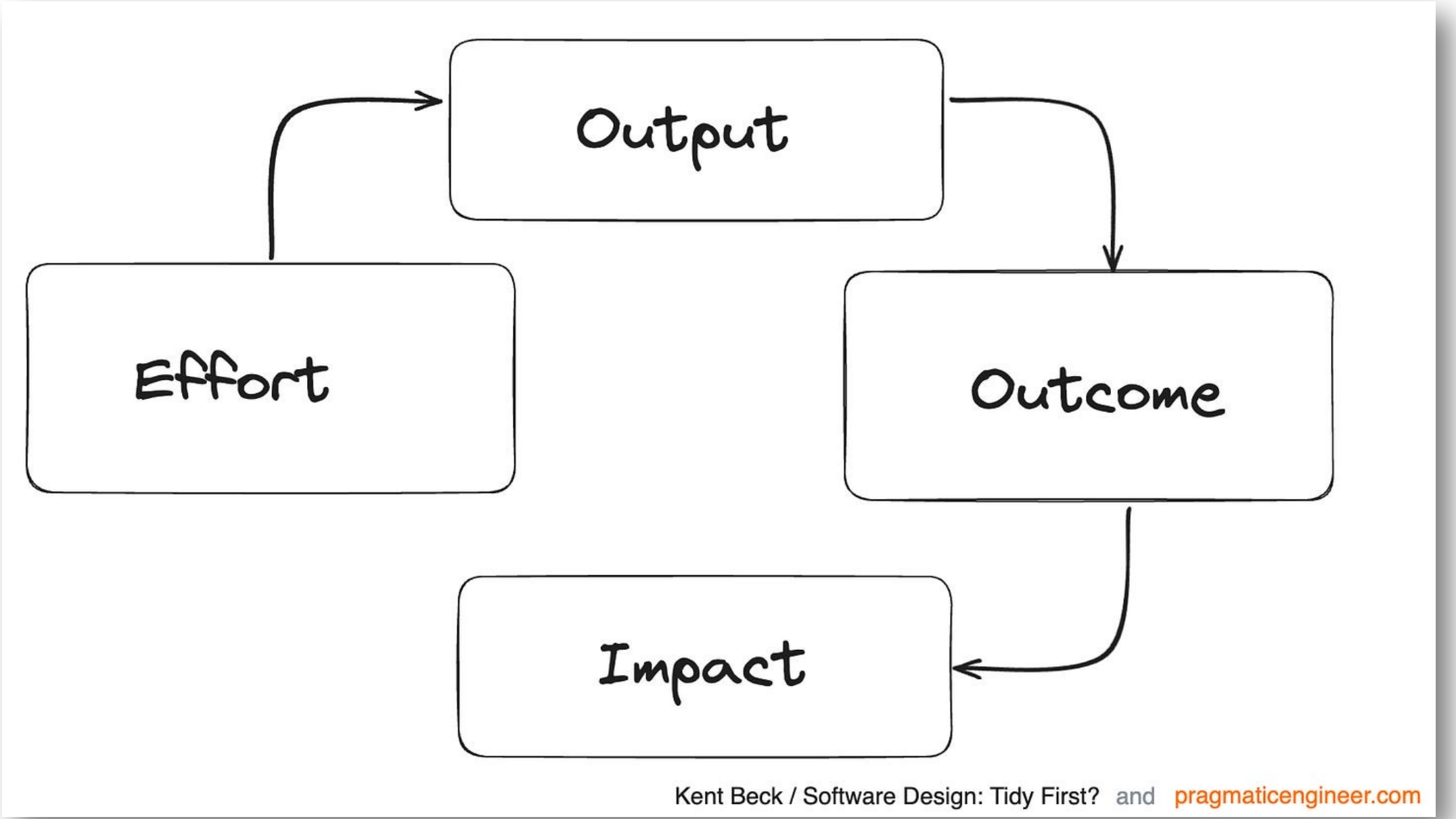
1. Are we paying too much?
2. Could we get by with paying less?
3. Who should I punish/fire?
4. (Who should I give more money to?)

(Also, sure,
craft optimization
& personal improvement)

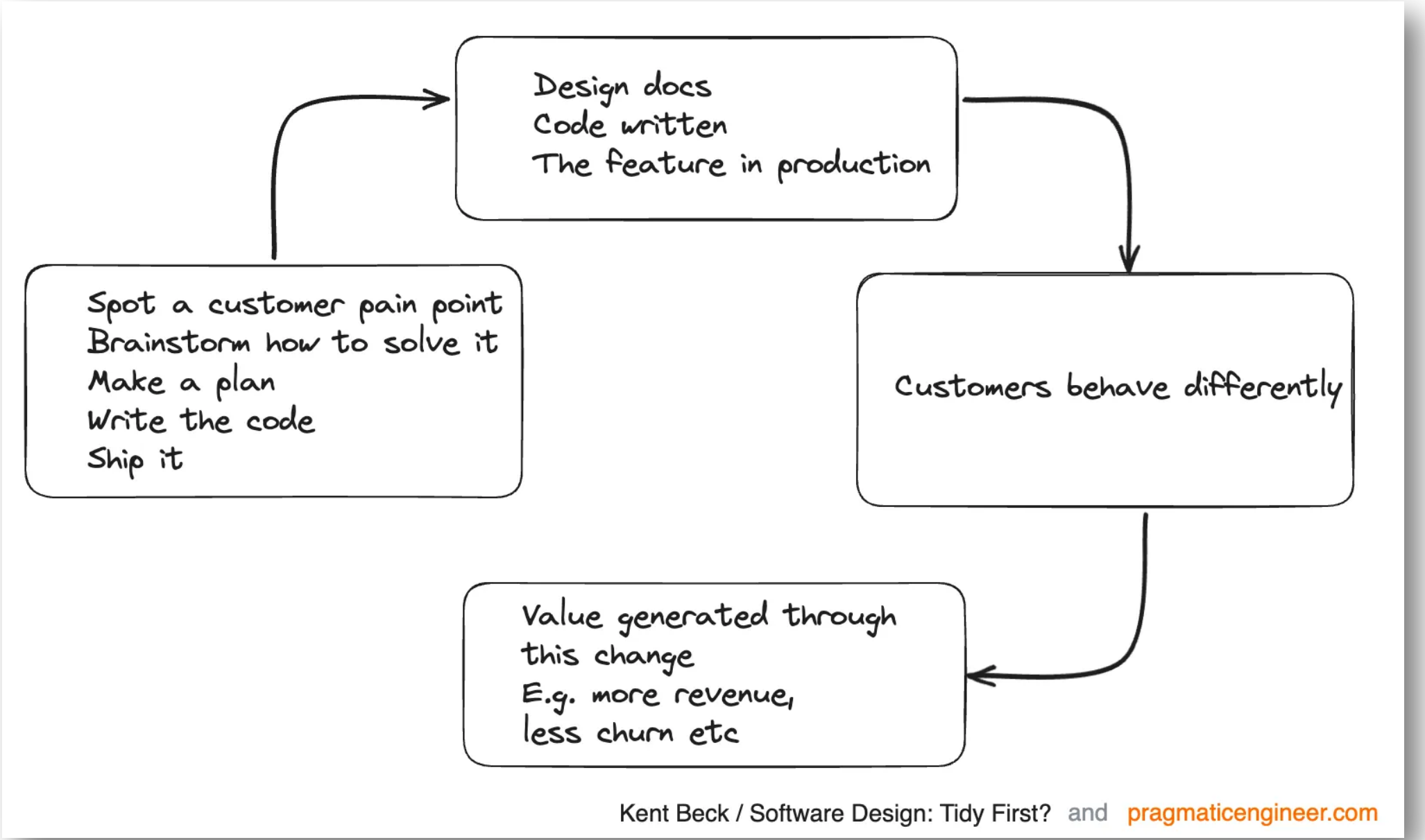
What is “developer productivity”?

“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.



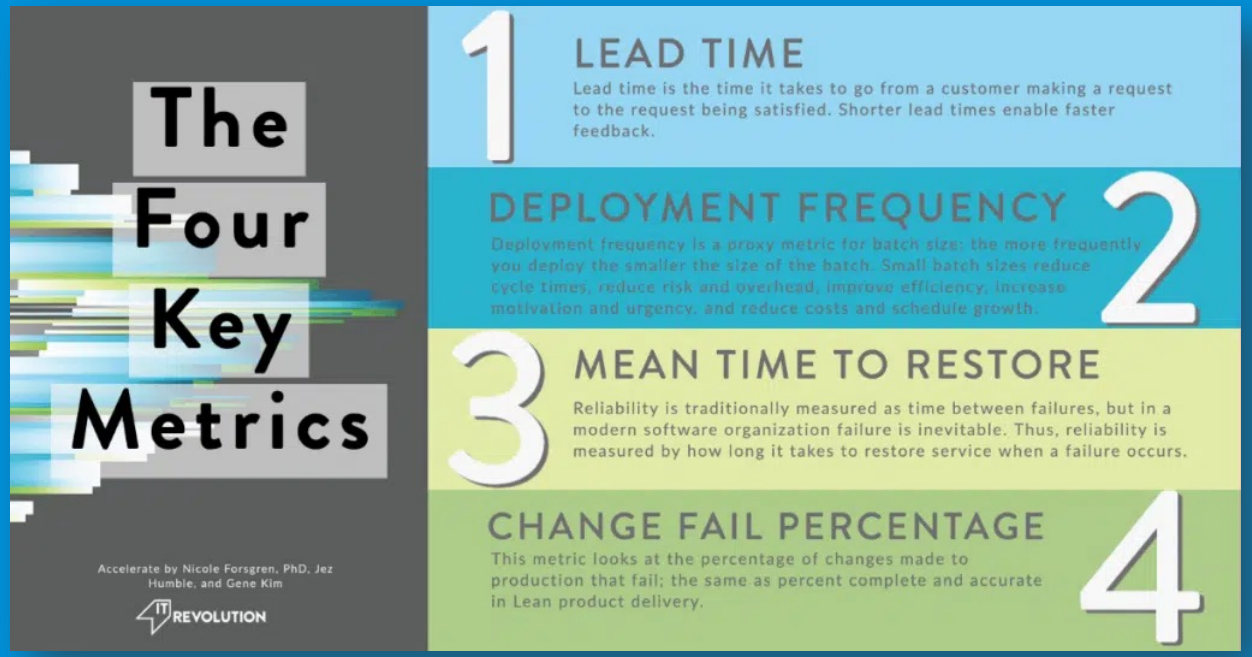
Source: "Measuring developer productivity? A response to McKinsey," Kent Beck and Gergely Orosz, August 2023.



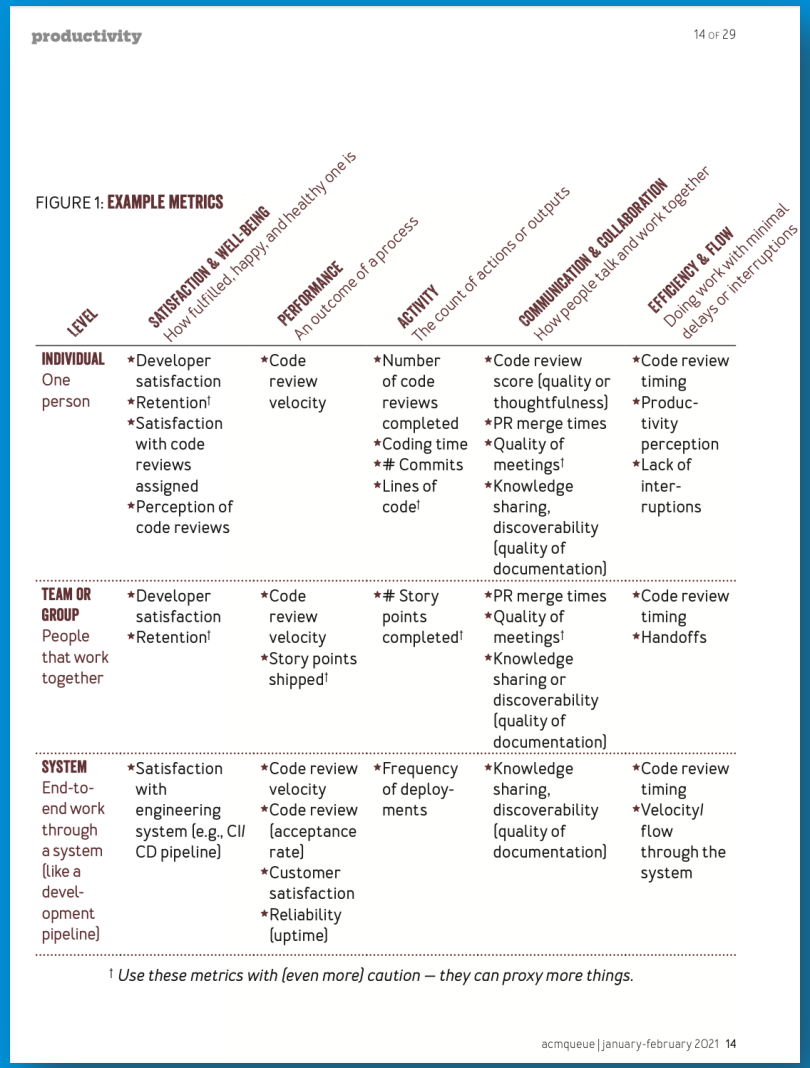
Source: "Measuring developer productivity? A response to McKinsey," Kent Beck and Gergely Orosz, August 2023.

At the Metrics Buffett

DORA



SPACE



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 *Produc-tivity perception *Lack of inter-ruptions
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 devel-opment pipeline)	*Satisfaction with engineering system (e.g., CI/ CD pipeline)	*Code review velocity *Code review (acceptance rate) *Customer satisfaction *Reliability (uptime)	*Frequency of deploy-ments	*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

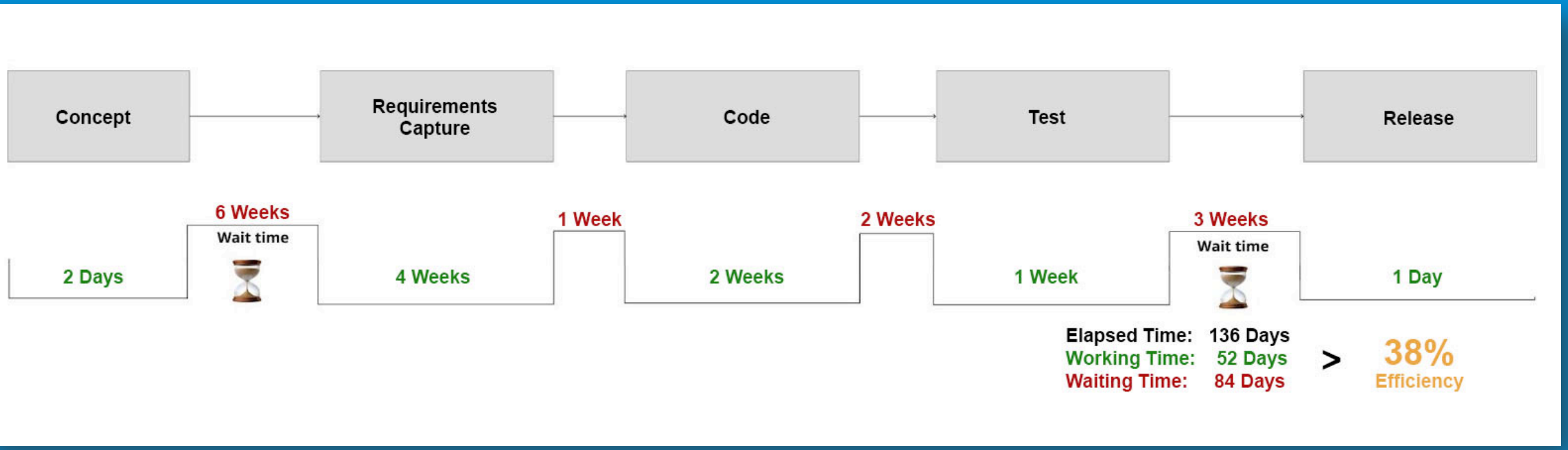
Find the Developer Toil, Confusion, Blockers

- 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.

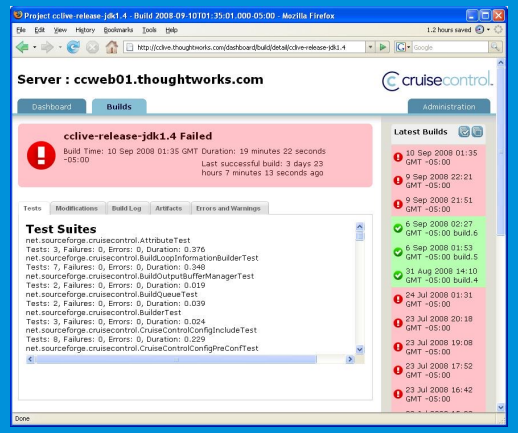
Sources: "Developer Toil: The Hidden Tech Debt," Susie Forbath, Tyson McNulty, and Coté, August, 2022. See also Michael Galloway's interview questions for platform product managers.

Put CI/CD in place

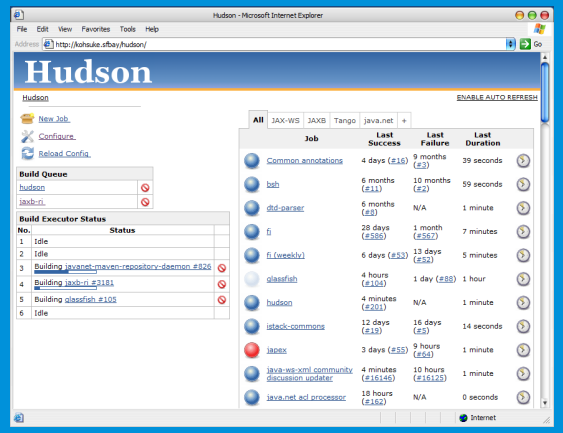
Waste is outside the box



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

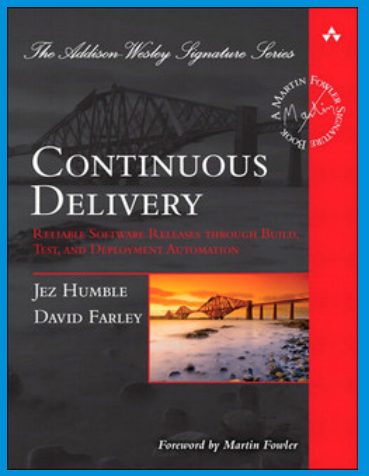


2001

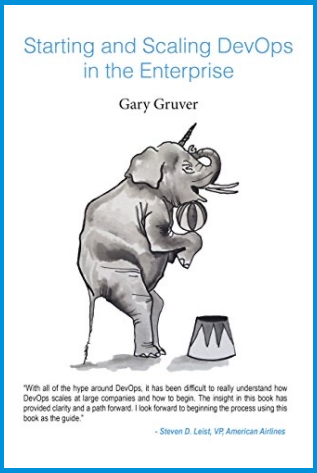


2005

2011



2010

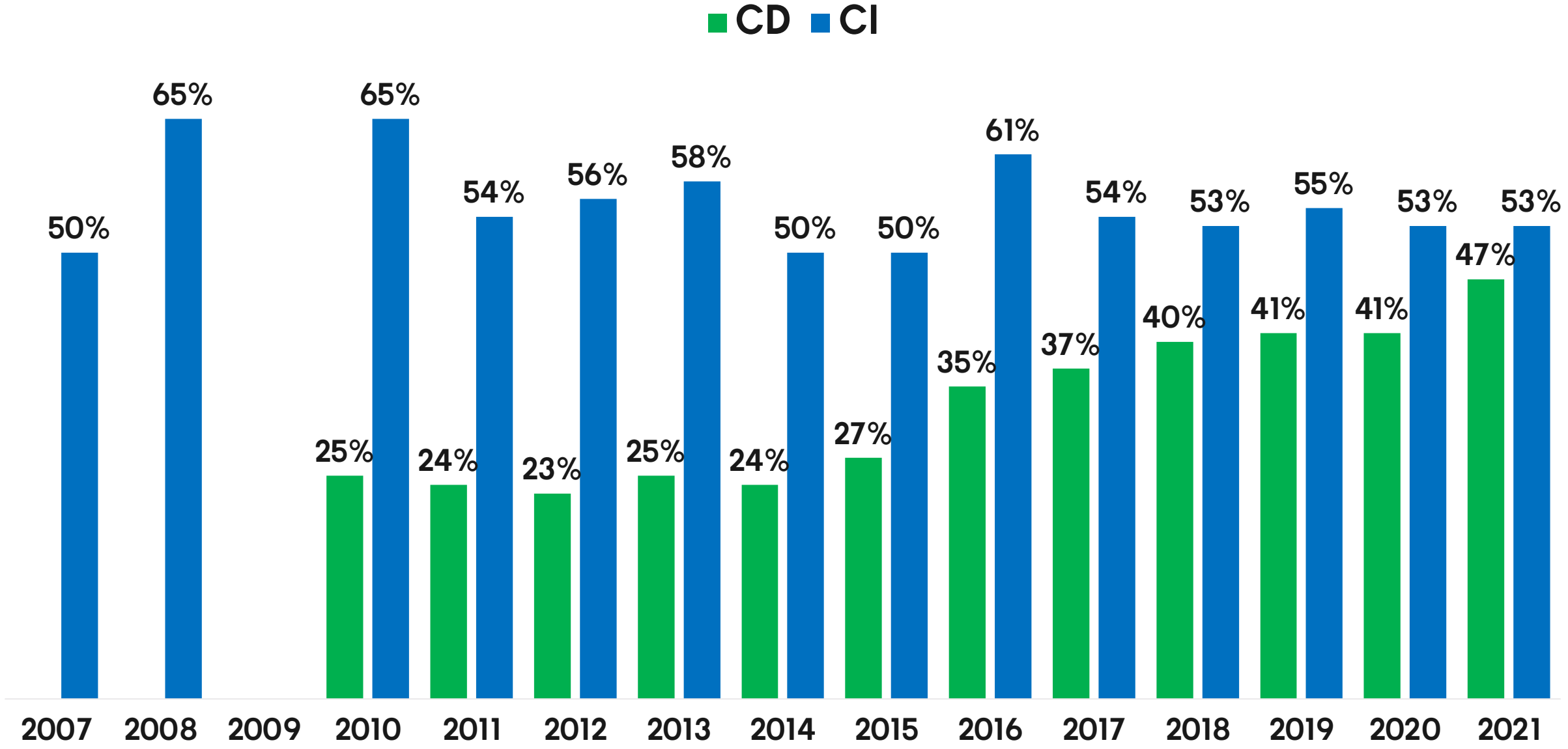


2016

"Printer firmware? Hold my beer."

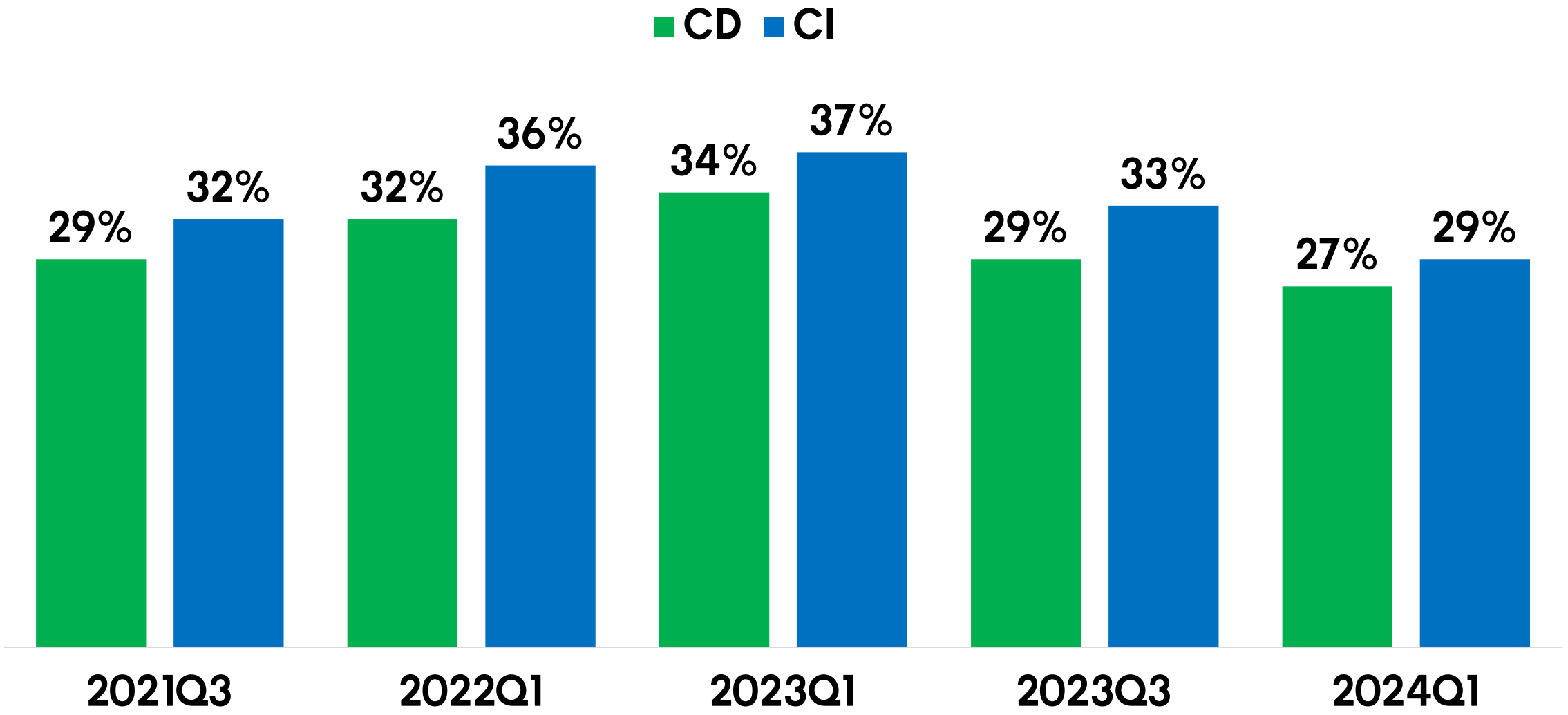
Note: originally attributed to Grady Booch in 1991, also XP principal in 1998. Sources: book listings, Sourceforge(!), Wikipedia on Sep 1st, 2023.

CI and CD usage, 2007 to 2021



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

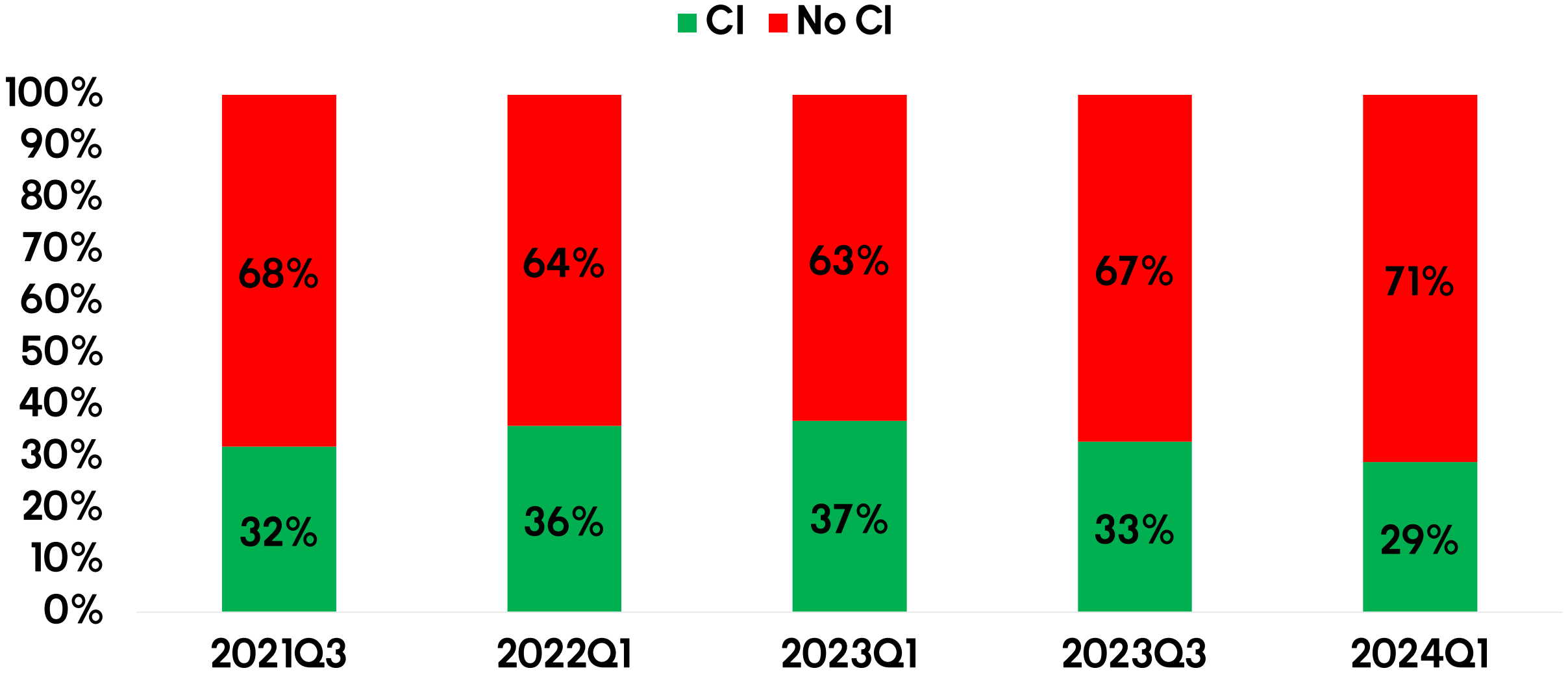
CI and CD Usage, 2021 to 2024



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.

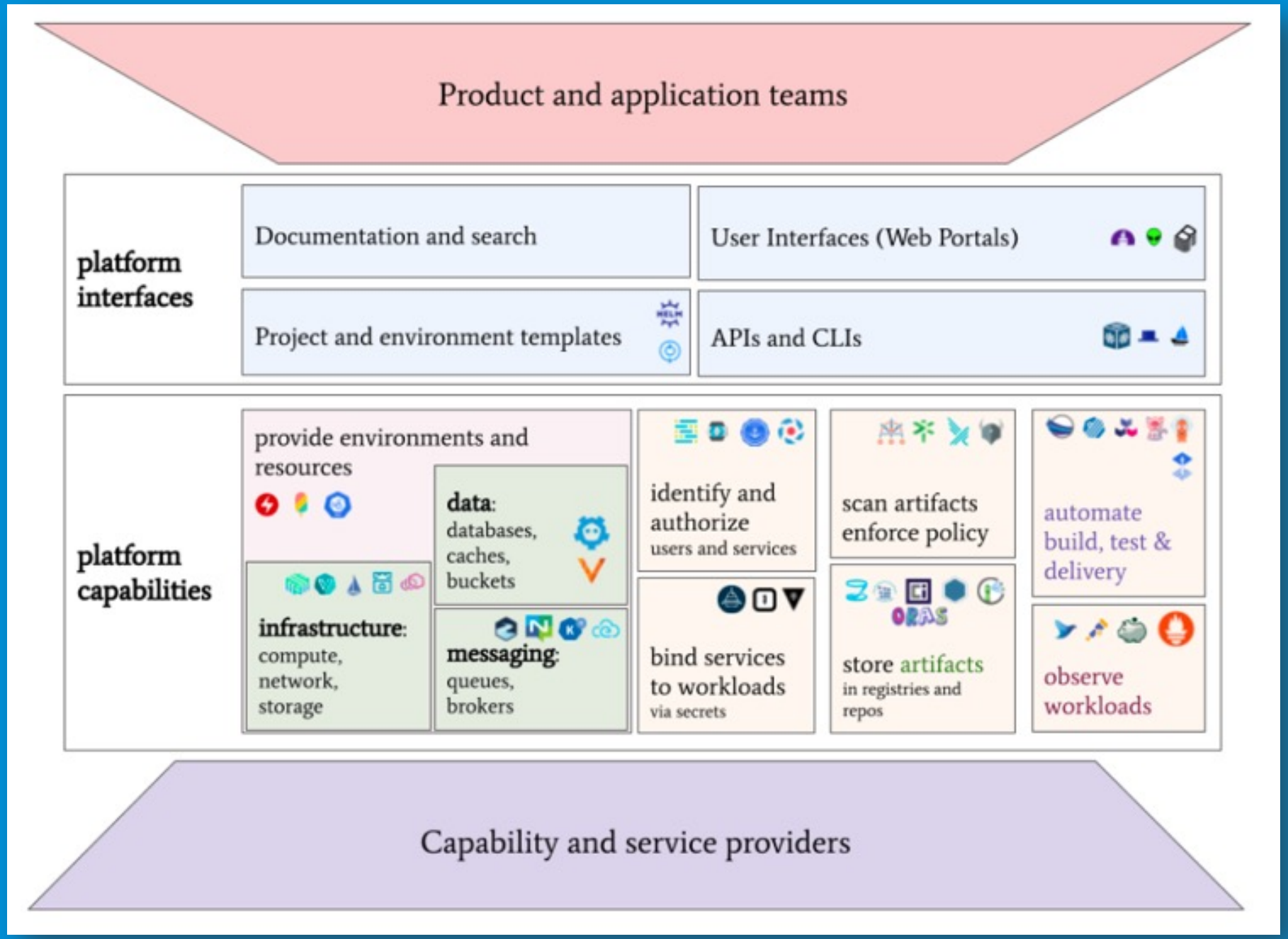
CI Usage, 2021 to 2024



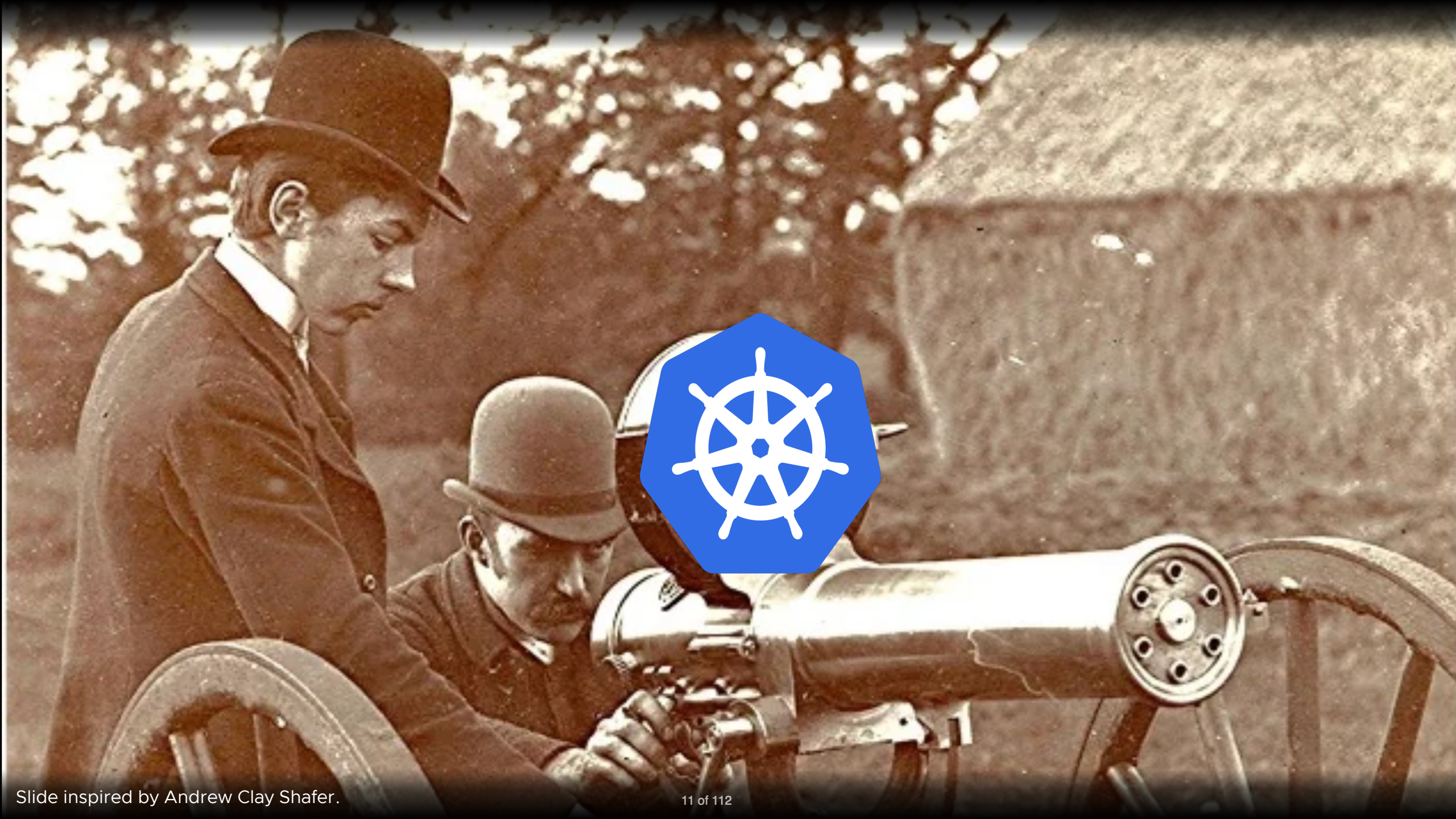
Question: Which of the following technologies have you used as part of your development activities in the last 12 months? Source: CD Foundation Surveys (Slashdata).

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

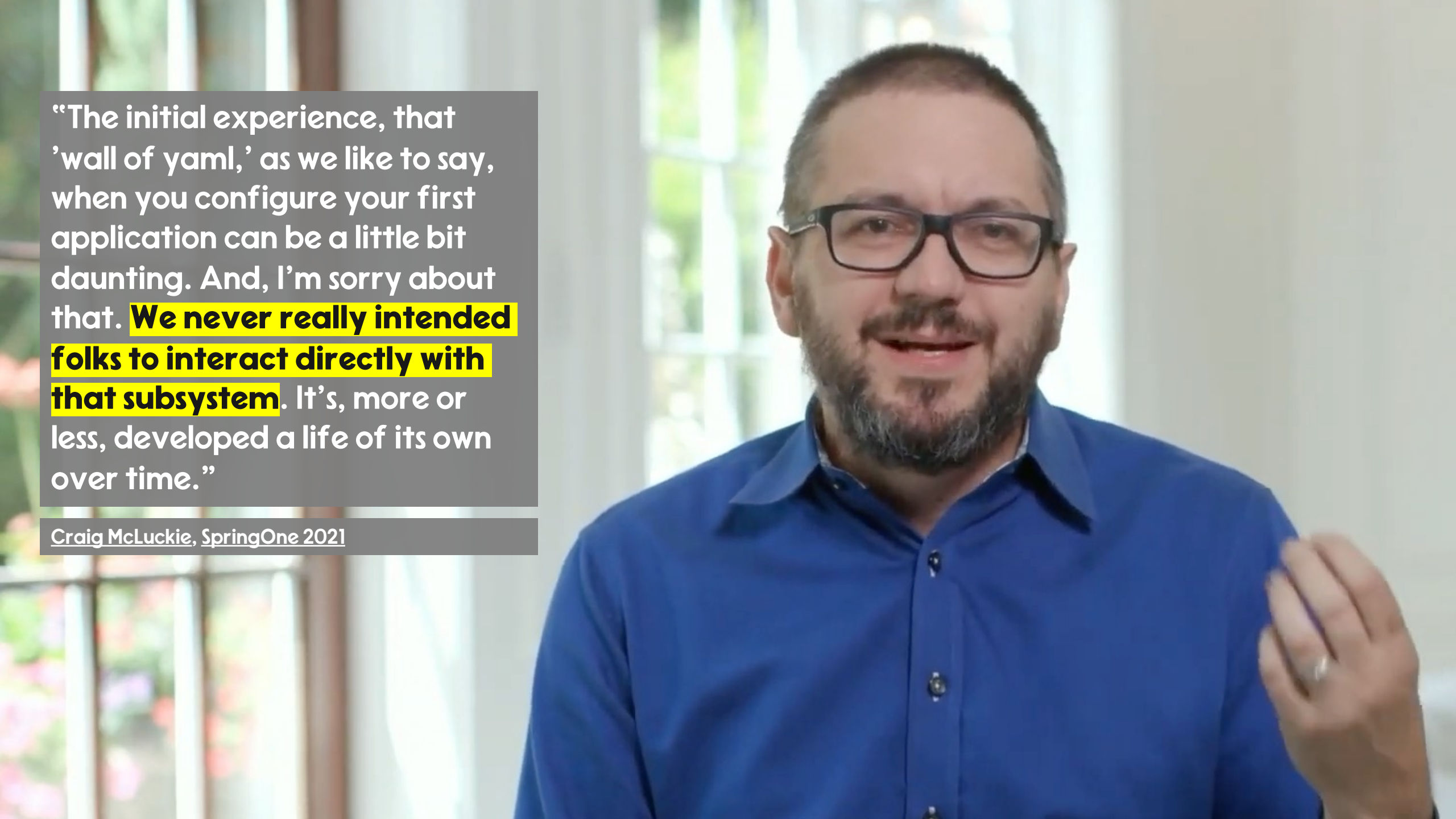
Stop building your own platforms, or, at least, have less of them



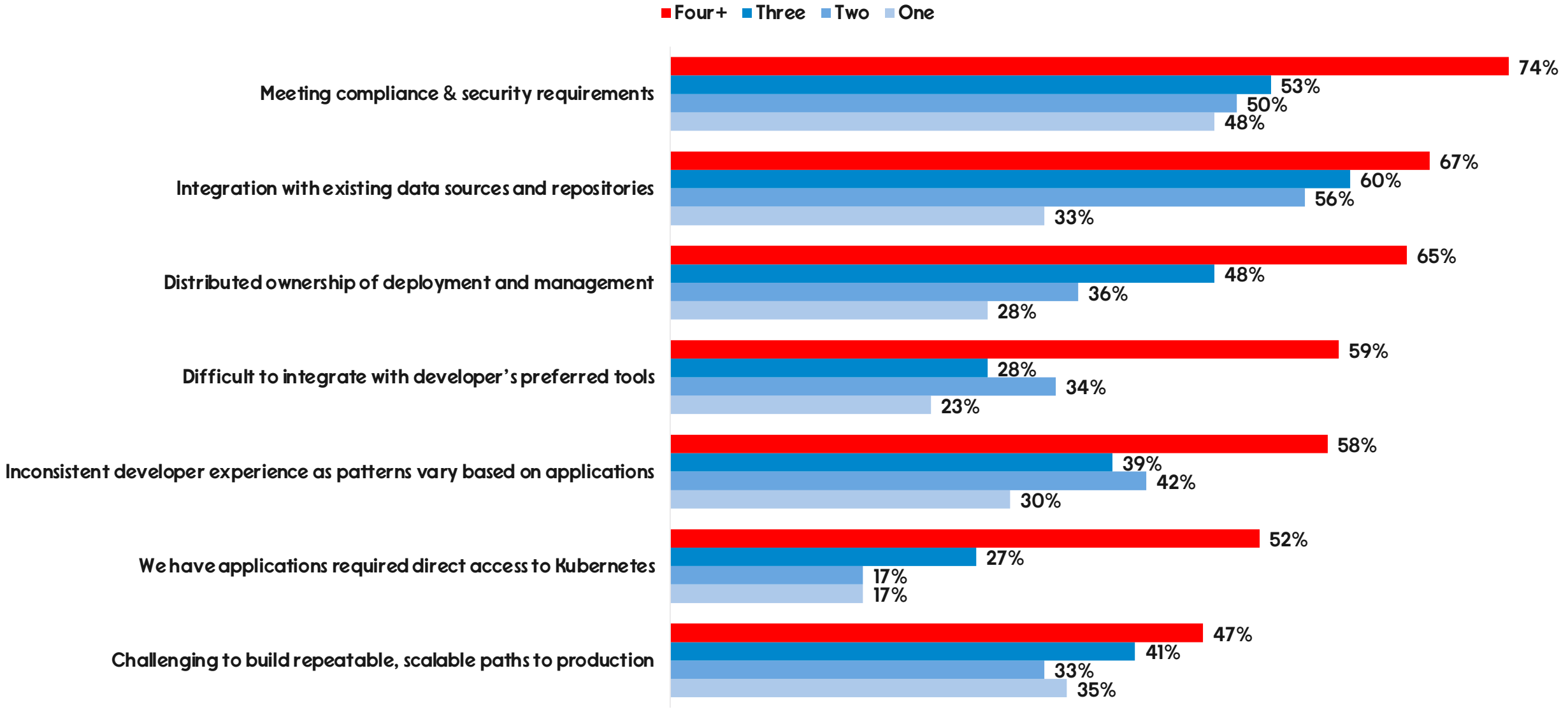
Source: "CNCF Platforms White Paper," March 2023.



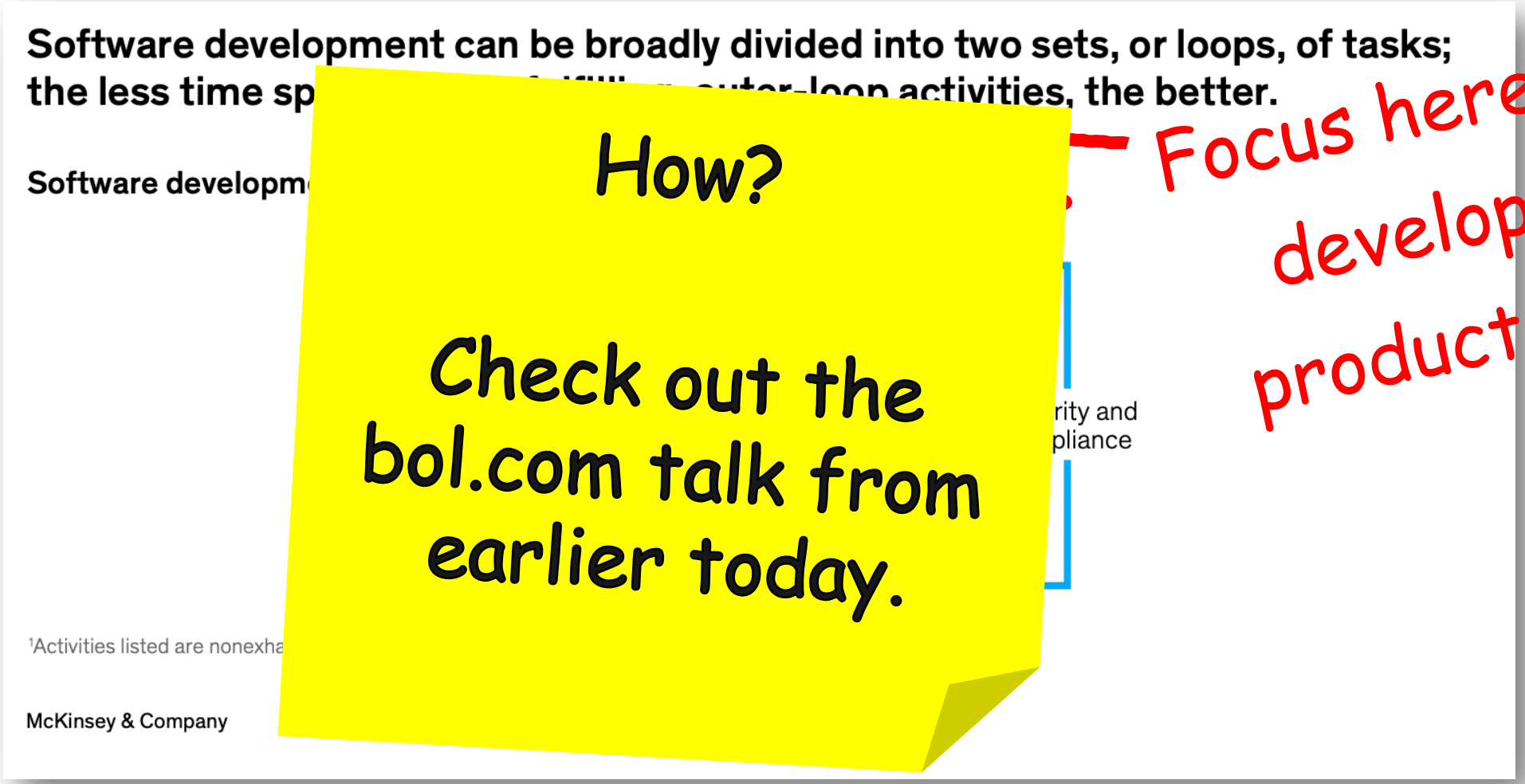
Slide inspired by Andrew Clay Shafer.


"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.



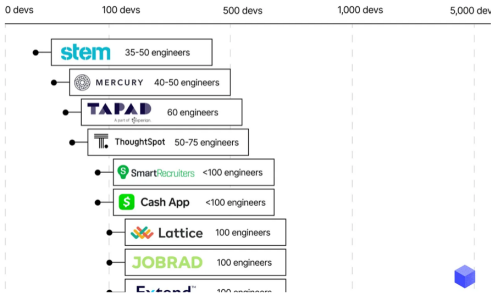
Focus here for developer productivity

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?



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

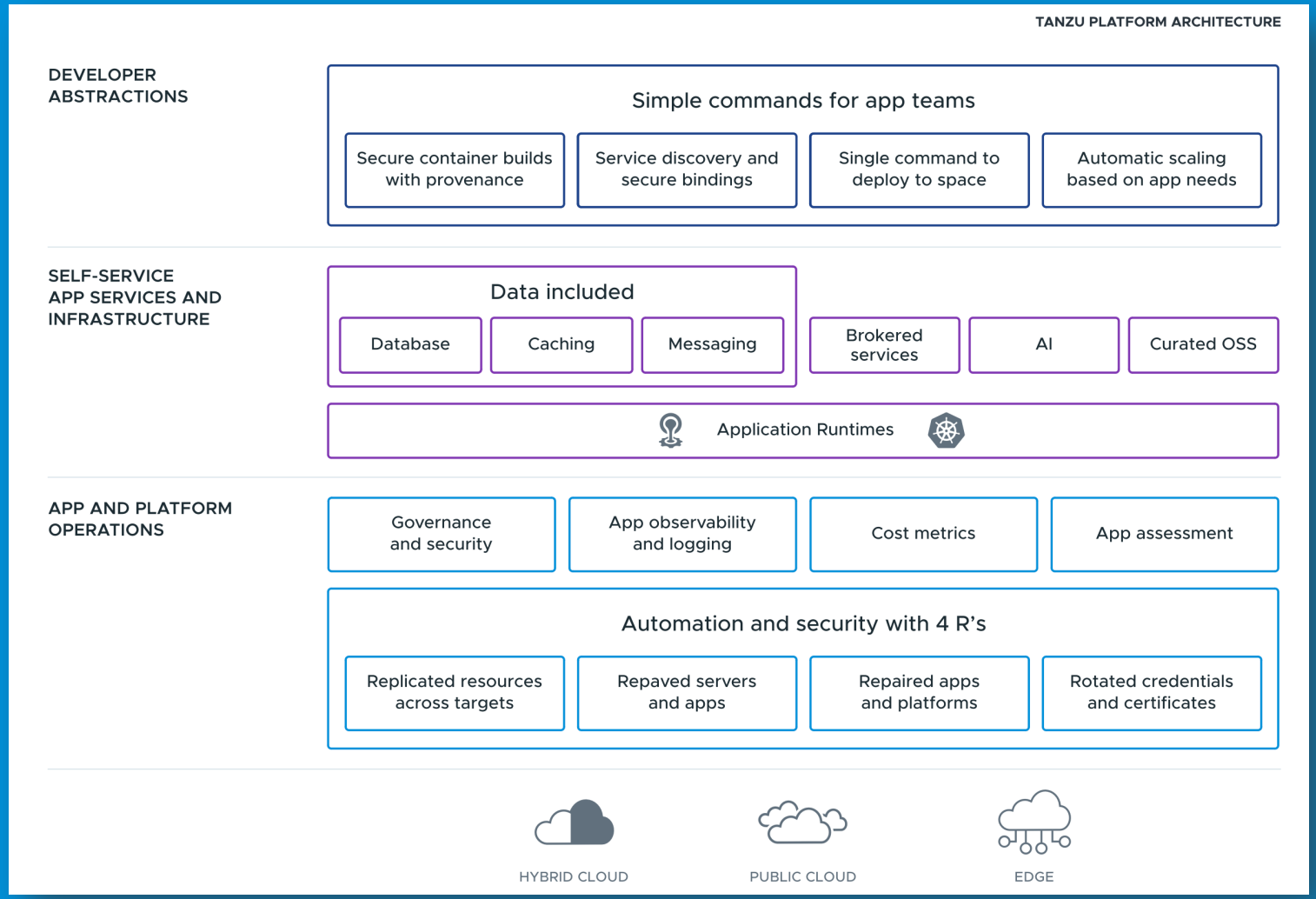
The latest research and perspectives on developer productivity.

Subscribed

Recommendations

The Pragmatic Engineer

Gergely Orosz



Thank You!

Slides !

https://tanzu.vmware.com/platform

https://newsletter.cote.io/