



Software Developments

New insights and surprises from DORA research

Dave Stanke

Developer Advocate, DORA / Google

DevOpsDays Denver 2023



Assertion:

**We can measure software
delivery performance.**

*...and, informed by such measurements,
through a process of continuous learning...*

We can **improve software
delivery performance.**

can we, though?

Agenda

- 01 DORA: DevOps Research and Assessment
- 02 Durable findings
- 03 Emerging insights
- 04 Surprises in 2022
- 05 Q&A
- 06 One last surprise

01 02 03 04 05 06

DORA

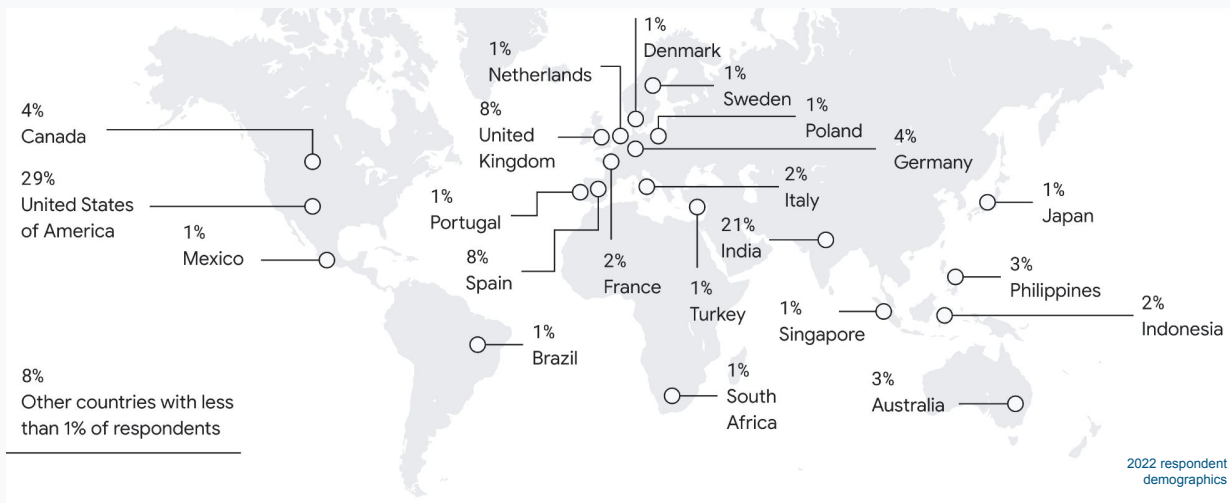
DevOps Research and Assessment



Let's look at the science

Research

Over **33,000 professionals** have participated in DORA research surveys since 2014



Data

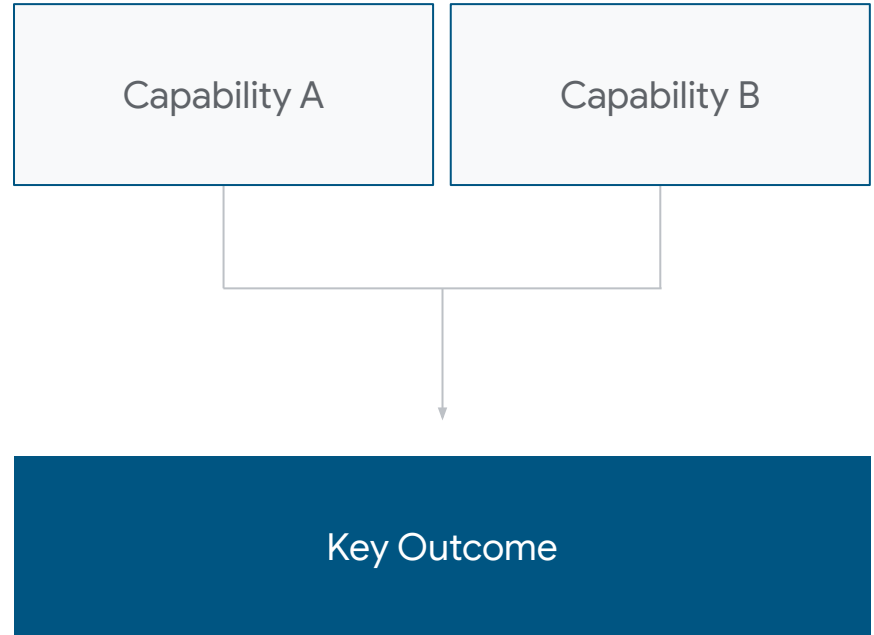
Want to participate in an upcoming study?
Join [dora.community](#) for announcements!

A small sample of the information we anonymously collect as part of our research:

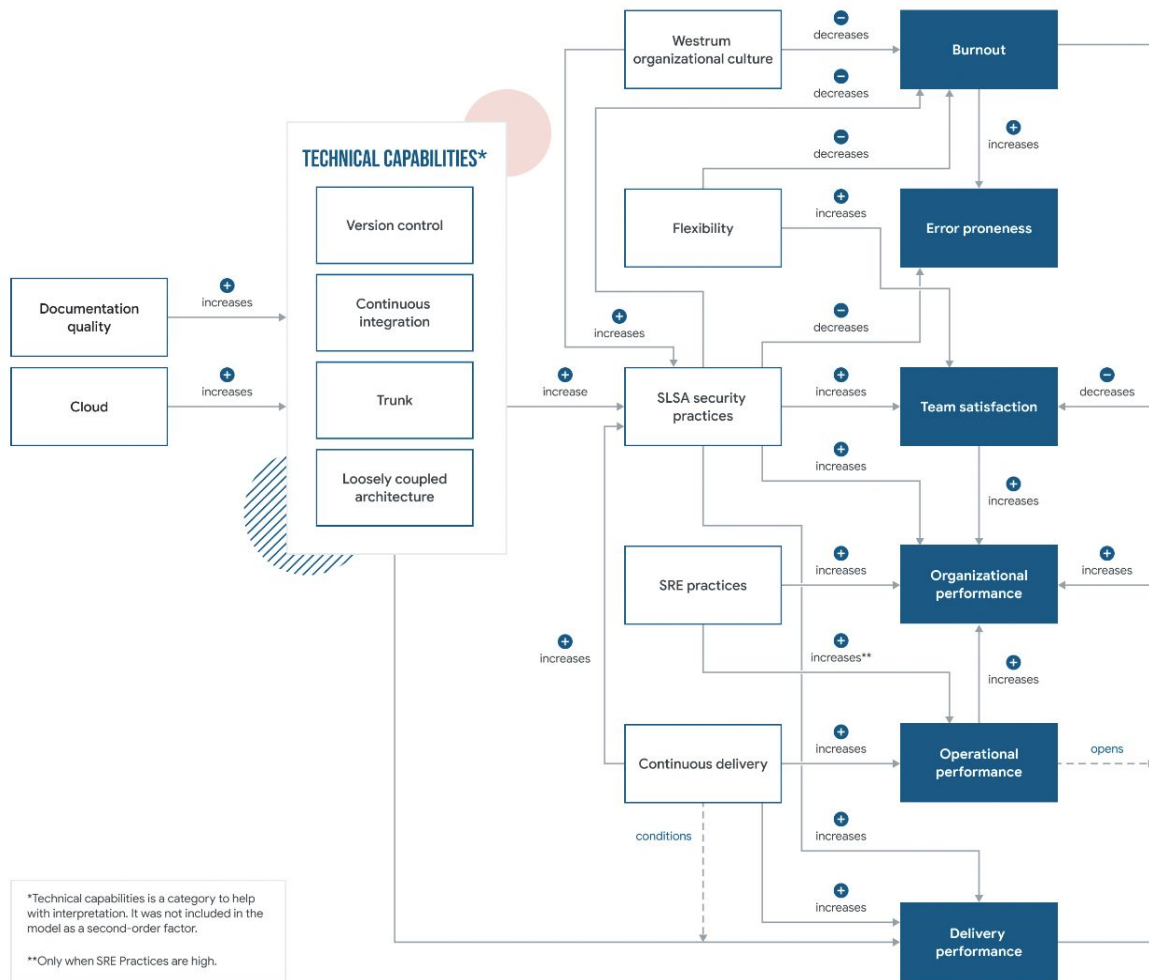
- My team can deploy and release our product or service on demand, independently of other services it depends upon
- On my team, we can make large-scale changes to the design of our system without depending on other teams to make changes in their systems
- The cloud my product or service runs on serves multiple teams and applications, with compute and infrastructure resources dynamically assigned and re-assigned based on demand
- We can deploy our system to production, or to end users, at any time, on demand
- Code commits result in an automated build of the software
- What is the principal industry of your organization?
- Technical documentation is updated as changes are made
- Failures are treated primarily as opportunities to improve the system
- Reliability reviews are performed throughout the development process for all major features on the applications I work on
- Build metadata (e.g., dependencies, build process, build environment) about an artifact includes all build parameters
- For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?
- I feel burned out from my work.
- Our org has processes in place to identify and document all security requirements for the software our organization develops or acquires (including third-party and open source)
- Most of the people that were on this team 12 months ago are still on the team today
- There are fewer than three active branches on the application's code repo
- Our application configurations are in a version control system
- Currently, how inflexible or flexible is your company with regard to employee work arrangements (e.g., voluntary work from home, full-time remote work, hybrid schedules, etc.)?
- Cross-functional collaboration is encouraged and rewarded

Analysis

Predictive analysis by DORA



Analysis



*Technical capabilities is a category to help with interpretation. It was not included in the model as a second-order factor.

**Only when SRE Practices are high.



bit.ly/dora-sodr

01 02 03 04 05 06

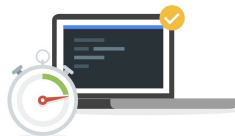
Durable findings

We can measure software delivery



Deployment frequency

How often does your organization deploy code to production or release it to end users?



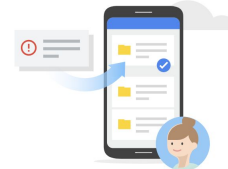
Lead time for changes

How long does it take to go from code committed to code successfully running in production?



Change fail rate

What percentage of changes to production or releases to users result in degraded service?



Time to restore service

How long does it generally take to restore service when a service incident or a defect that impacts users occurs?

Throughput

Stability

Software delivery matters

Software delivery performance **predicts** organizational performance, as measured by essential KPIs, like **increased market share**, and organizational health metrics, like **reduced burnout**.

Capabilities drive performance

The screenshot shows a web browser displaying the DORA DevOps capabilities catalog. The page features a navigation bar with the DORA logo and links for Publications, Research, Capabilities, Quick Check, Resources, and Community. The main heading is "Capability catalog", followed by an introductory paragraph. Below this, a section titled "Technical capabilities" lists twelve categories, each with a brief description and a "Learn more" link. The categories are: Cloud infrastructure, Code maintainability, Continuous delivery, Continuous integration, Continuous testing, Database change management, Deployment automation, Empowering teams to choose tools, Loosely coupled architecture, Monitoring and observability, Shifting left on security, and Test data management.

dora.dev/capabilities

Culture matters

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

Anyone can do it

We've observed these dynamics in every...

- **technical context**
 - mainframe, microservice, mobile
- **industry**
 - financial services, government, retail
- **age of organization**
 - legacy, startup

But only you are you

The DORA framework is not a maturity model, it's a capability model. **Your context is unique.** Find your bottleneck. Then improve, measure, and repeat.

Capabilities

Technical

- Trunk-based development
- Cloud infrastructure
- Shifting left on security
- ...

Process

- Work in small batches
- Streamlined change approval
- Visibility of work in value stream
- ...

Cultural

- Generative, trust-based
- Learning culture
- Transformational leadership
- ...

Predict

Software Delivery and Operations Performance

Predict

Organizational Outcomes

(e.g. market share, profitability, employee retention)

As measured by

- **Throughput**
 - lead time for changes
 - deployment frequency
- **Stability**
 - time to restore service
 - change failure rate
- **Reliability**

01 02 03 04 05 06

Emerging insights

Security

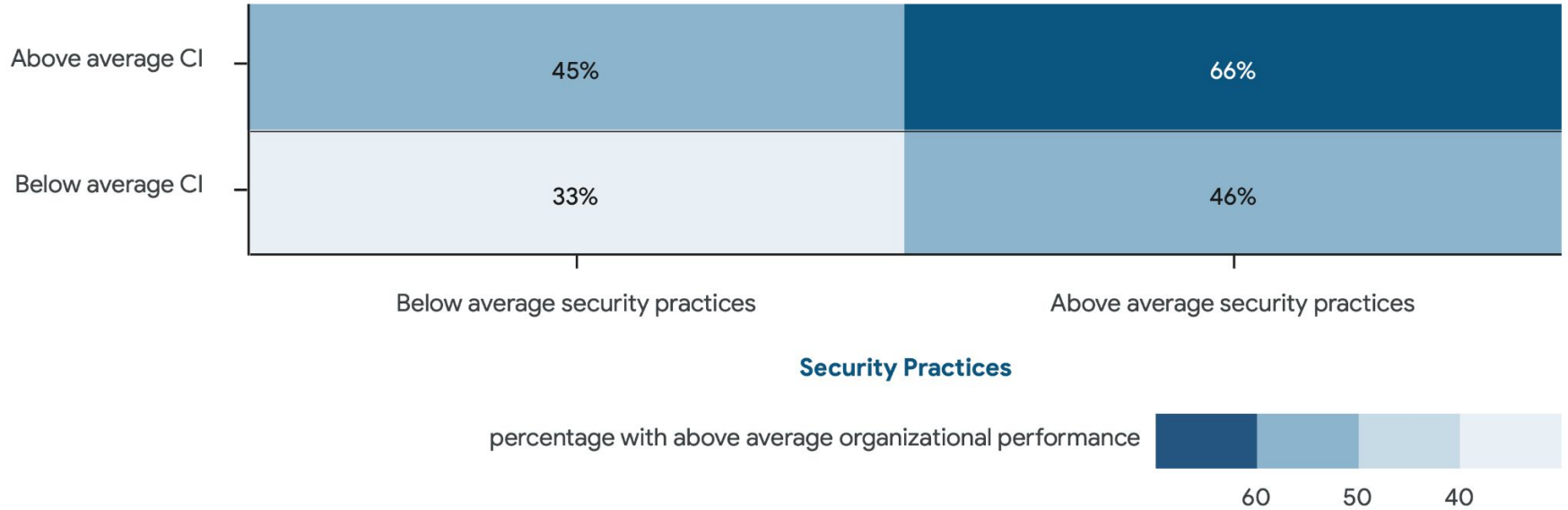
Established finding:

“Shifting Left” on security predicts software delivery performance

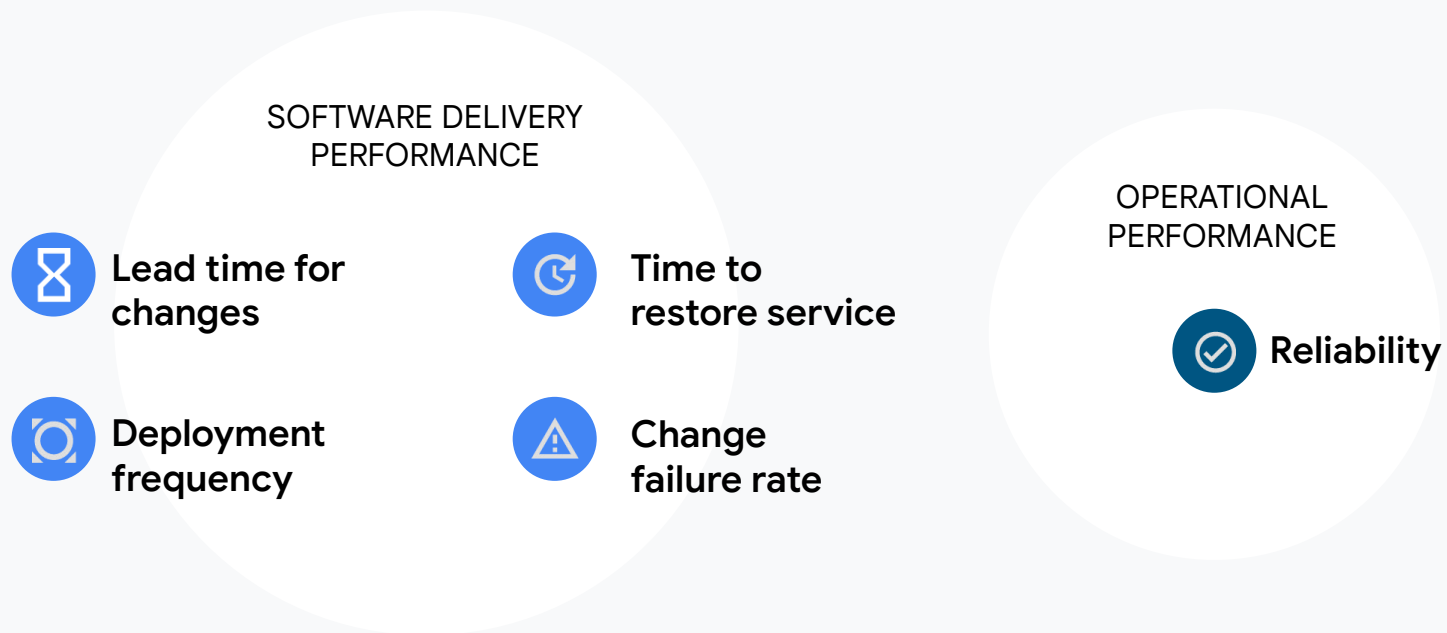
New research avenues:

- Exploring the Software Supply Chain in depth
- Leveraging industry frameworks like SLSA and SSDF

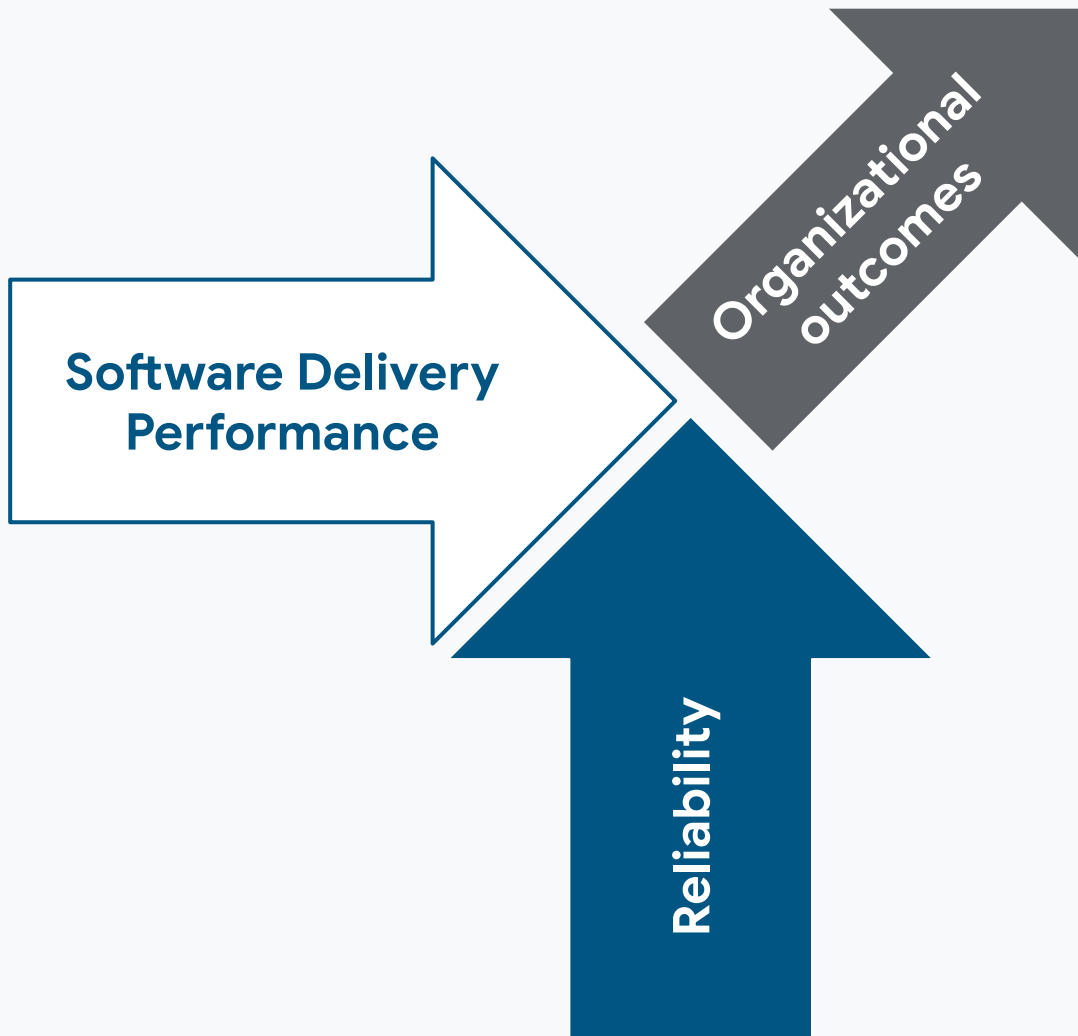
Security



Reliability



Reliability

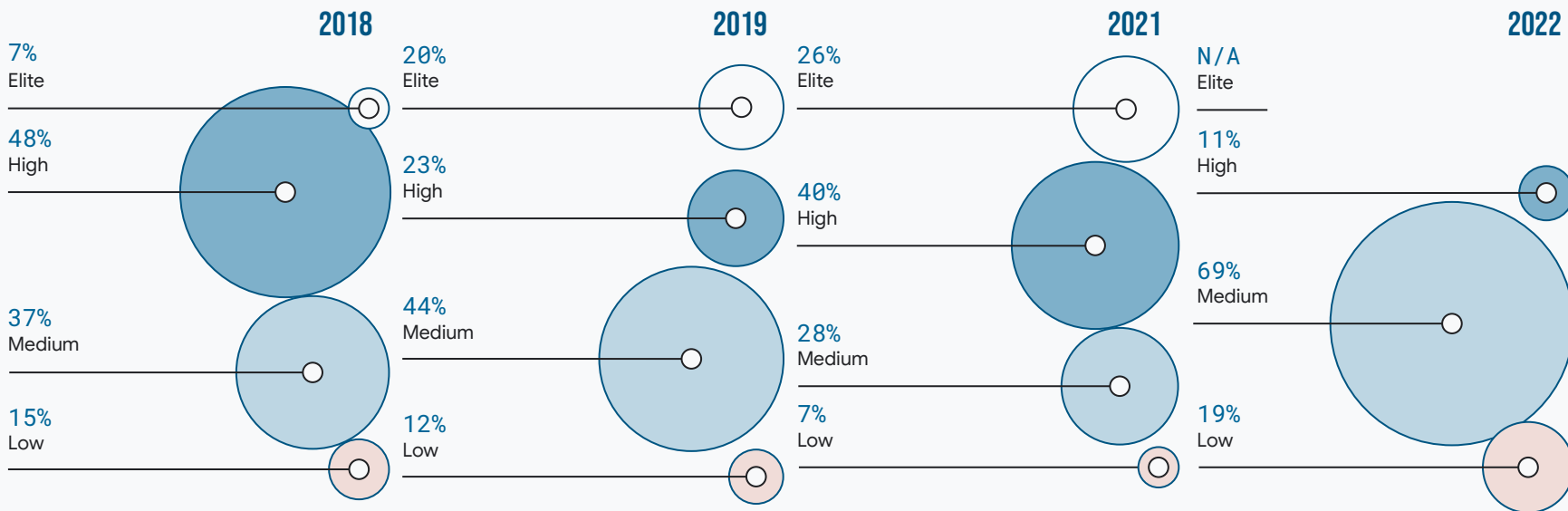


01 02 03 04 05 06

Surprises in 2022

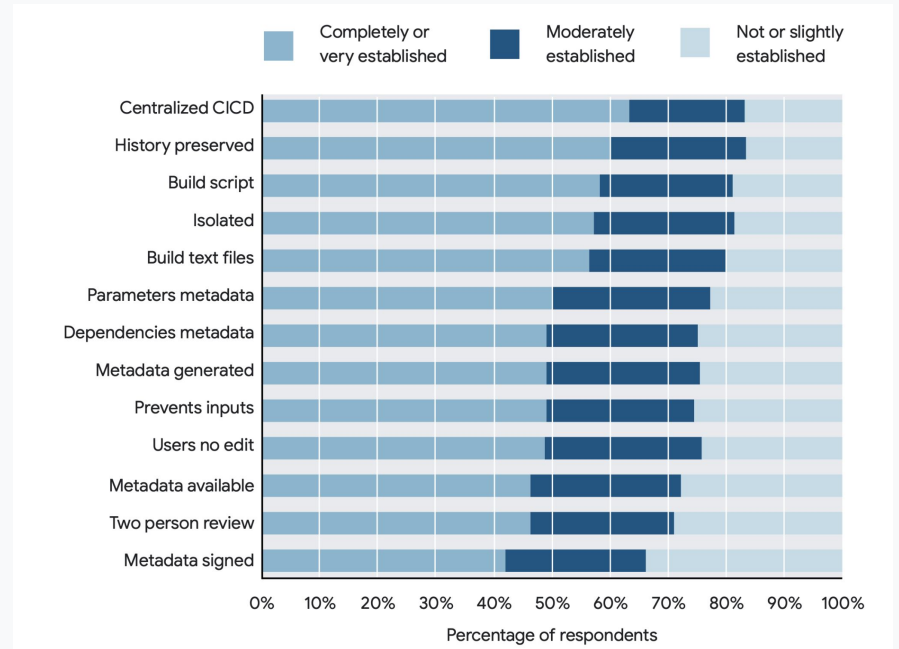
Remember, this is science

A shift in the clusters



Security *drives* software delivery performance

Security practices were found to be the mechanism through which technical capabilities impacted software delivery performance and organizational performance.



Trunk-based development has unclear effect

Trunk-based development

Short-lived branches (less than 1 day), frequently merged to main/trunk

Prior to 2022:

trunk-based development → higher software delivery performance

2022:

trunk-based development → *lower* software delivery performance

Reliability is more than predictive; it's essential

Software delivery performance's effect on organizational performance depends on operational performance (reliability), such that **high software delivery performance is only beneficial to organizational performance when operational performance is also high.**

01 02 03 04 05 06

Q&A

01 02 03 04 05 06

One last surprise

...which, really, we probably sorta knew already...

Continuous Improvement

A finding from the State of DevOps Report 2022:

Teams that recognize the need to continuously improve tend to have higher organizational performance than those that don't.



dora.dev