Developer Productivity is Waste

Preparing for the continuous deployment of DevX mania.

Coté – June 12th, 2024

Developer Productivity: The quick answers

Happiness, flow, features

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

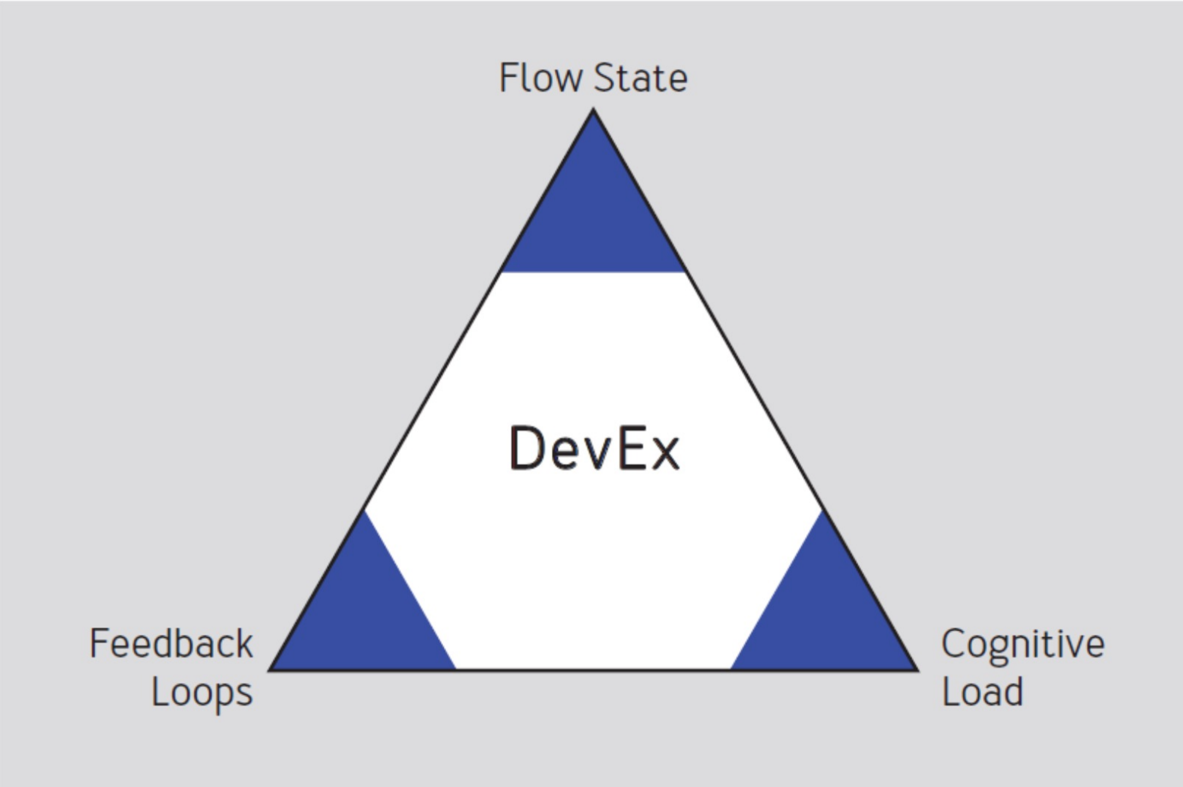
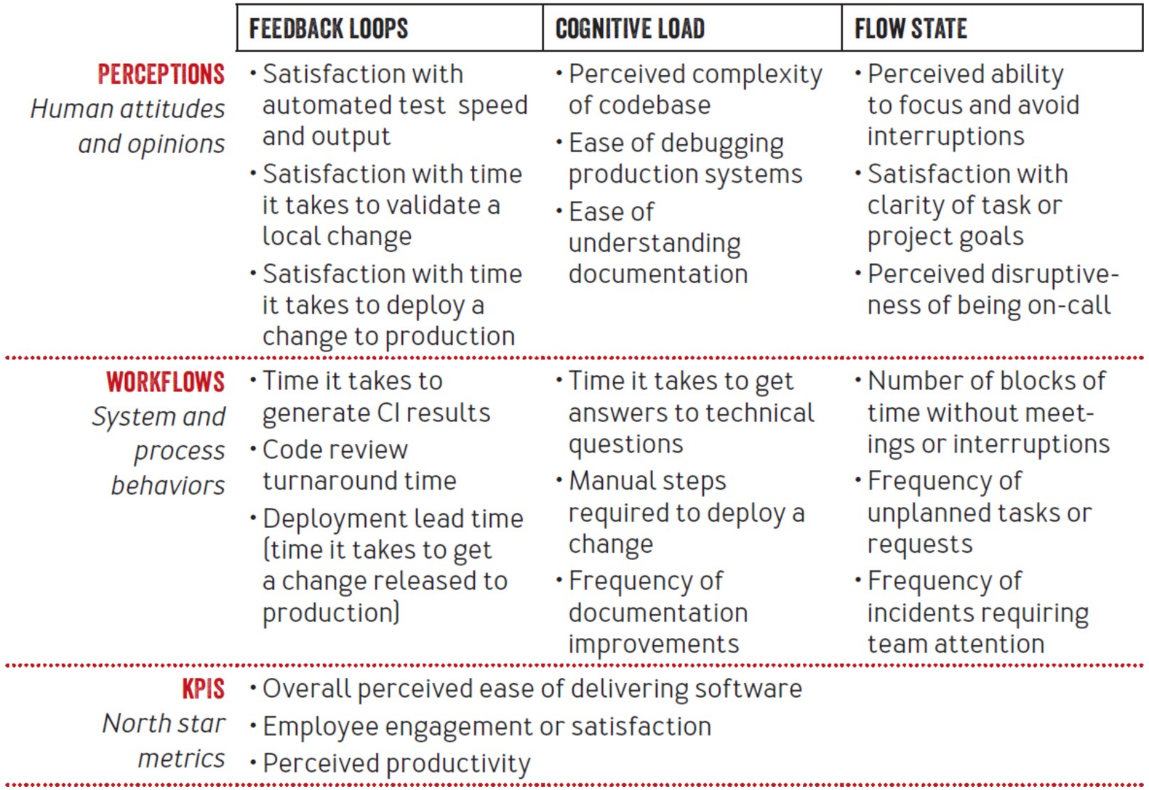
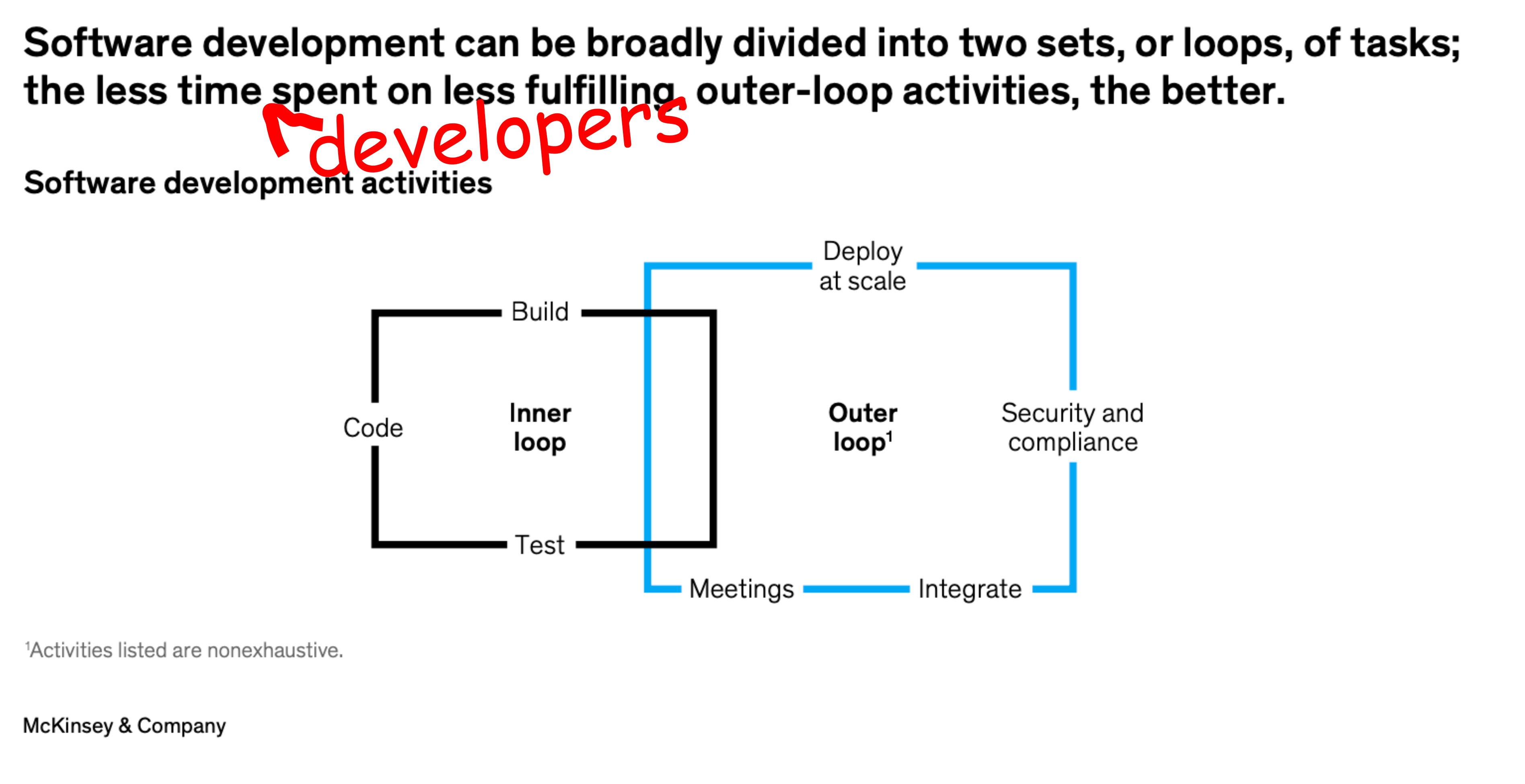


TABLE 1: **EXAMPLE DEVEX METRICS**



	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: "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.

developers

Software development activities

Build

Code

Inner loop

Test

Deploy at scale

Outer loop[1]

Security and compliance

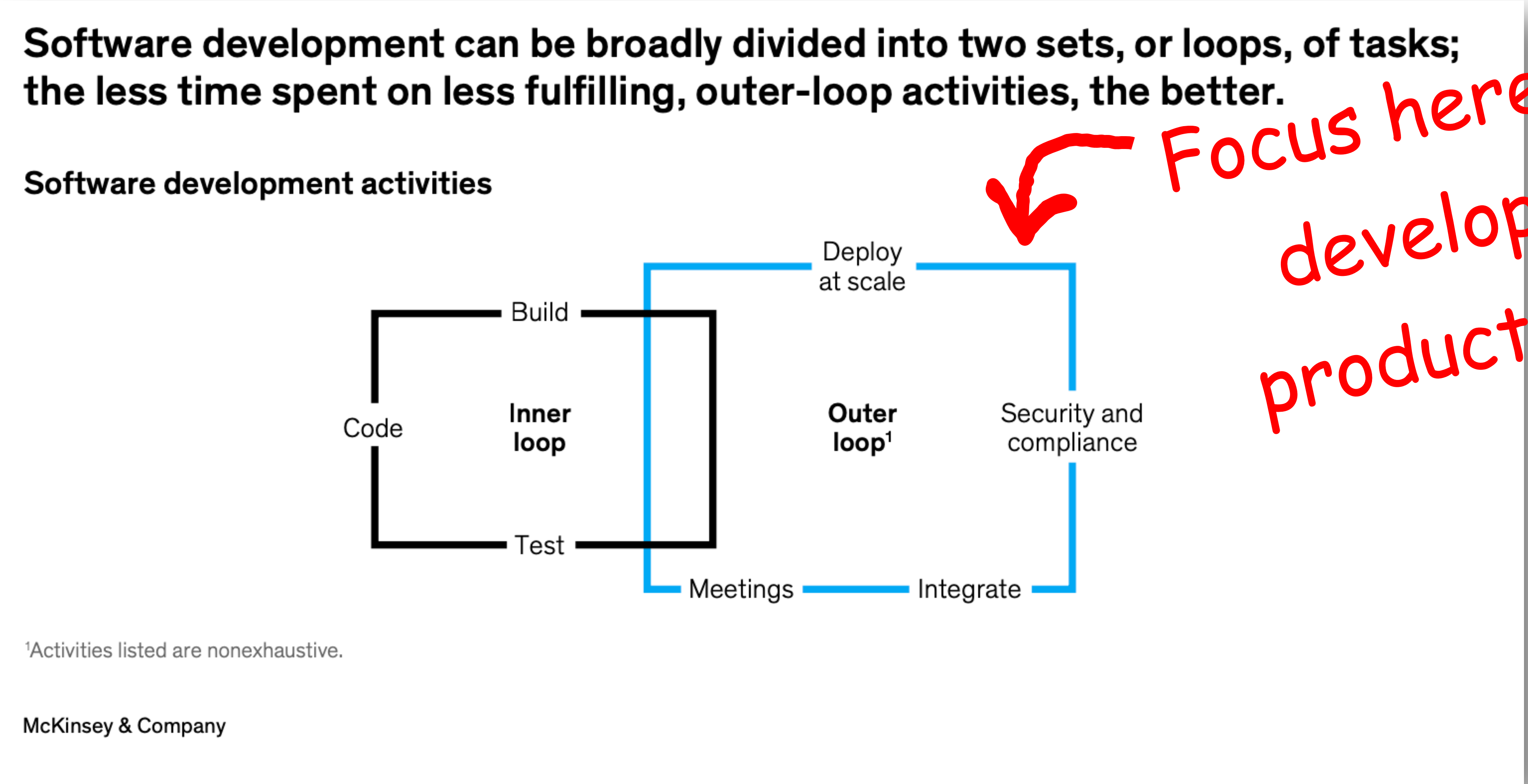
Meetings

Integrate

[1]Activities listed are nonexhaustive.

McKinsey & Company

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

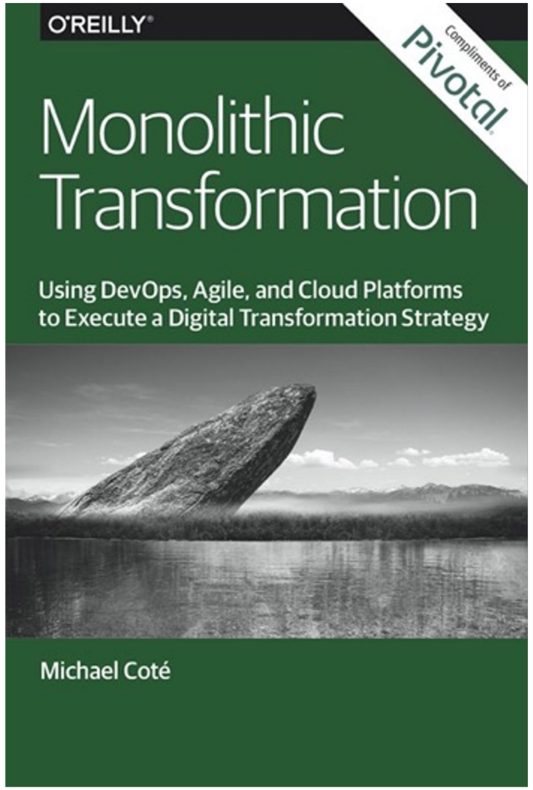
McKinsey & Company

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



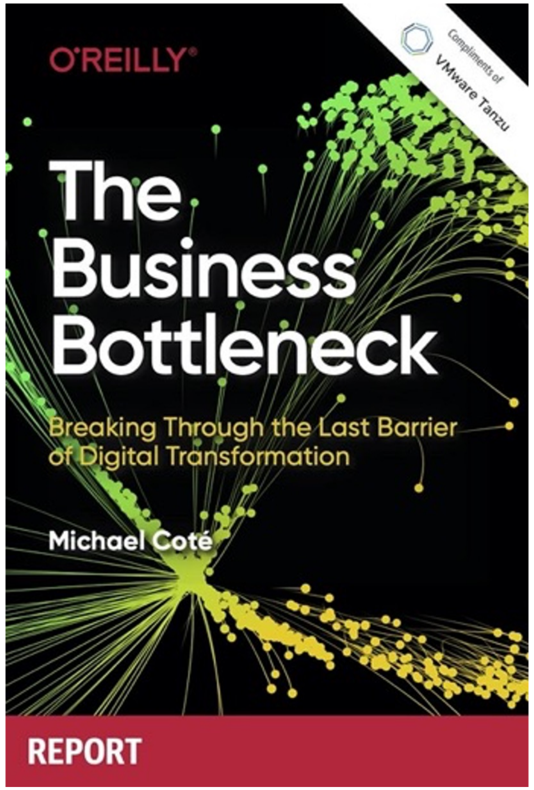
O'REILLY

Compliments of Pivotal

Monolithic Transformation

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

Michael Coté



O'REILLY

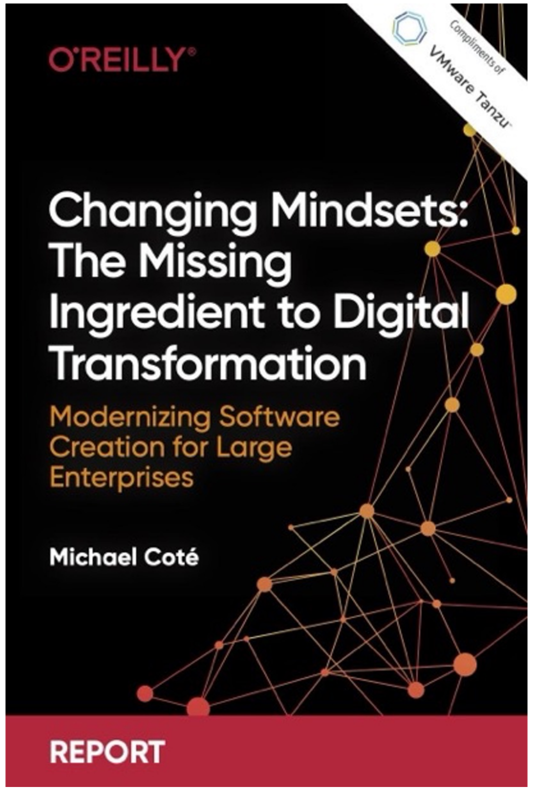
Compliments of VMware Tanzu

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

McKinsey & Company

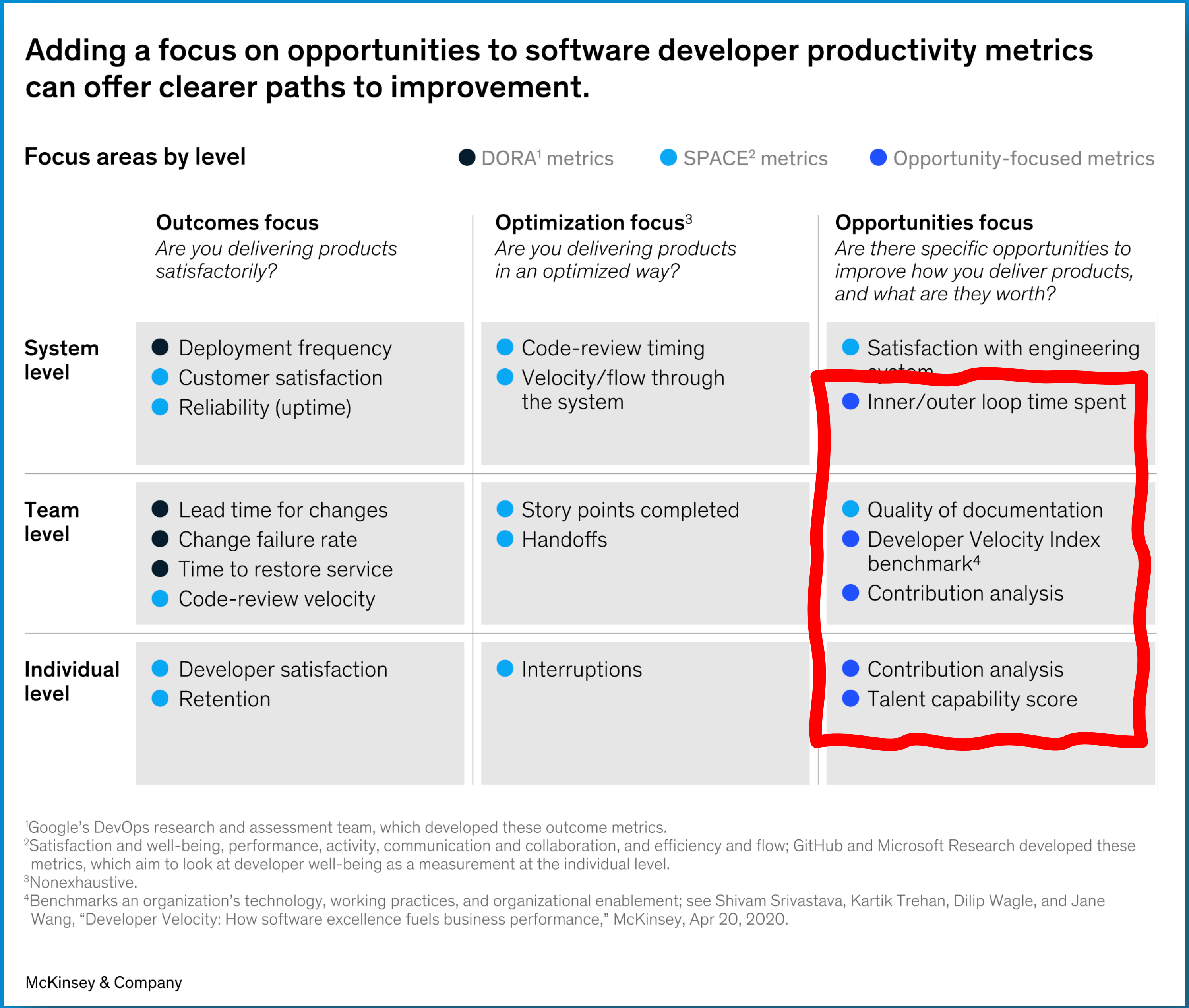
Technology, Media & Telecommunications Practice

Yes, you can measure software developer productivity

Measuring, tracking, and benchmarking developer productivity has long been considered a black box. It doesn't have to be that way.

This article is a collaborative effort by Chandra Gnanasambandam, Martin Harrysson, Alharith Hussin, Jason Keovichit, and Shivam Srivastava, representing views from McKinsey's Digital and Technology, Media & Telecommunications Practices.

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



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

Focus areas by level

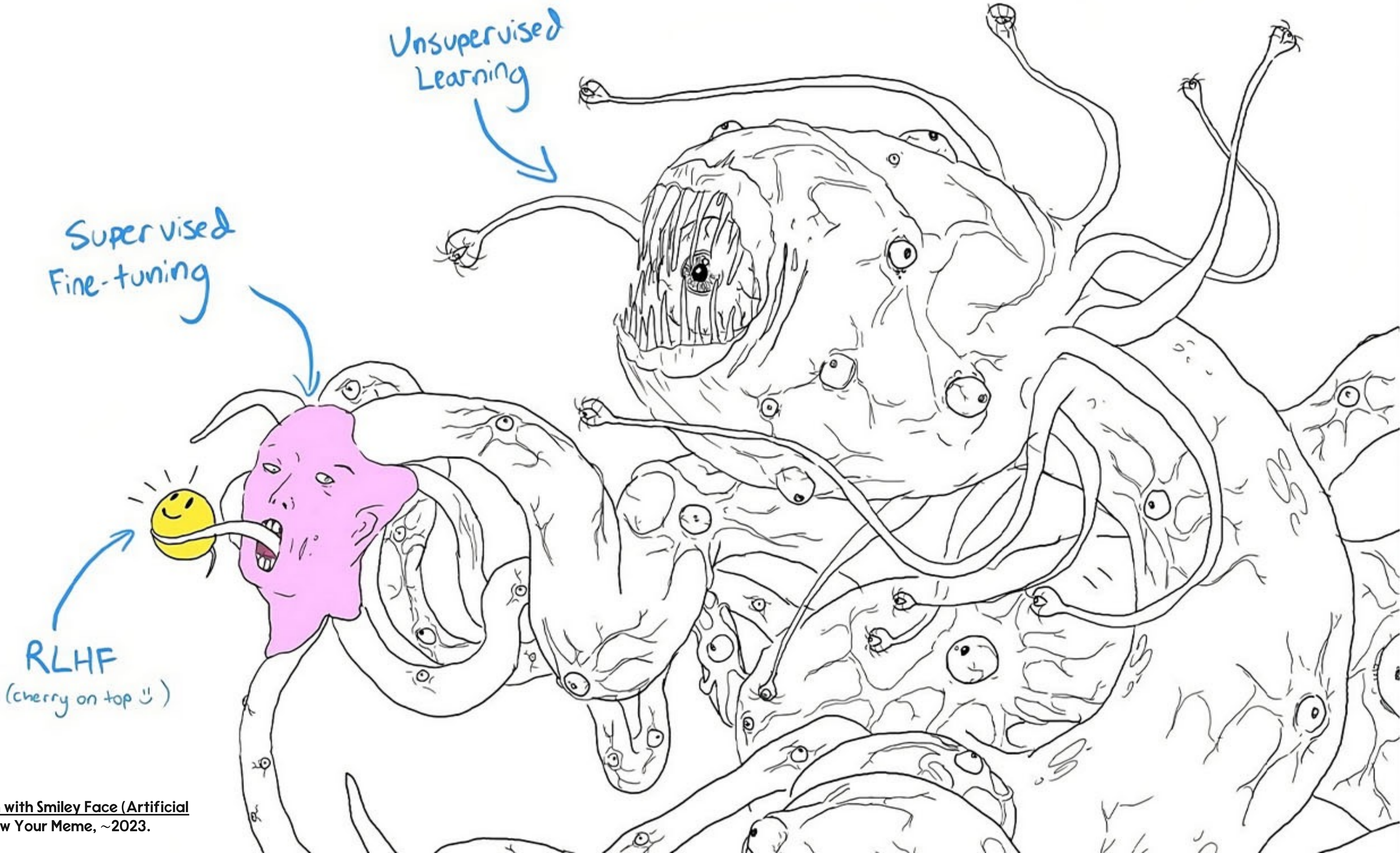
● DORA[1] metrics 🔵 SPACE[2] metrics 🔷 Opportunity-focused metrics

	Outcomes focus *Are you delivering products satisfactorily?*	**Optimization focus**[3] *Are you delivering products in an optimized way?*	**Opportunities focus** *Are there specific opportunities to improve how you deliver products, and what are they worth?*
System level	● Deployment frequency 🔵 Customer satisfaction 🔵 Reliability (uptime)	🔵 Code-review timing 🔵 Velocity/flow through the system	🔵 Satisfaction with engineering system 🔷 Inner/outer loop time spent
Team level	● Lead time for changes ● Change failure rate ● Time to restore service 🔵 Code-review velocity	🔵 Story points completed 🔵 Handoffs	🔵 Quality of documentation 🔷 Developer Velocity Index benchmark[4] 🔷 Contribution analysis
Individual level	🔵 Developer satisfaction 🔵 Retention	🔵 Interruptions	🔷 Contribution analysis 🔷 Talent capability score

[1] Google's DevOps research and assessment team, which developed these outcome metrics.
[2] Satisfaction 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.
[3] Nonexhaustive.
[4] Benchmarks 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

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.



Source: “Shoggoth with Smiley Face (Artificial Intelligence),” Know Your Meme, ~2023.

26% of organizations are using GenAI to support application development, testing, and management and 25% are utilizing machine learning with the greatest focus on leveraging AI to support DevOps analytics and process, governance, and security testing.

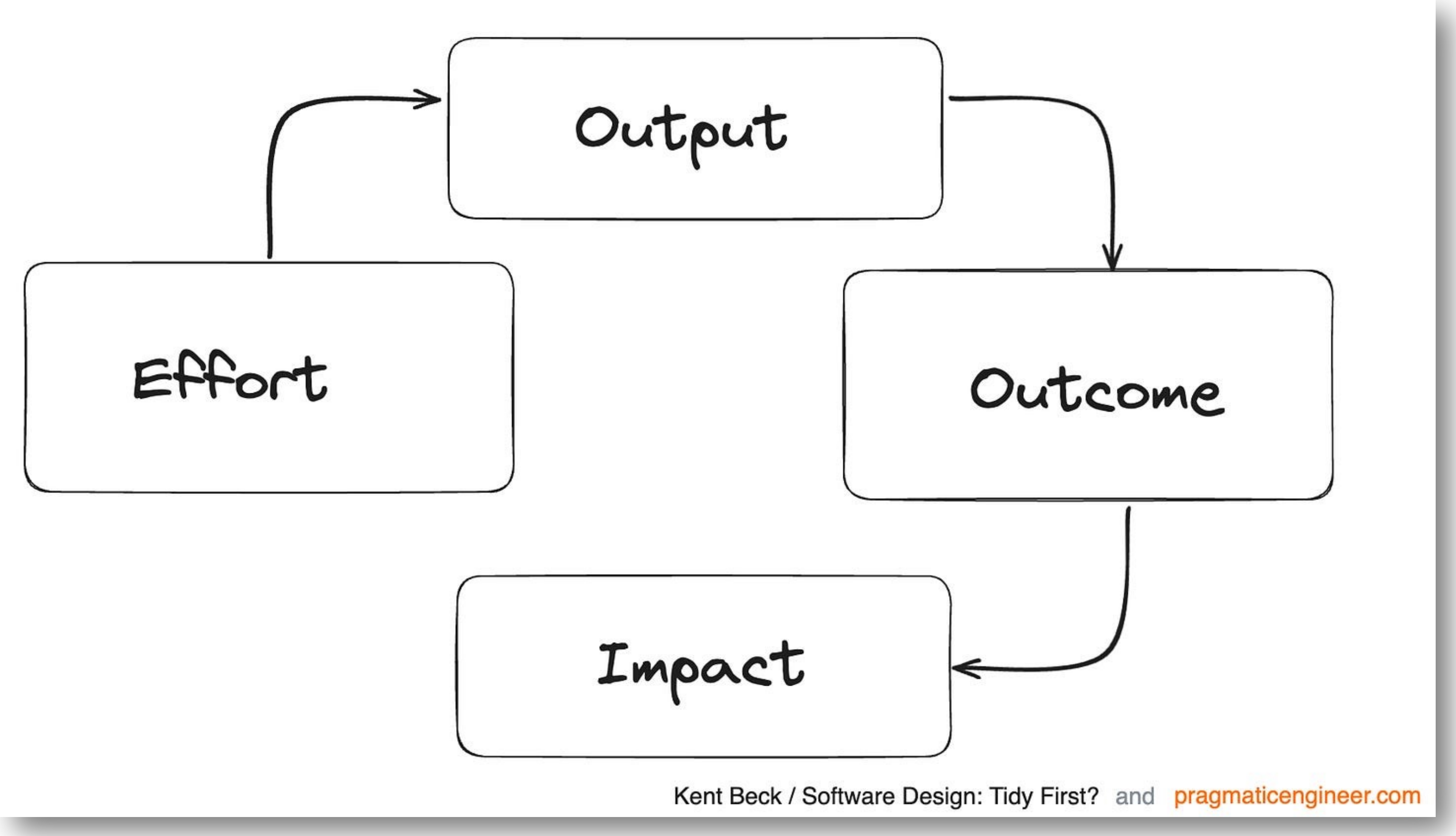
What is “developer productivity”?

The Problem

"Engineering leaders have long sought to improve the productivity of their developers, but knowing how to measure or even define developer productivity has remained elusive."

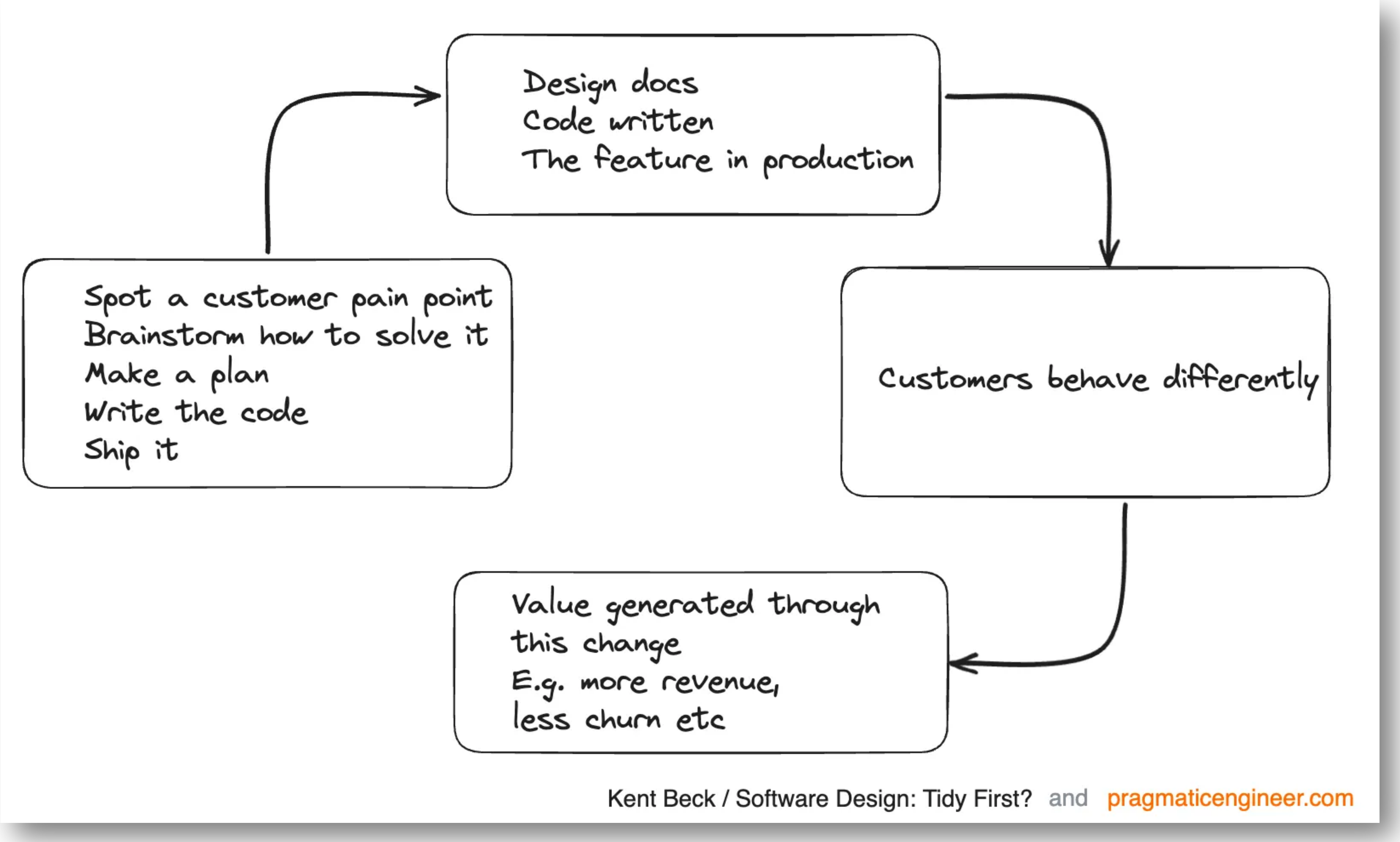
"there is no clear-cut definition of productivity and its influencing factors, although research has been conducted for more than a century. Like in software engineering, this lack of common agreement on what actually constitutes productivity, is perceived as a major obstacle for a substantiated discussion of productivity."

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.



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

Design docs
Code written
The feature in production

Customers behave differently

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

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

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

Why?

. . .and who’s asking?

Improvement.

Developers getting better at craft.

It's usually this person asking

Sources: Know Your Meme.

Money.

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

Business Growth.

Adding more developers

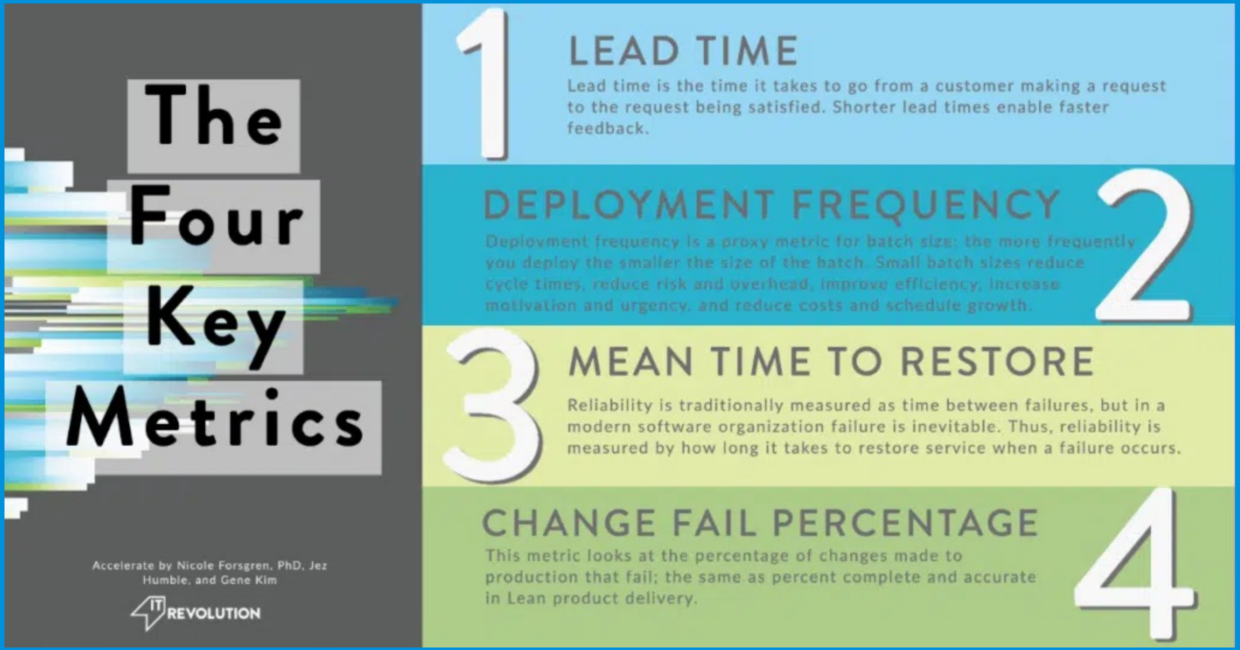
vs.

Increasing productivity per developer

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

At the Metrics Buffett

DORA



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.

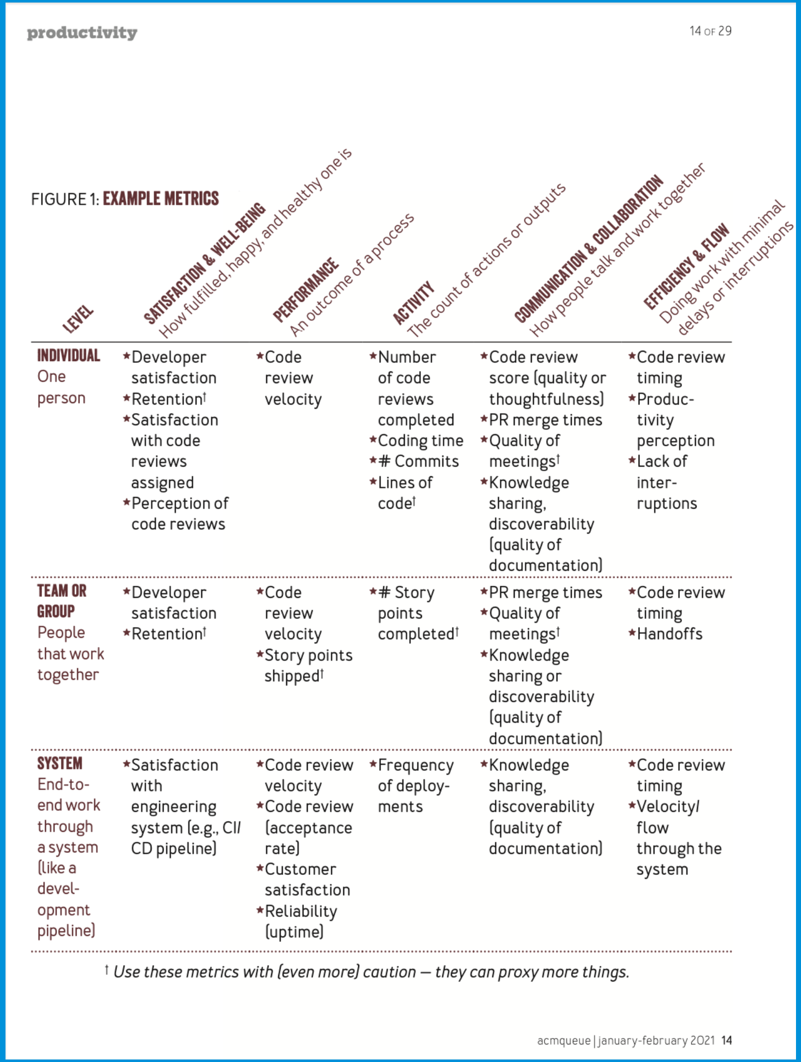
3 MEAN TIME TO RESTORE

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.

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 *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, features

Or, “stop interrupting me!”

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

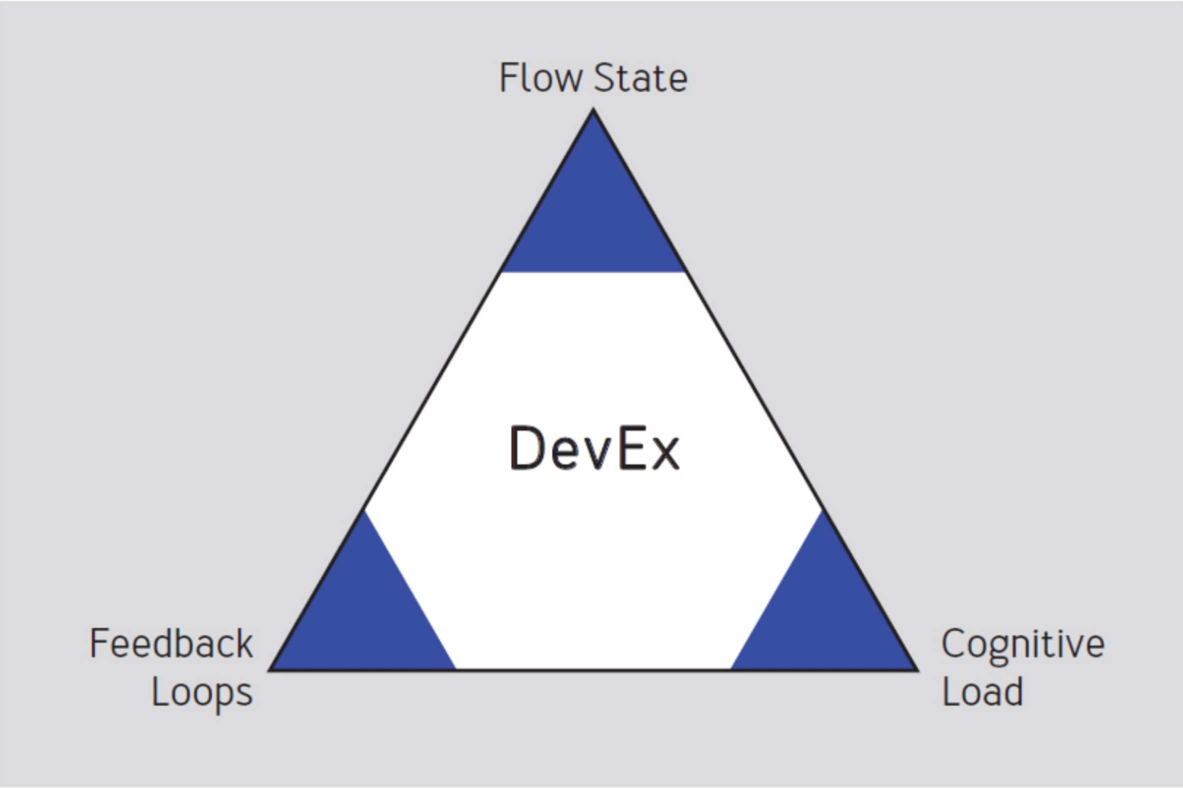
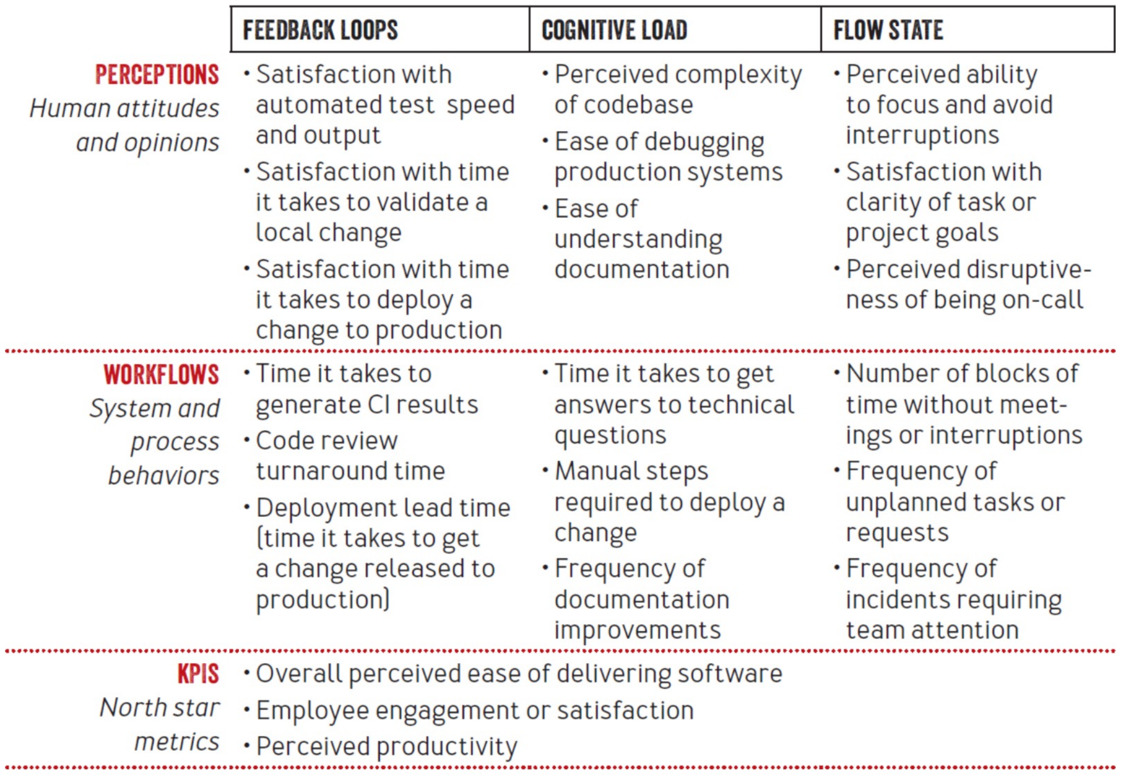


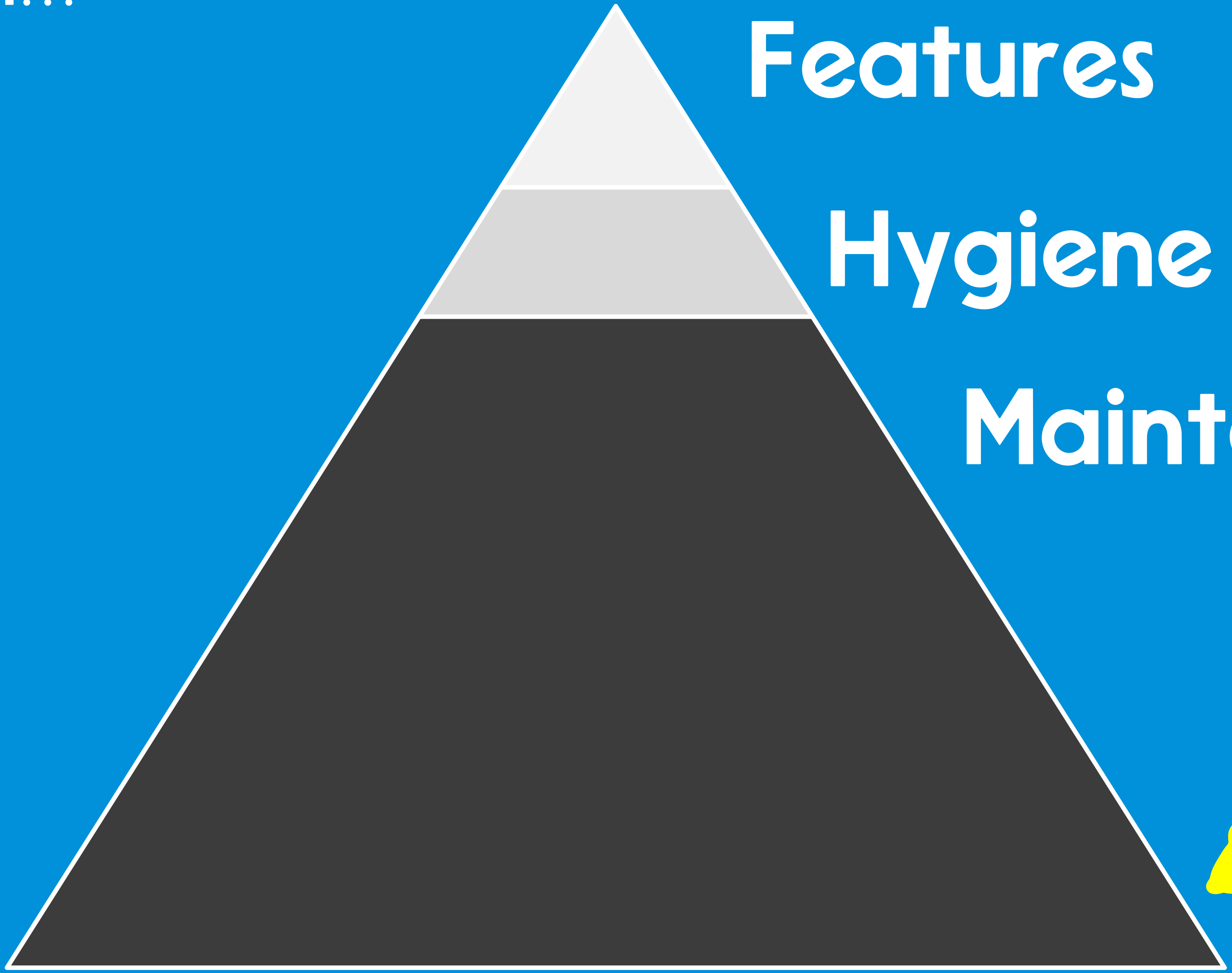
TABLE 1: **EXAMPLE DEVEX METRICS**



	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: “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.

Time spent on…



Features

Hygiene

Maintenance

Not good.

Source: “A Useful Productivity Measure?” James Shore, May 2024.

Developer Productivity Tools

Find the Developer Toil, Confusion, Blockers

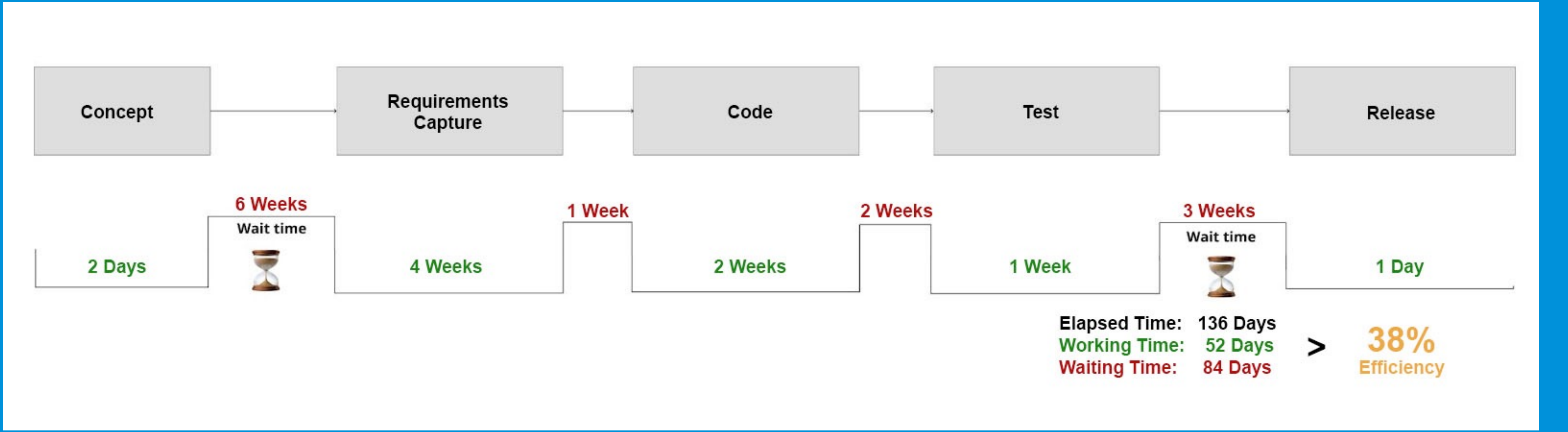
Find the Developer Toil, Confusion, Blockers

- 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.

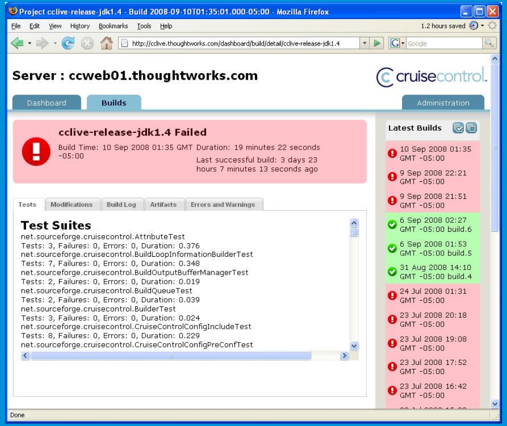
Sources: "Developer Toil: The Hidden Tech Debt," Susie Forbath, Tyson McNulty, and Coté, August, 2022. See also Michael Galloway's interview questions for platform product managers.

Put CI/CD in place

Waste is outside the box



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



Project cclive-release-jdk1.4 - Build 2008-09-10T01:35:01.000-05:00 - Mozilla Firefox

http://cclive.thoughtworks.com/dashboard/build/detail/cclive-release-jdk1.4

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

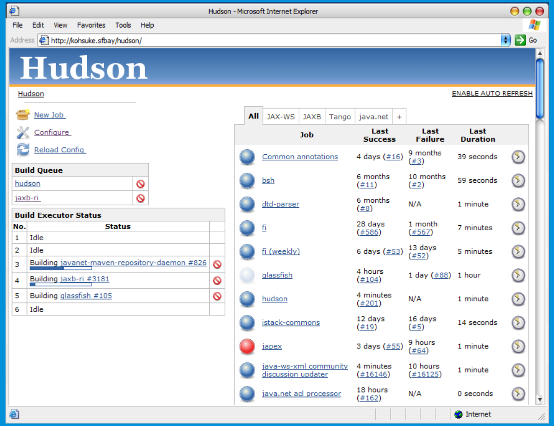
Test Suites

```
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

http://kohsuke.sfbay/hudson/

Build Queue

- hudson
- jaxb-ri

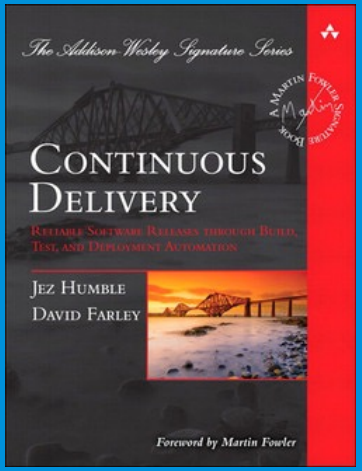
Build Executor Status

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

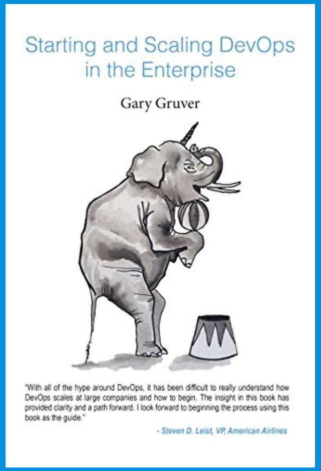
CONTINUOUS DELIVERY

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

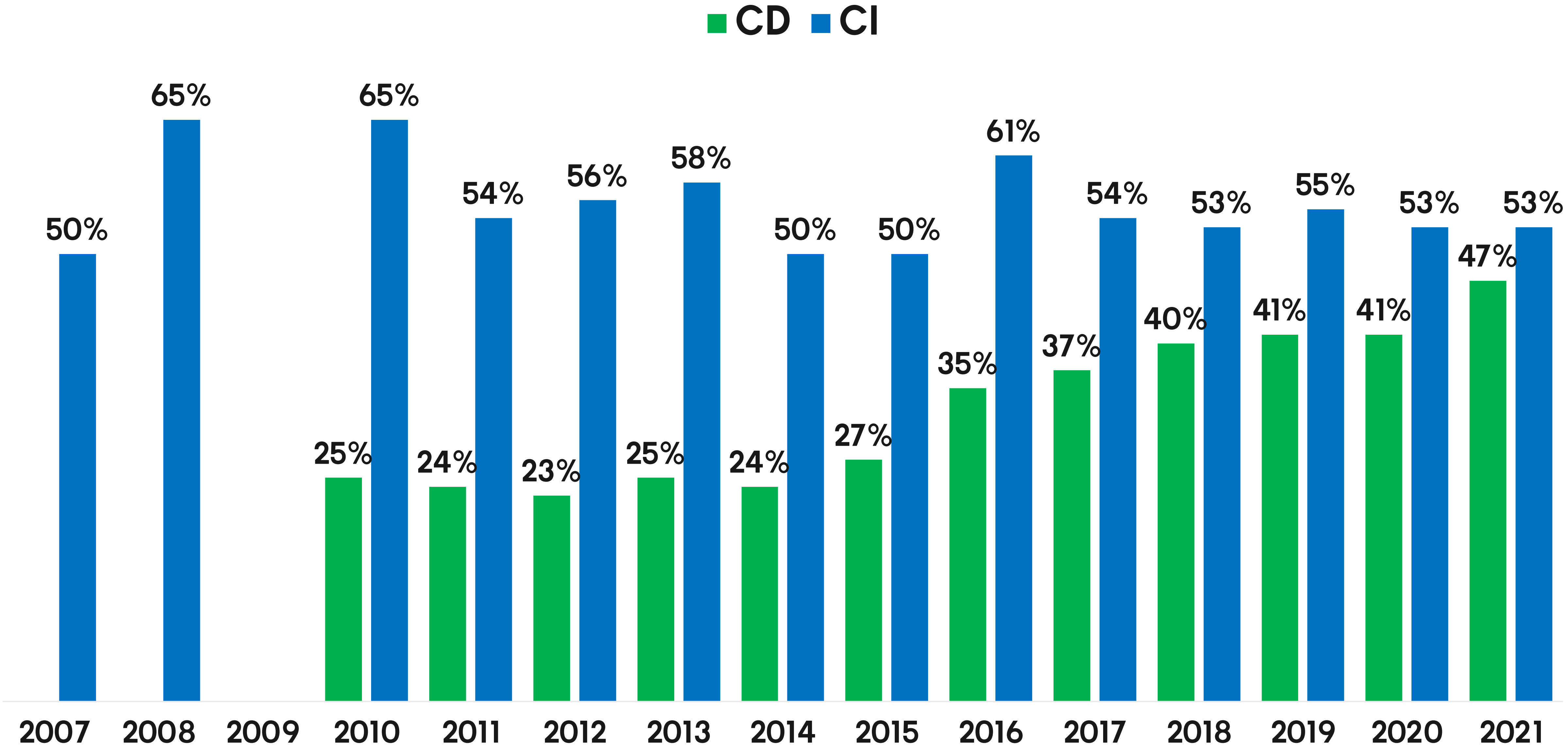
2016

Printer firmware?

Hold my beer.

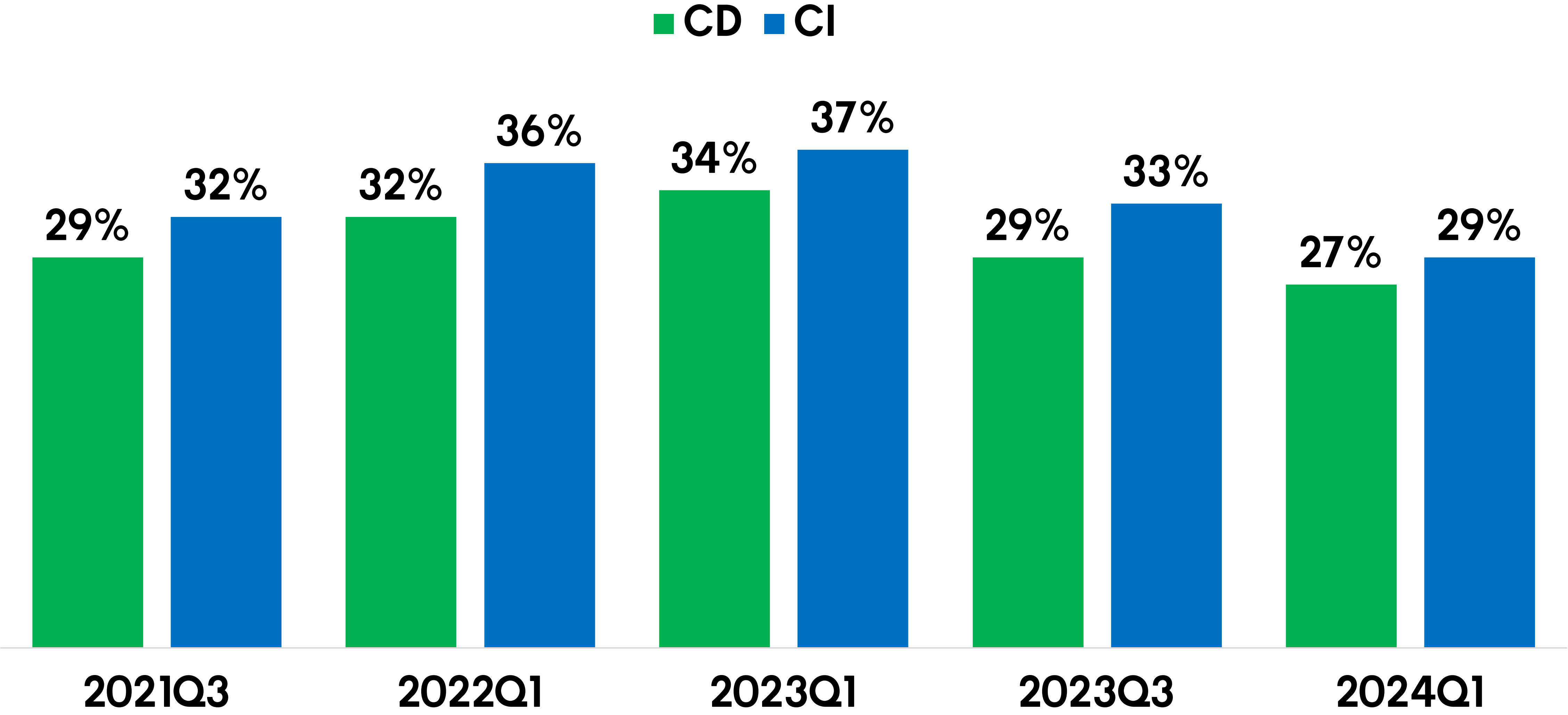
Note: originally attributed to Grady Booch in 1991, also XP principal in 1998. Sources: book listings, Sourceforge(!), Wikipedia on Sep 1st, 2023.

CI and CD usage, 2007 to 2021



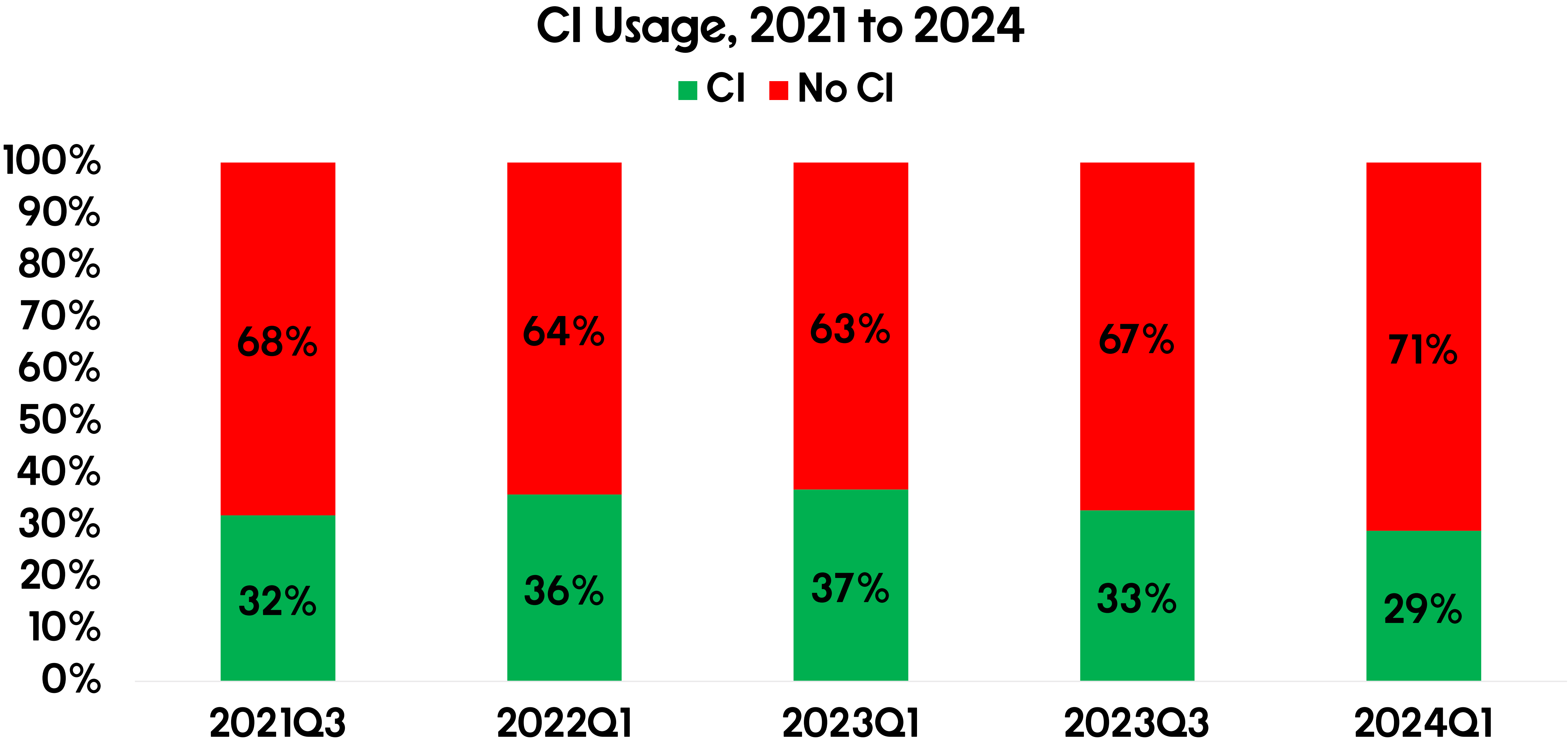
Sources: State of Agile Surveys, VersionOne/CollabNet/digital.ai.

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).

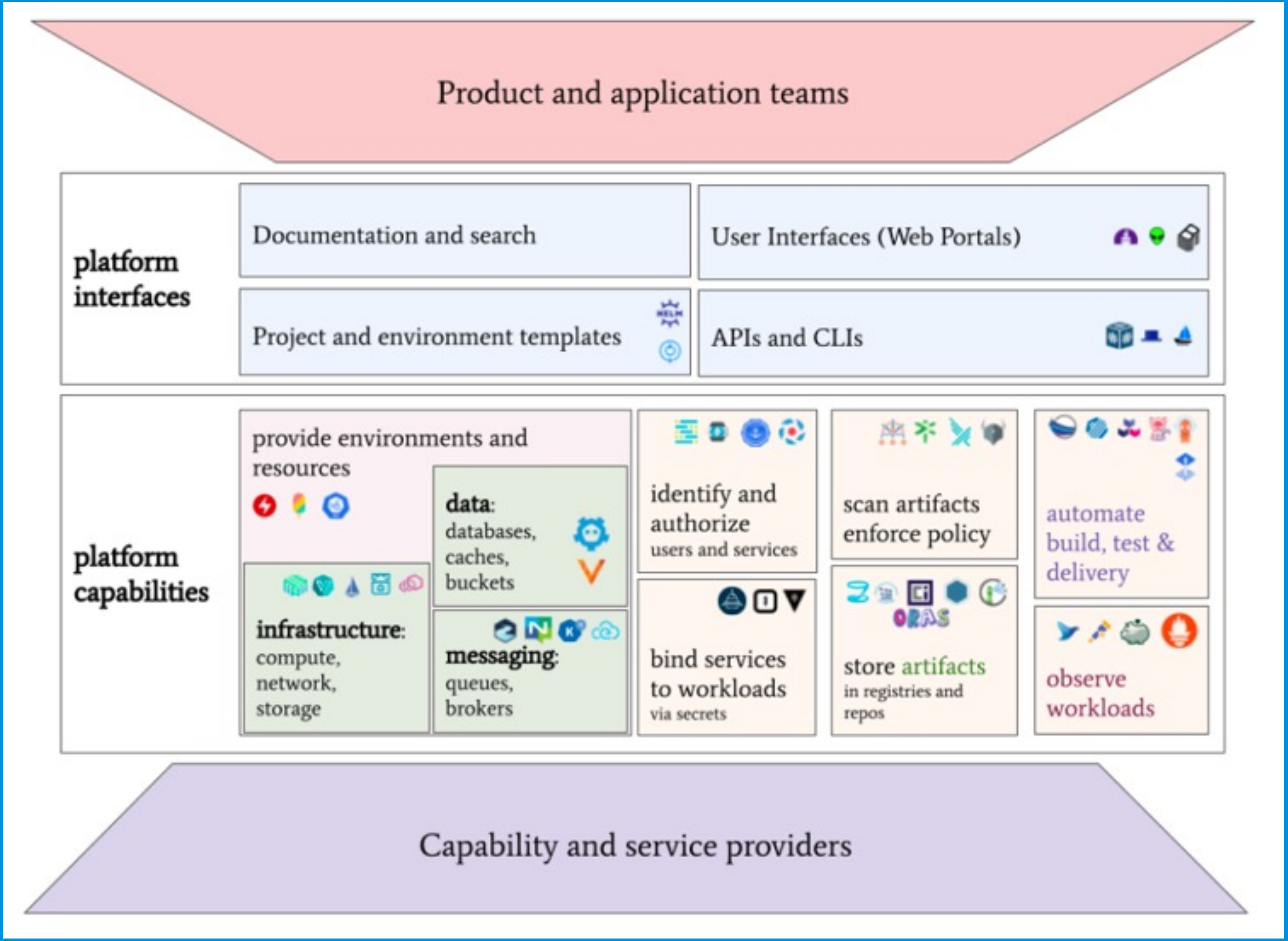
Source: "State of CI/CD Report 2024: The Evolution of Software Delivery Performance," CD Foundation and SlashData, April, 2024, and previous studies.



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, etc.


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

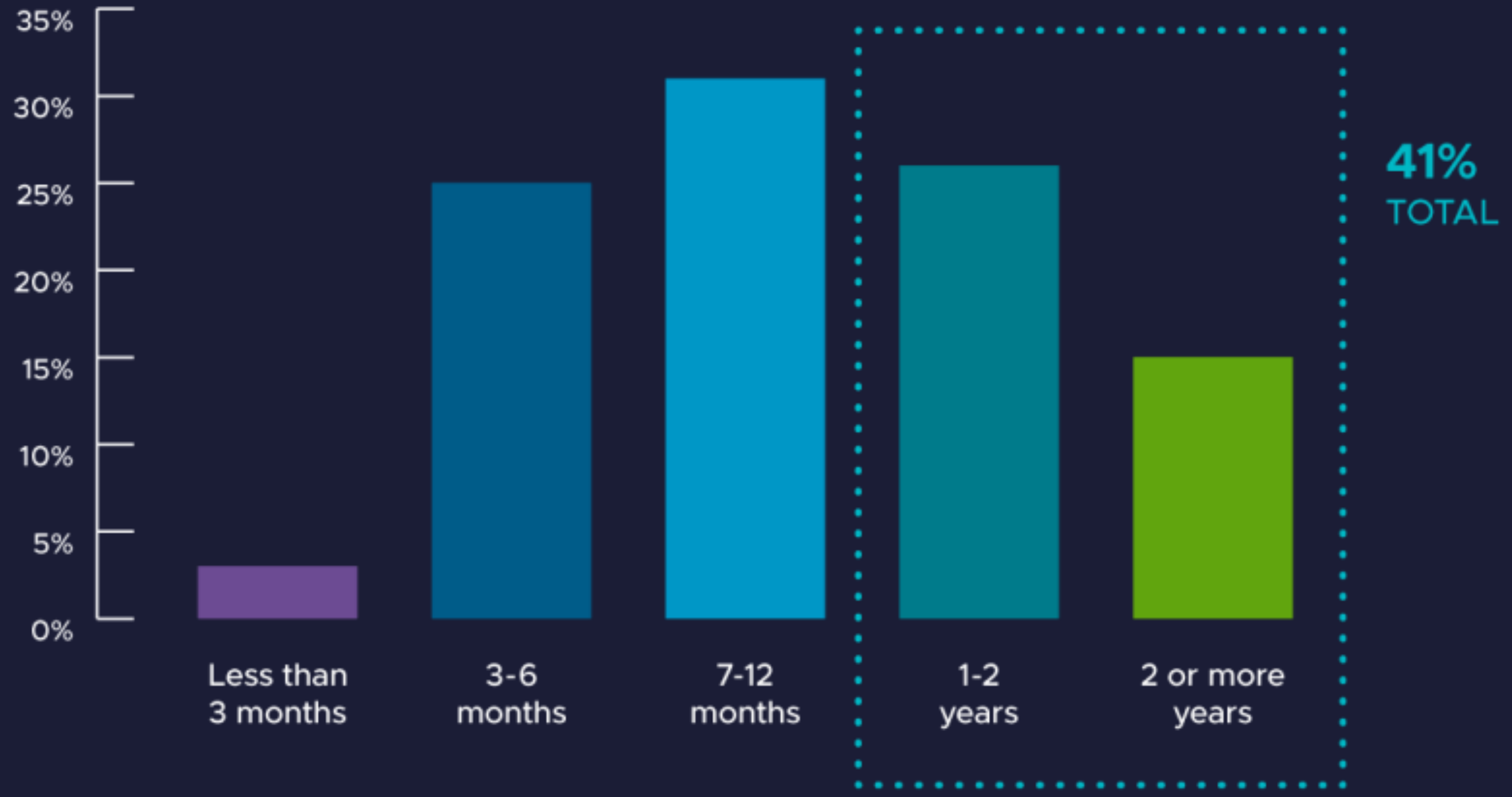
automate build, test & delivery

observe workloads

Capability and service providers

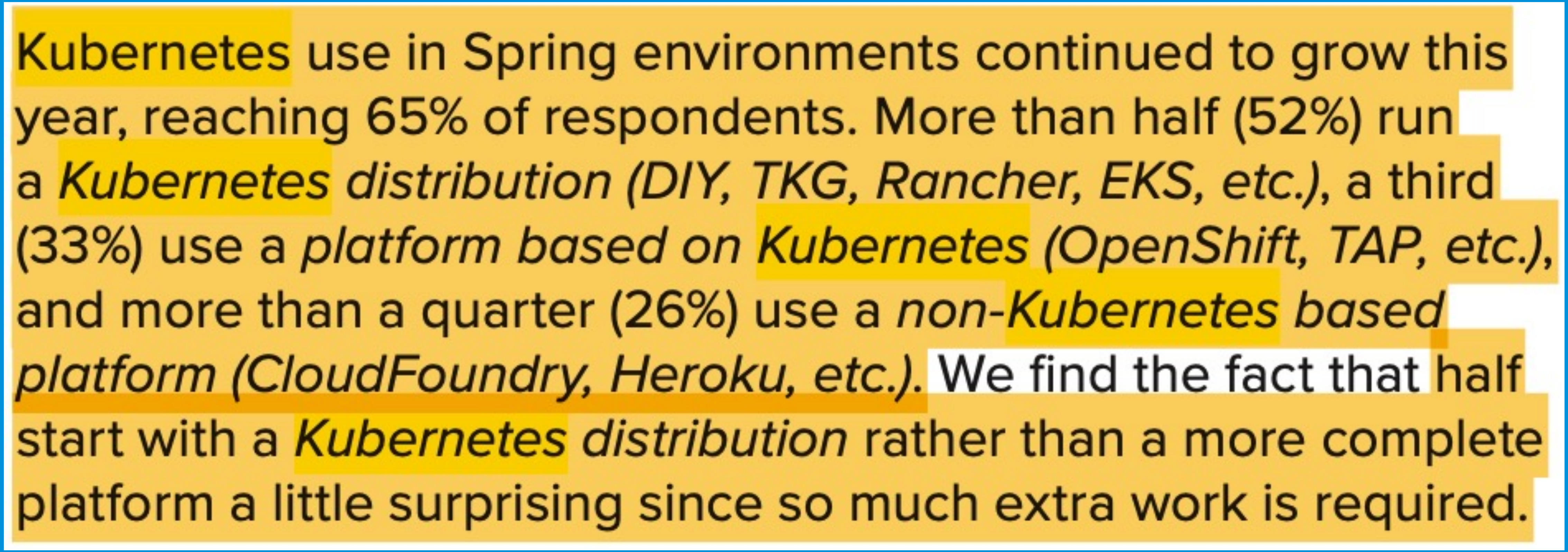
Source: "CNCF Platforms White Paper," March 2023.

Building a custom platform takes **41%** of organizations more than a year.



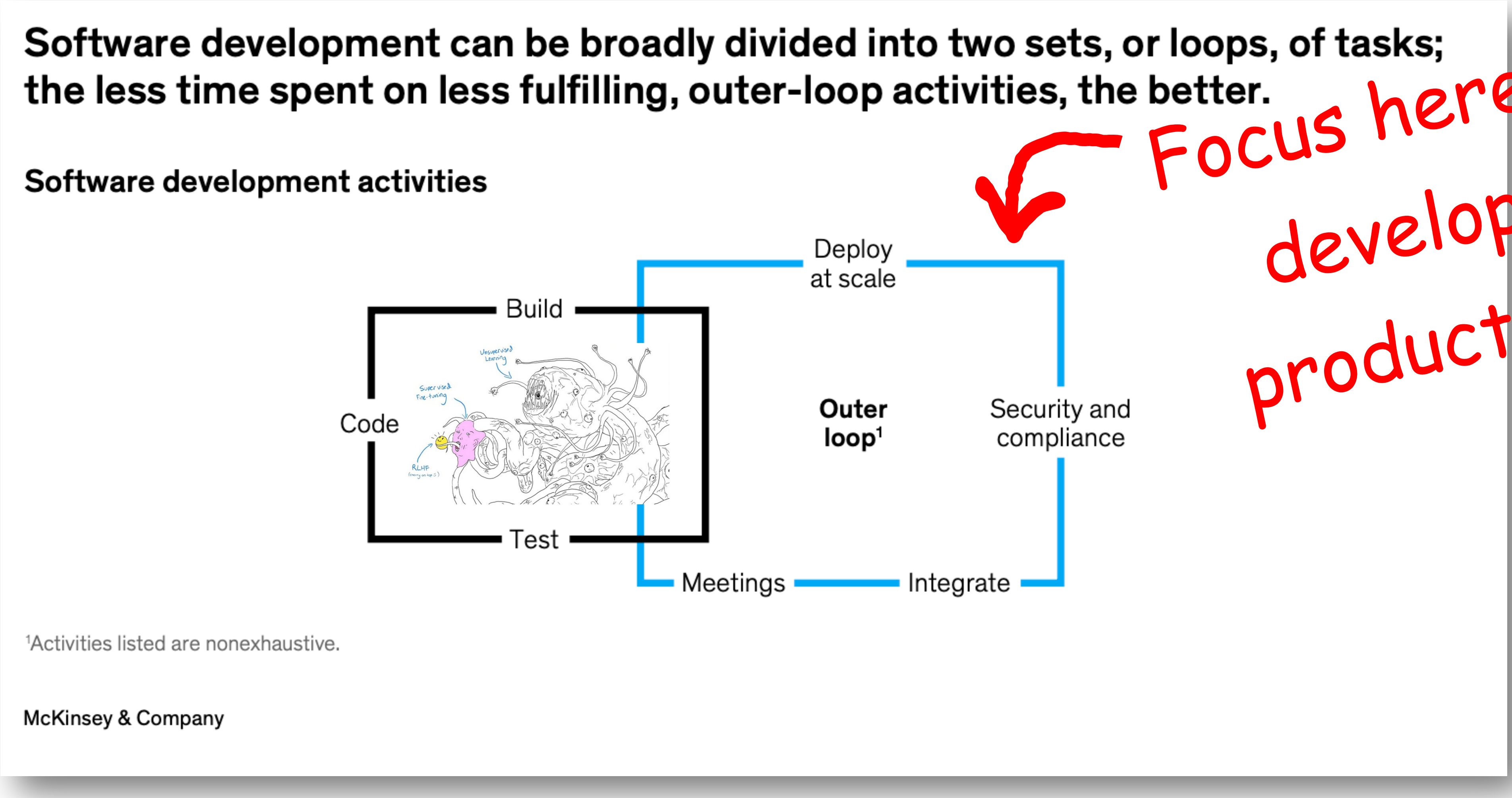
Sources: "State of Kubernetes Survey 2024," VMware Tanzu, June, 2024.

Accidental platform gardening



Kubernetes use in Spring environments continued to grow this year, reaching 65% of respondents. More than half (52%) run a *Kubernetes distribution (DIY, TKG, Rancher, EKS, etc.)*, a third (33%) use a *platform based on Kubernetes (OpenShift, TAP, etc.)*, and more than a quarter (26%) use a *non-Kubernetes based platform (CloudFoundry, Heroku, etc.)*. We find the fact that half start with a *Kubernetes distribution* rather than a more complete platform a little surprising since so much extra work is required.

Sources: ”The State of Spring 2024,” VMware Tanzu, May 2024.



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.

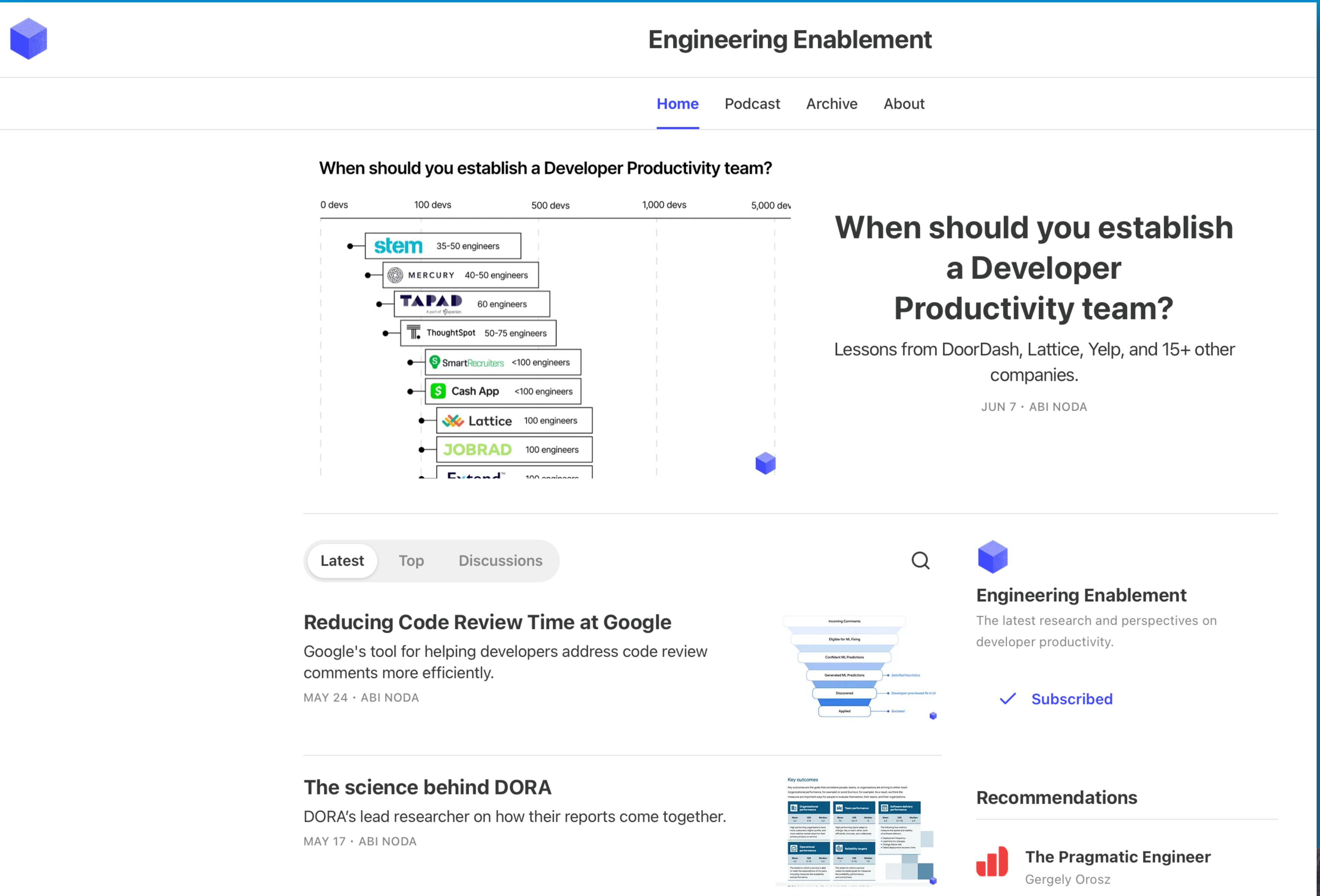
Software development activities

[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.



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

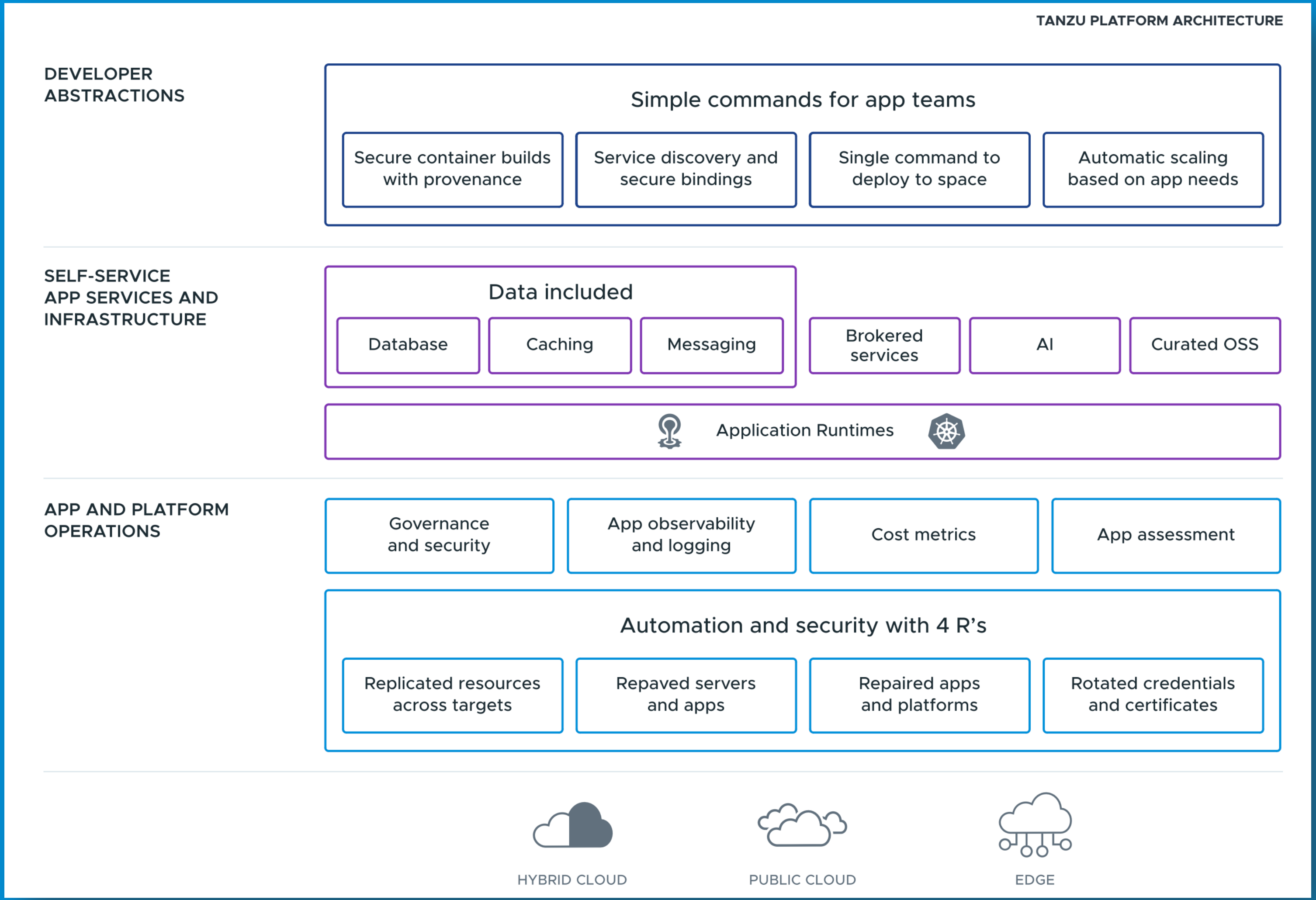
Engineering Enablement

The latest research and perspectives on developer productivity.

✓ Subscribed

Recommendations

The Pragmatic Engineer
Gergely Orosz



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!

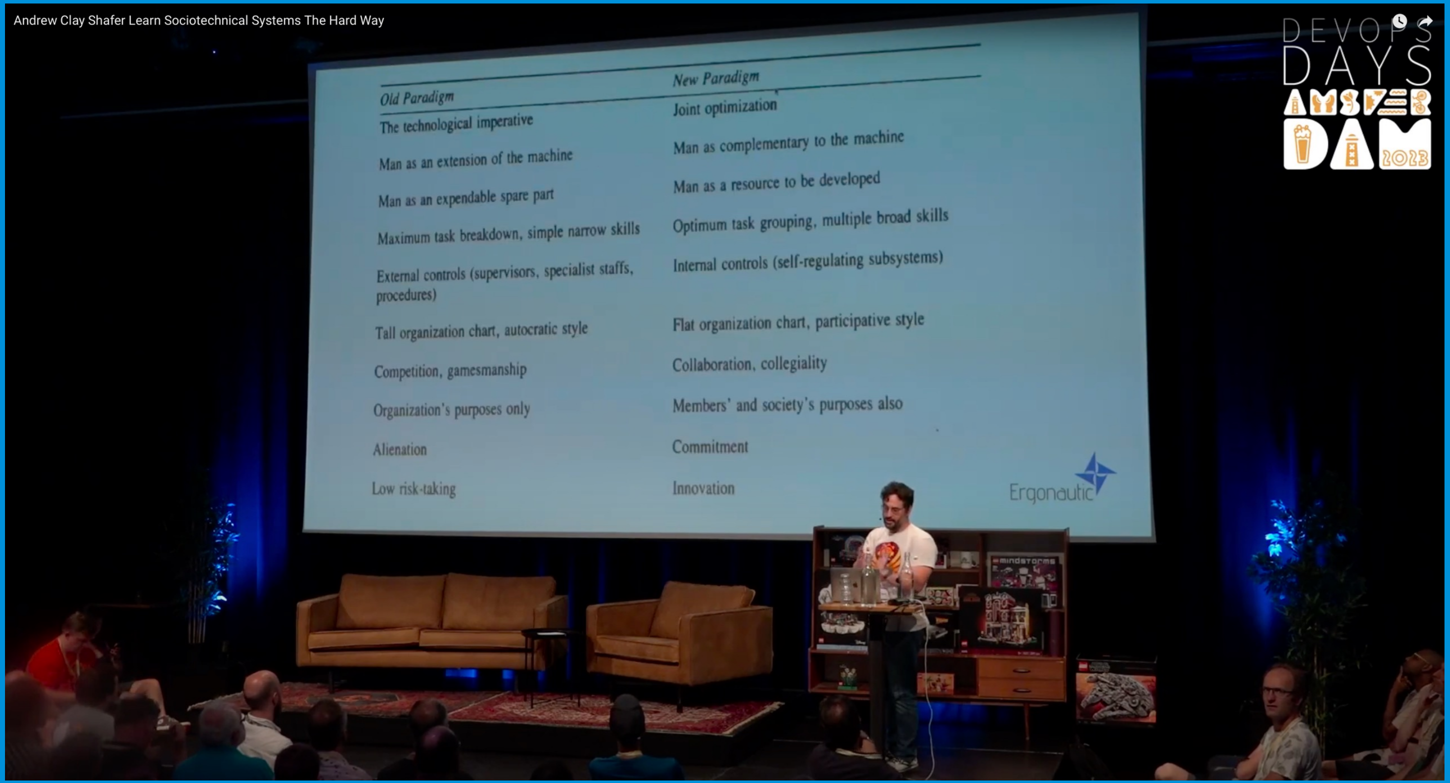
🏗️ https://tanzu.vmware.com/platform

📥 https://newsletter.cote.io/

Slides!

Appendix:
Grandpa Stories

1940’s to 1950’s



Andrew Clay Shafer Learn Sociotechnical Systems The Hard Way

Old Paradigm	*New Paradigm*
The technological imperative	Joint optimization
Man as an extension of the machine	Man as complementary to the machine
Man as an expendable spare part	Man as a resource to be developed
Maximum task breakdown, simple narrow skills	Optimum task grouping, multiple broad skills
External controls (supervisors, specialist staffs, procedures)	Internal controls (self-regulating subsystems)
Tall organization chart, autocratic style	Flat organization chart, participative style
Competition, gamesmanship	Collaboration, collegiality
Organization's purposes only	Members' and society's purposes also
Alienation	Commitment
Low risk-taking	Innovation

Sources: “Learn Sociotechnical Systems The Hard Way,” Andrew Clay Shafer, July, 2023.

Source: curated by “Tech Time Capsule: Early 1990s Clip Art Captured an Era,” Benj Edwards.

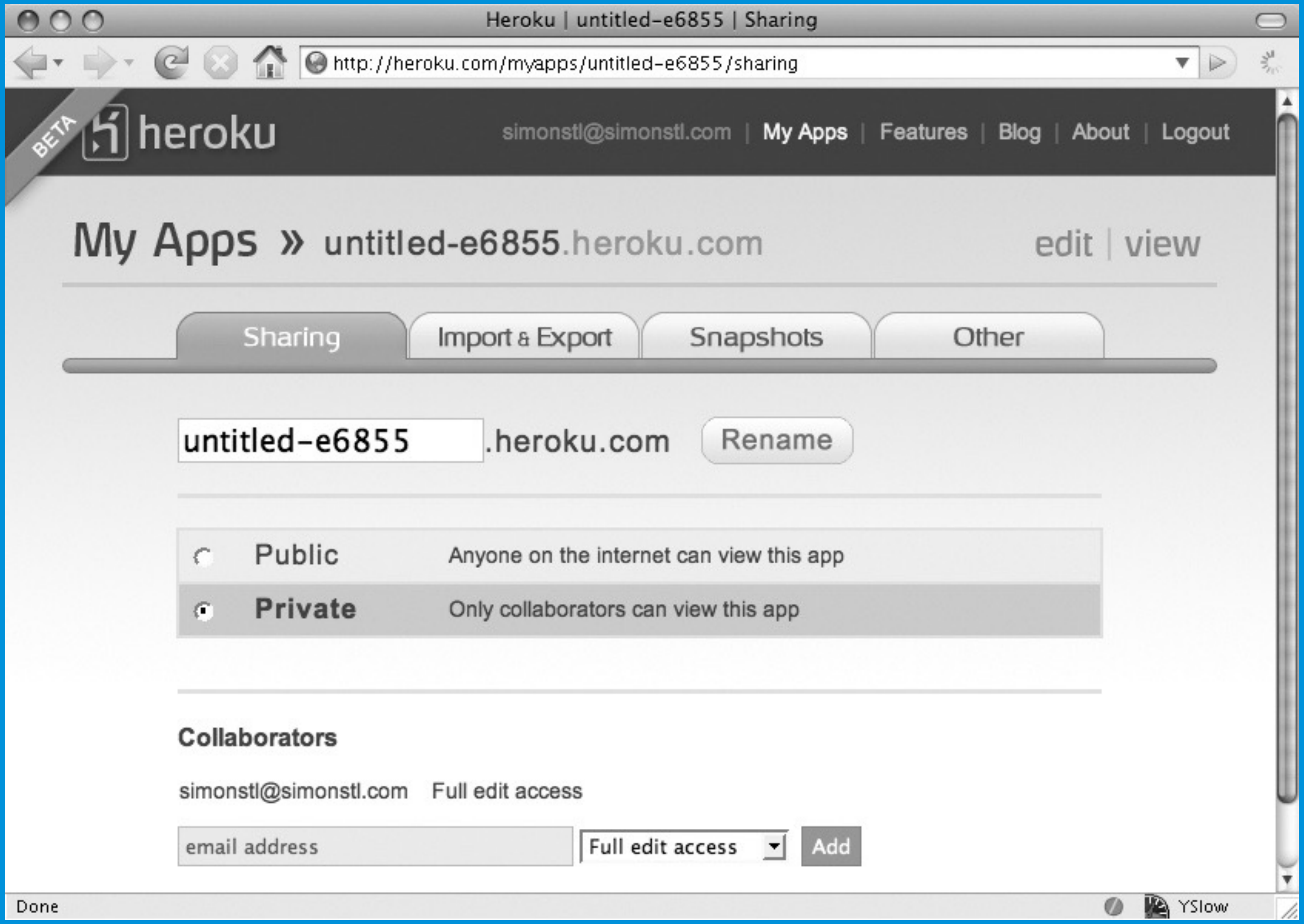
2003

spring® by VMware Tanzu

WS -

Sources: “They’re rebuilding the Death Star of complexity,” dhh, as recounted on Jan 2023.

2007



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

edit | view
Sharing
Import & Export
Snapshots
Other

untitled-e6855.heroku.com Rename

○ Public Anyone on the internet can view this app

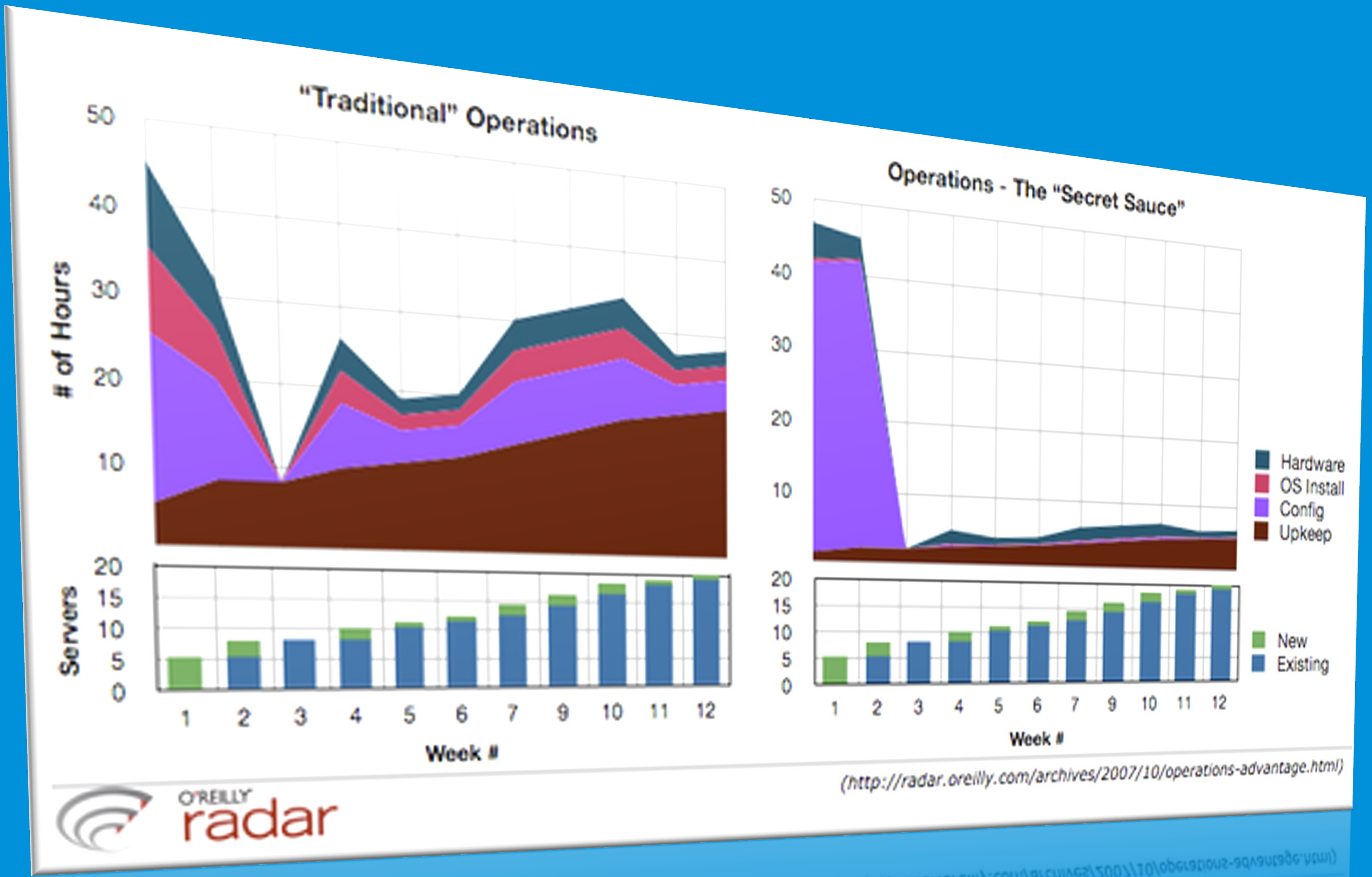
● Private Only collaborators can view this app

Collaborators

simonstl@simonstl.com Full edit access

email address | Full edit access | Add

Sources: *Learning Rails*, Simon St. Laurent and Edd Dumbill, 2008.



(http://radar.oreilly.com/archives/2007/10/operations-advantage.html)

Sources: “Operations is a competitive advantage… (Secret Sauce for Startups!)” Jesse Robins, Oct 2007.

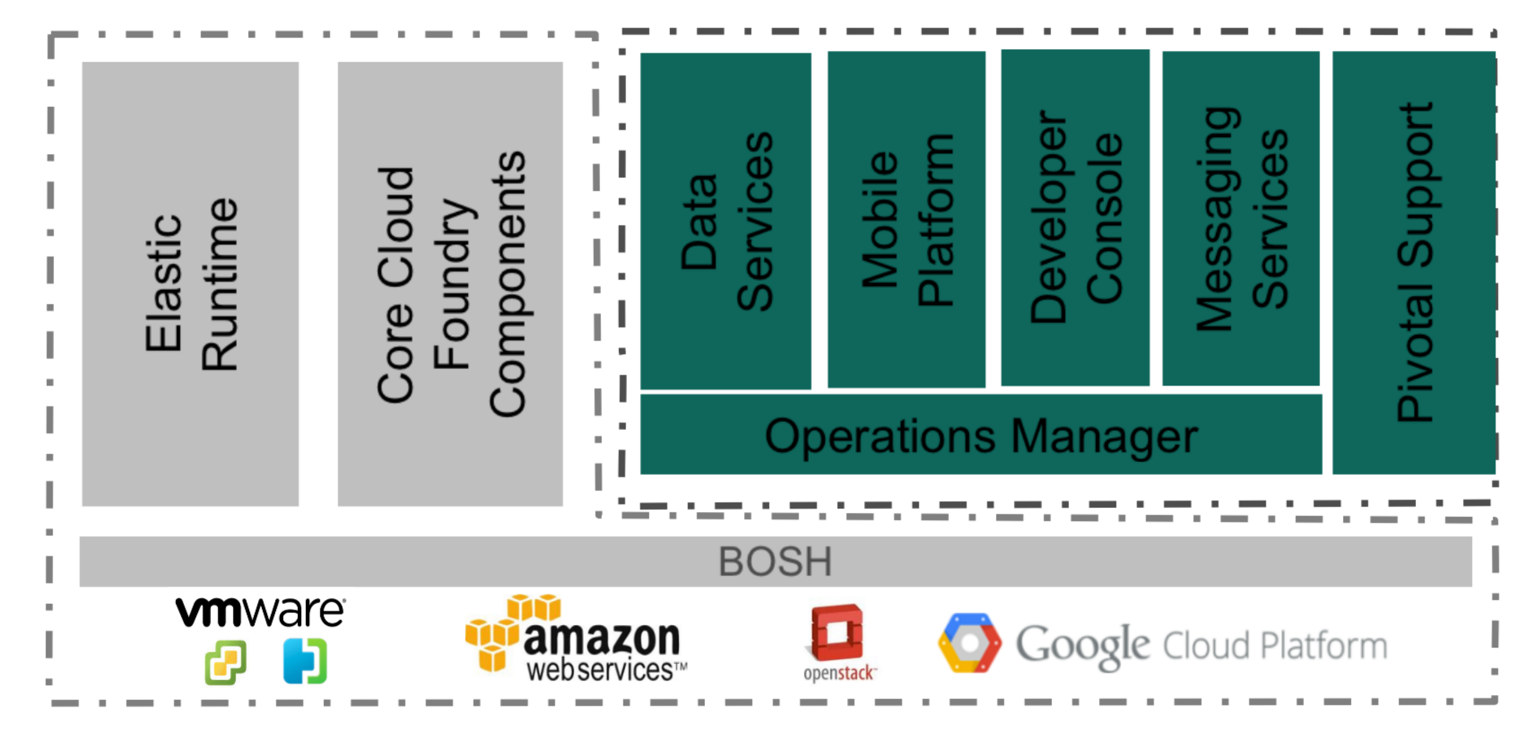
2009

Little bit weird
Sits closer to the boss
Thinks too hard

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

Source: “10+ Deploys Per Day: Dev and Ops Cooperation at Flickr,” John Allspaw & Paul Hammond, 2009.

2015



Sources: "Technical Dive into Cloud Native Application Platforms," Brian Gracely, 2015.

2017

containers

1 OF 11

TEXT ONLY

Check for updates

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

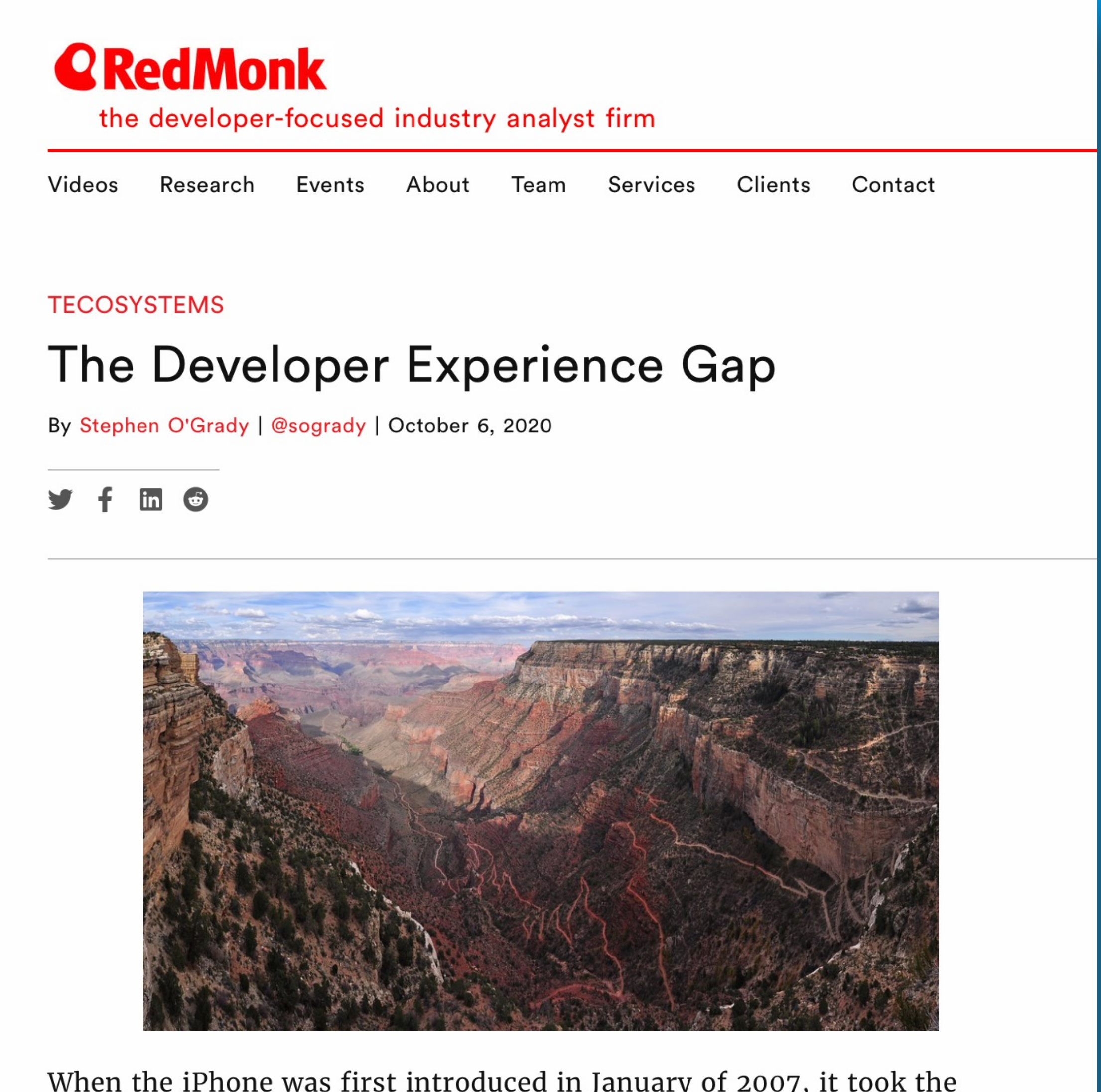
BRIDGET KROMHOUT

COMPLEX SOCIO-TECHNICAL

We focus so often on technical anti-patterns, neglecting similar problems inside our social

Source: "Containers Will Not Fix Your Broken Culture," Bridget Kromhout, Dec 2017.

2020



RedMonk

the developer-focused industry analyst firm

TECOSYSTEMS

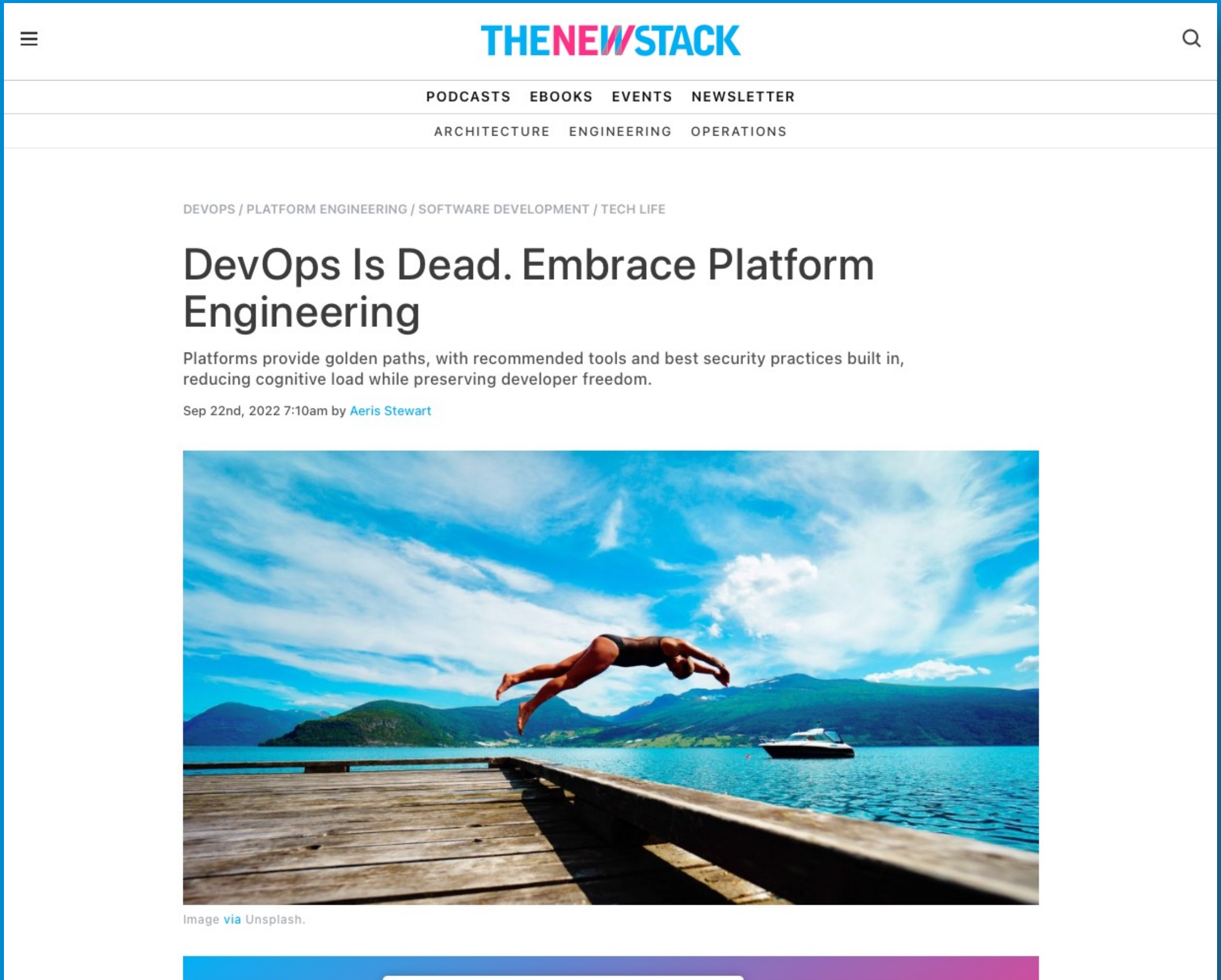
The Developer Experience Gap

By Stephen O'Grady | @sogrady | October 6, 2020

When the iPhone was first introduced in January of 2007, it took the

Sources: "The Developer Experience Gap," Stephen O'Grady, RedMonk, October 2020.

2022



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

Image via Unsplash.

Source: "DevOps Is Dead. Embrace Platform Engineering," Aeris Ransom, Humanitec, Sep 22nd

2023

McKinsey & Company

Technology, Media & Telecommunications Practice

Yes, you can measure software developer productivity

Measuring, tracking, and benchmarking developer productivity has long been considered a black box. It doesn't have to be that way.

This article is a collaborative effort by Chandra Gnanasambandam, Martin Harrysson, Alharith Hussin, Jason Keovichit, and Shivam Srivastava, representing views from McKinsey's Digital and Technology, Media & Telecommunications Practices.

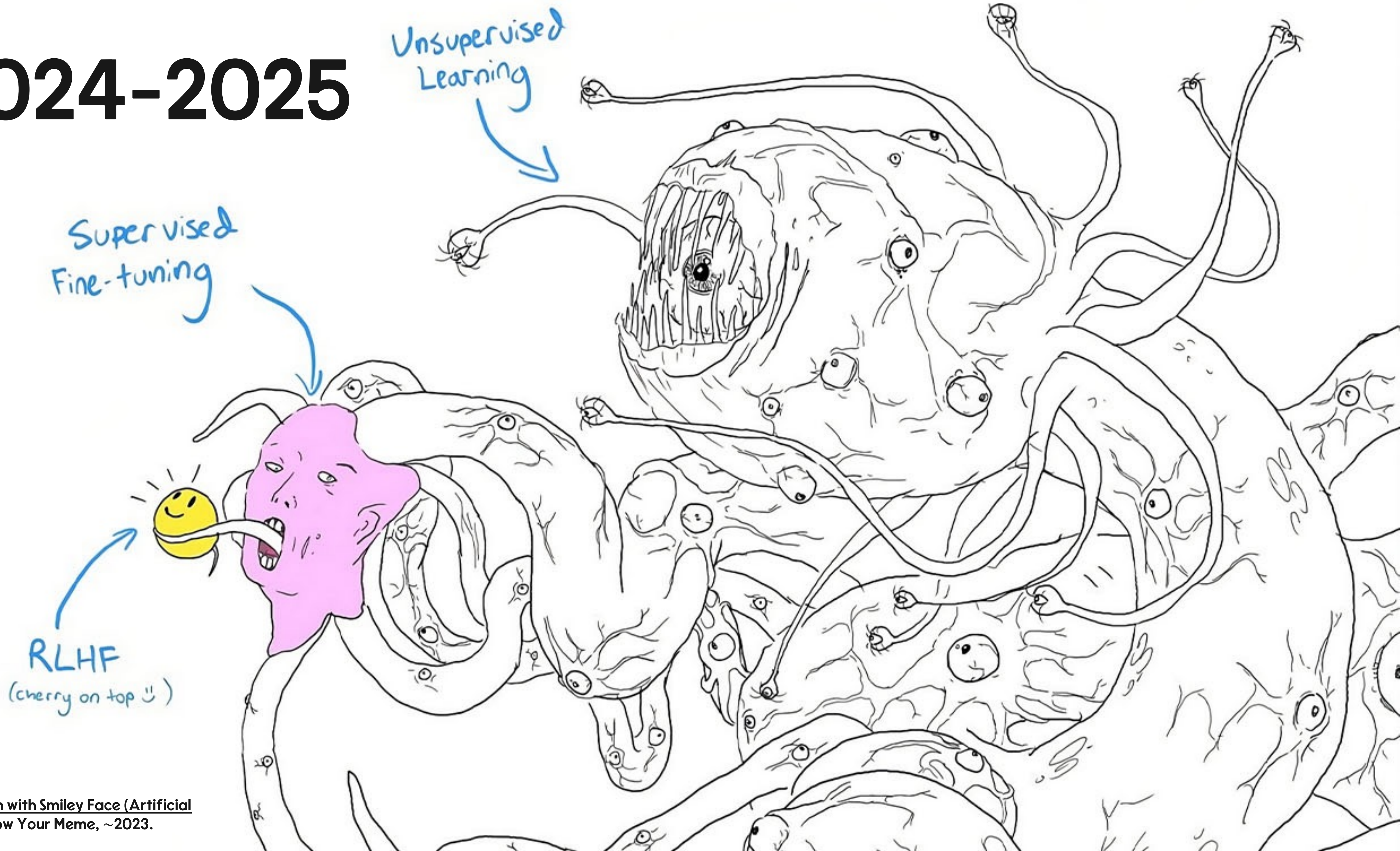
Source: "Yes, you can measure software developer productivity," Chandra Gnanasambandam, Martin Harrysson, Alharith Hussin, Jason Keovichit, Shivam Srivastava, McKinsey Aug(?) 2023.

2023

26% of organizations are using GenAI to support application development, testing, and management and 25% are utilizing machine learning with the greatest focus on leveraging AI to support DevOps analytics and process, governance, and security testing.

Source: "Applying Artificial Intelligence to Strengthen Application Development and Product Life-Cycle Management," Pete Marston, IDC, Nov 2023

2024-2025



Source: “Shoggoth with Smiley Face (Artificial Intelligence),” Know Your Meme, ~2023.