

DO, RE, Me

Measuring the effectiveness of Site Reliability Engineering

Dave Stanke

Google Cloud Developer Advocate

[@davidstanke](#)





[@davidstanke](#)

Dave Stanke is a Developer Advocate at Google, where he advises customers on their DevOps and SRE journeys. Throughout his career, he has worn all the hats, including startup CTO, product manager, customer support, software developer, sysadmin, and graphic designer. Dave has this crazy idea that software should be useful to the people who use it.

Dave is a contributor to the 2021 and 2022 State of DevOps reports.

RED

OARS

EAR

RODS

RARE

SOD

DORA

SRE



Site Reliability Engineering

DORA

SRE

DORA: DevOps Research and Assessment

Over 32,000 professionals have participated in DORA research surveys since 2014

- International
- Experienced
- Multi Industry
- Startup to Enterprise
- Ops/Devs; ICs/Mgrs

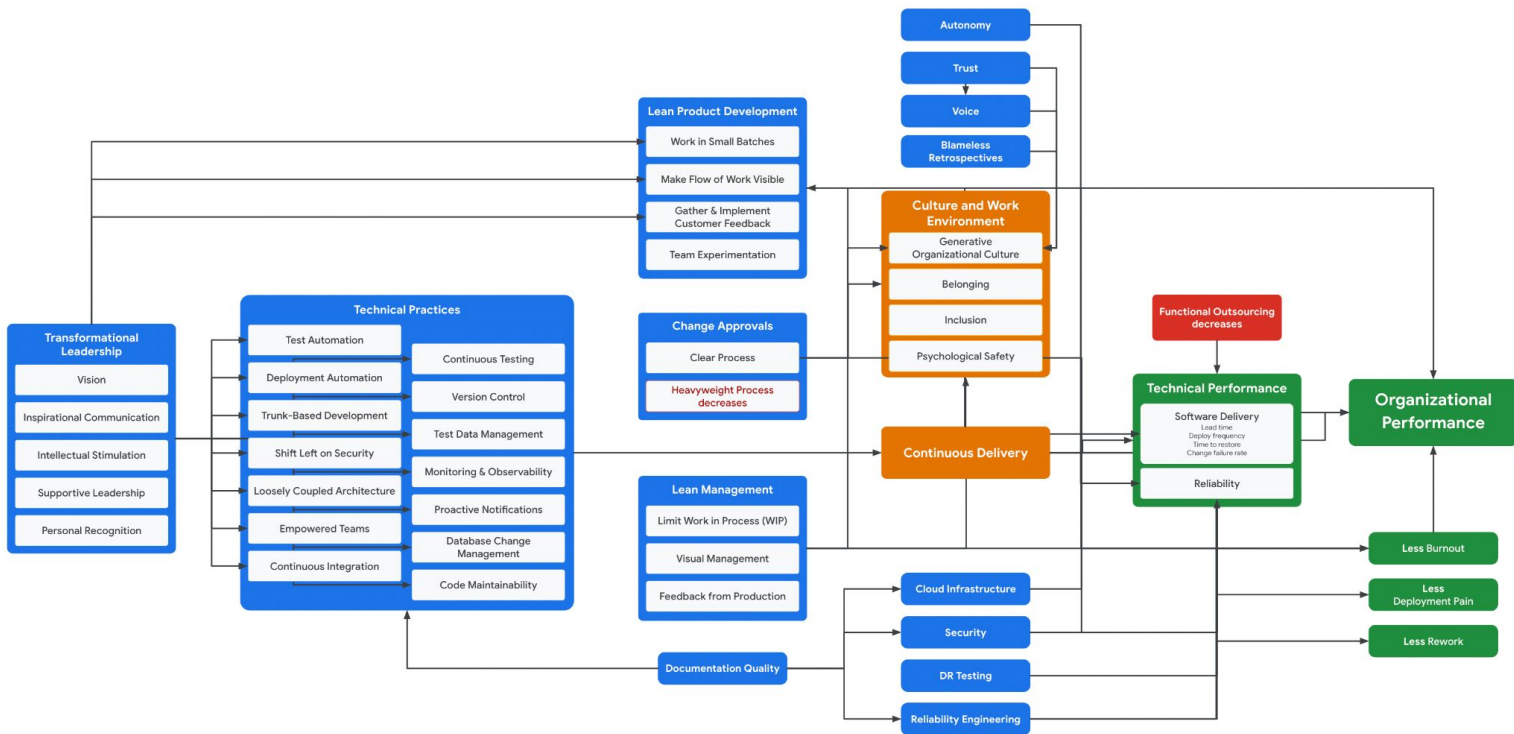


For each of the following performance indicators, how well did your organization meet its goals over the past year?

	Performed well above goals	Performed above goals	Performed slightly above goals	Met goals	Performed slightly below goals	Performed below goals	Performed well below goals	I don't know/ NA
Your organization's overall profitability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Your organization's overall performance	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Increased number of customers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Relative market share for primary products	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

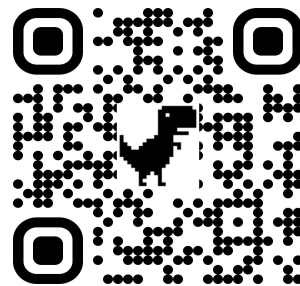
BACK

NEXT



State of DevOps 2021

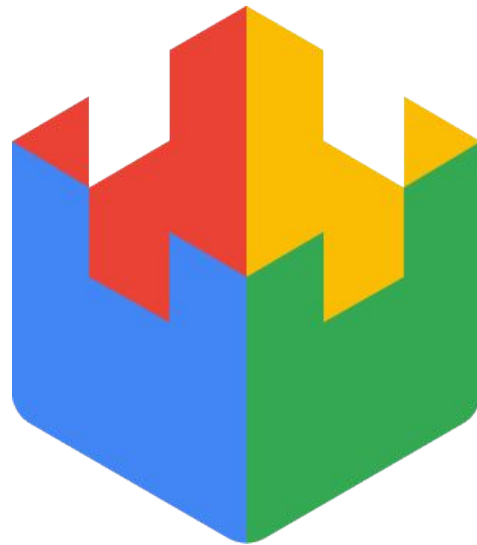
Google Cloud



bit.ly/dora-sodr

SRE: Site Reliability Engineering

- A framework for modern technical ops
- Originated at Google in the early 2000s
- Designed to help distributed services scale sustainably to global user bases and 24x7 uptime



sre.google

SRE: Site Reliability Engineering

SLIs/SLOs

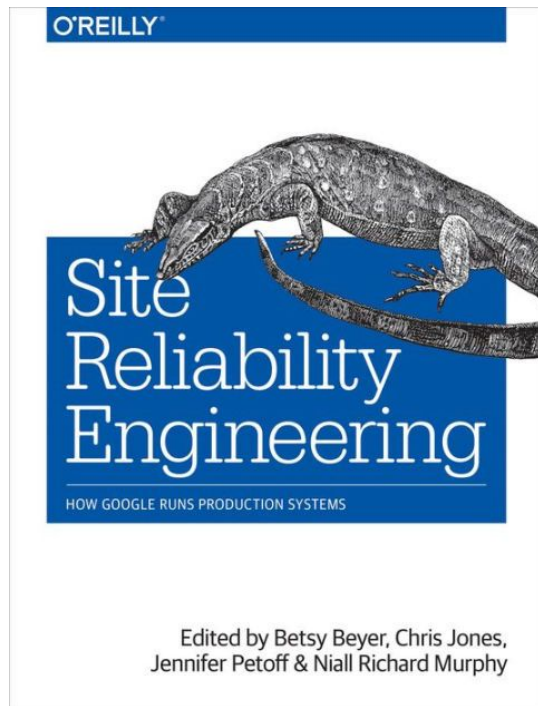
Error budgets

Rational alerting

Disaster preparedness

Toil reduction

Blameless postmortems

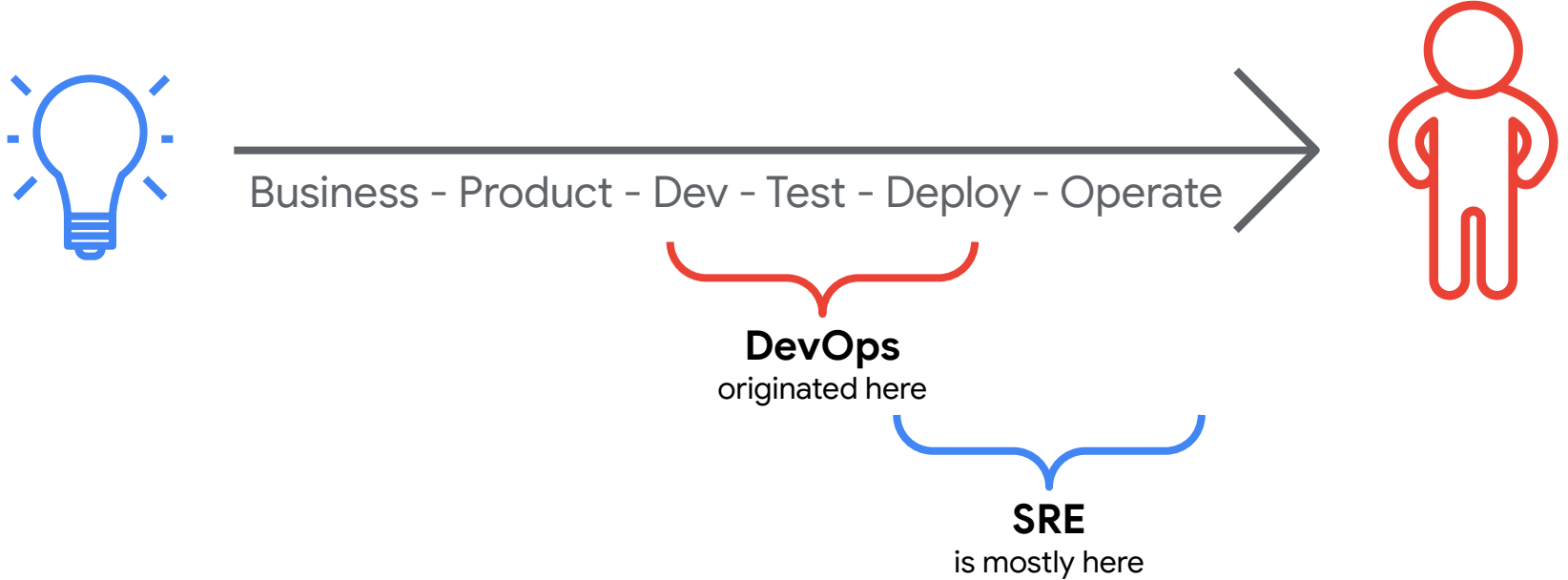


DORA + **SRE** = ?

DevOps' original scope



DevOps' original scope

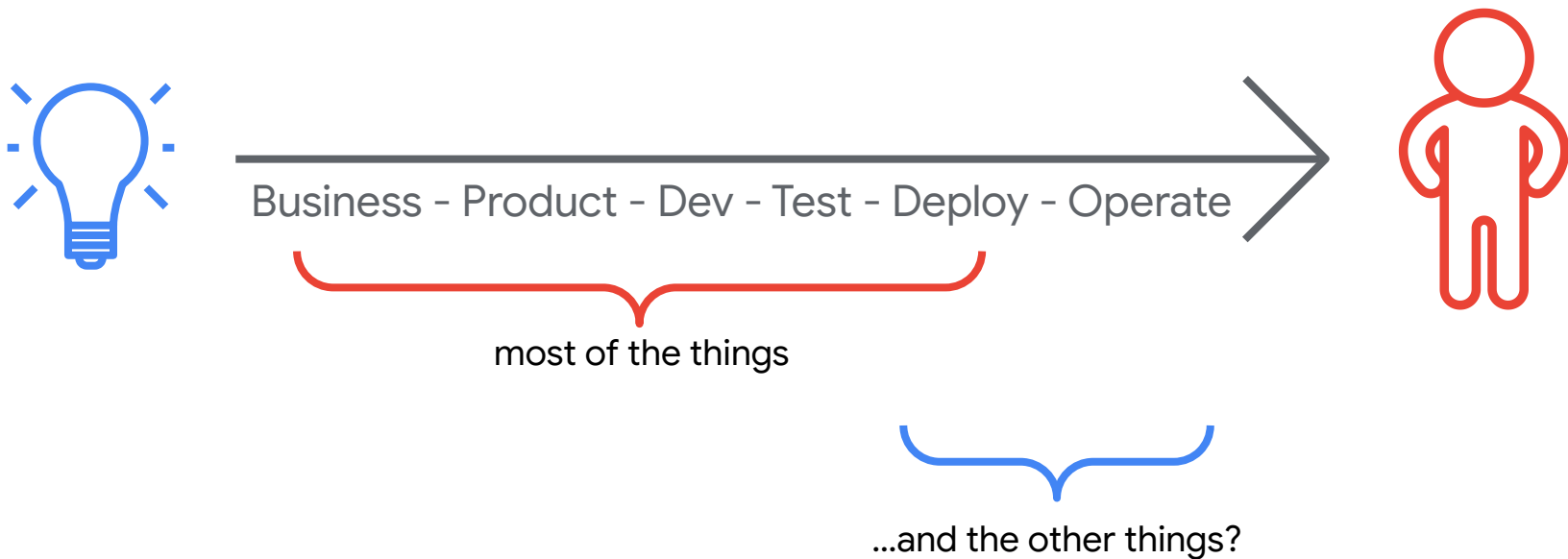


DevOps scope creep

(BizDevOps, DevSecOps, etc)



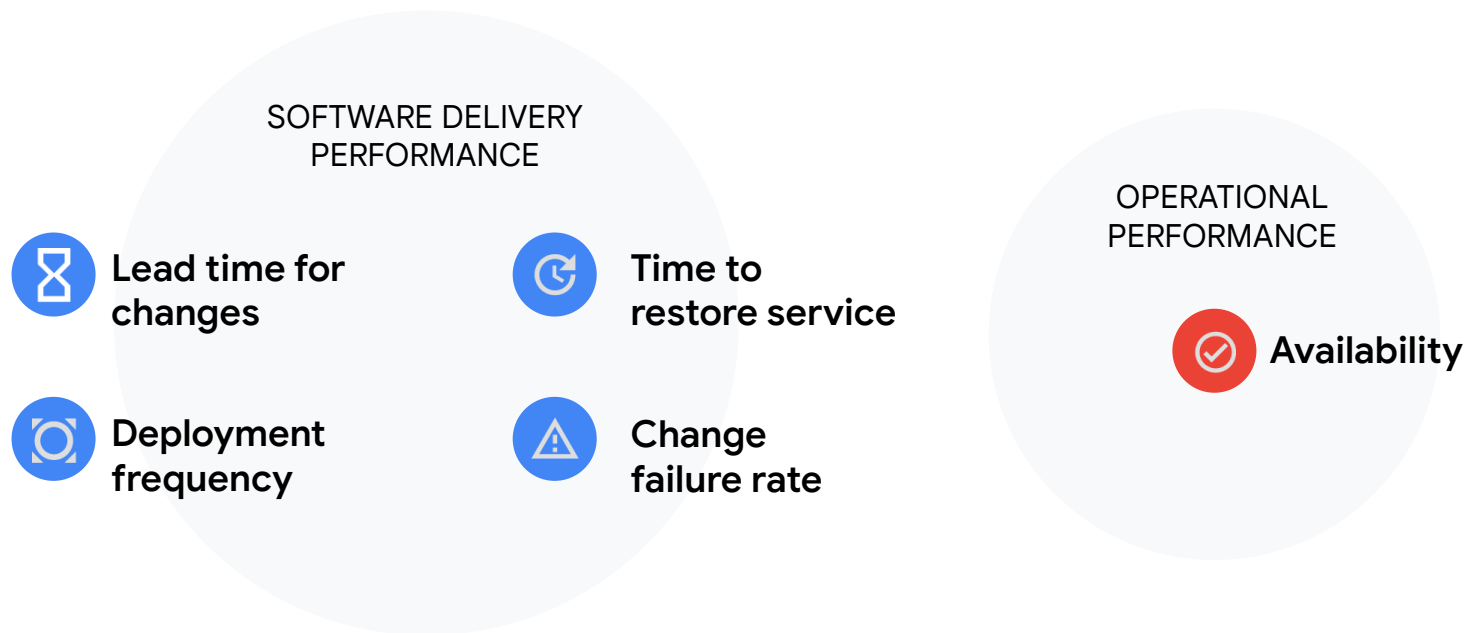
Is DevOps *all the things*?



2021: DORA tackles SRE

Four + 1

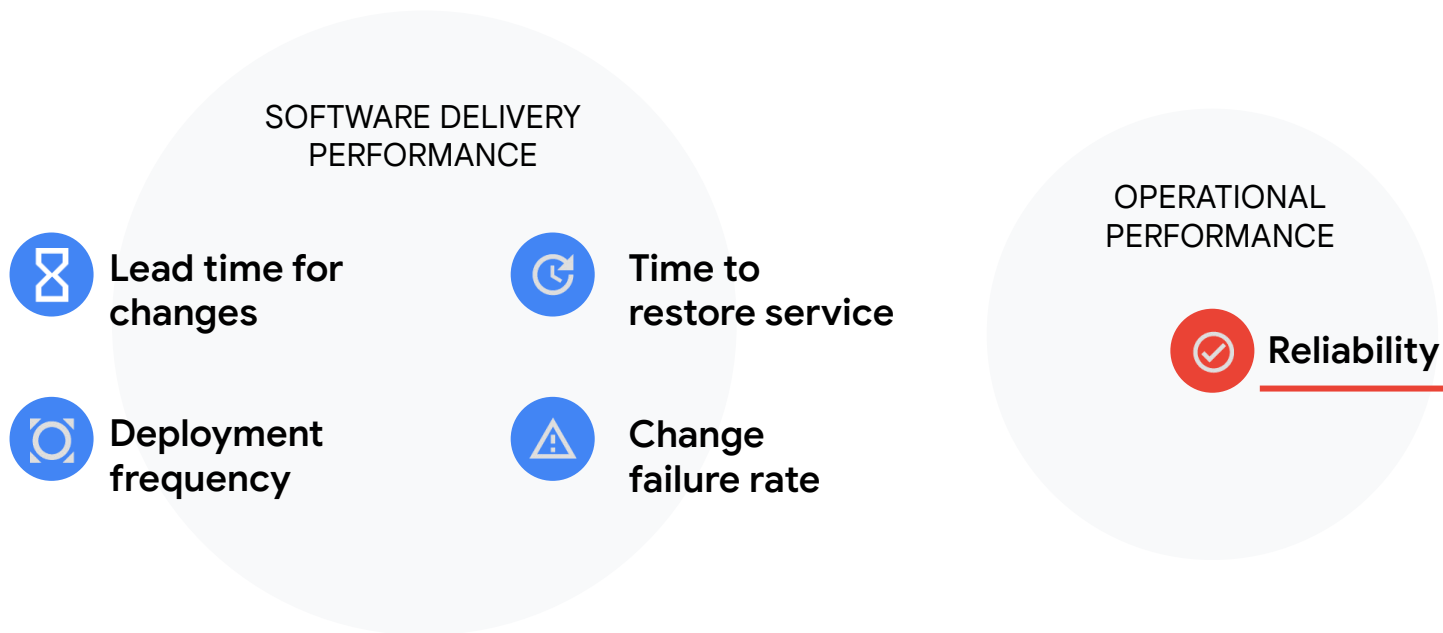
The keys to Software Delivery and Operations



Availability → Reliability

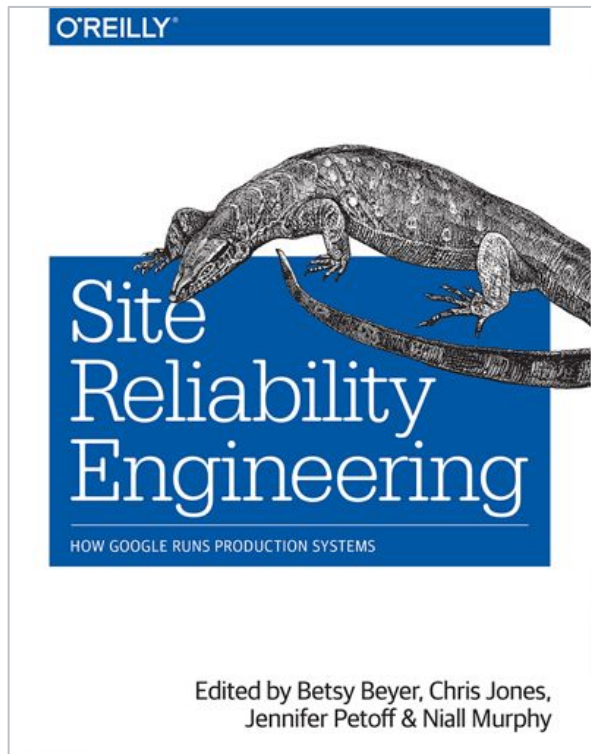
Four + 1

The keys to Software Delivery and Operations



How do we know if a team is doing SRE?





Alerts are generated based on service degradation from the perspective of end users or dependent systems (as opposed to system metrics like CPU %, ICMP latency, or disk IO).

My team actively identifies and eliminates, automates, or streamlines repetitive manual work.

My team is empowered to maintain reliability standards throughout the product development process.

Alerts are configured so that humans are only paged when an immediate response is needed.

My team has well-defined protocols and tools for responding to production incidents.

Reliability reviews are performed throughout the development process for all major features on the applications I work on.

My team regularly practices incident response procedures.

My team regularly reviews and revises reliability targets based on evidence.

My team proactively plans for capacity demand using data from multiple sources.

When we miss our availability targets, we perform improvement work and/or re-prioritize.

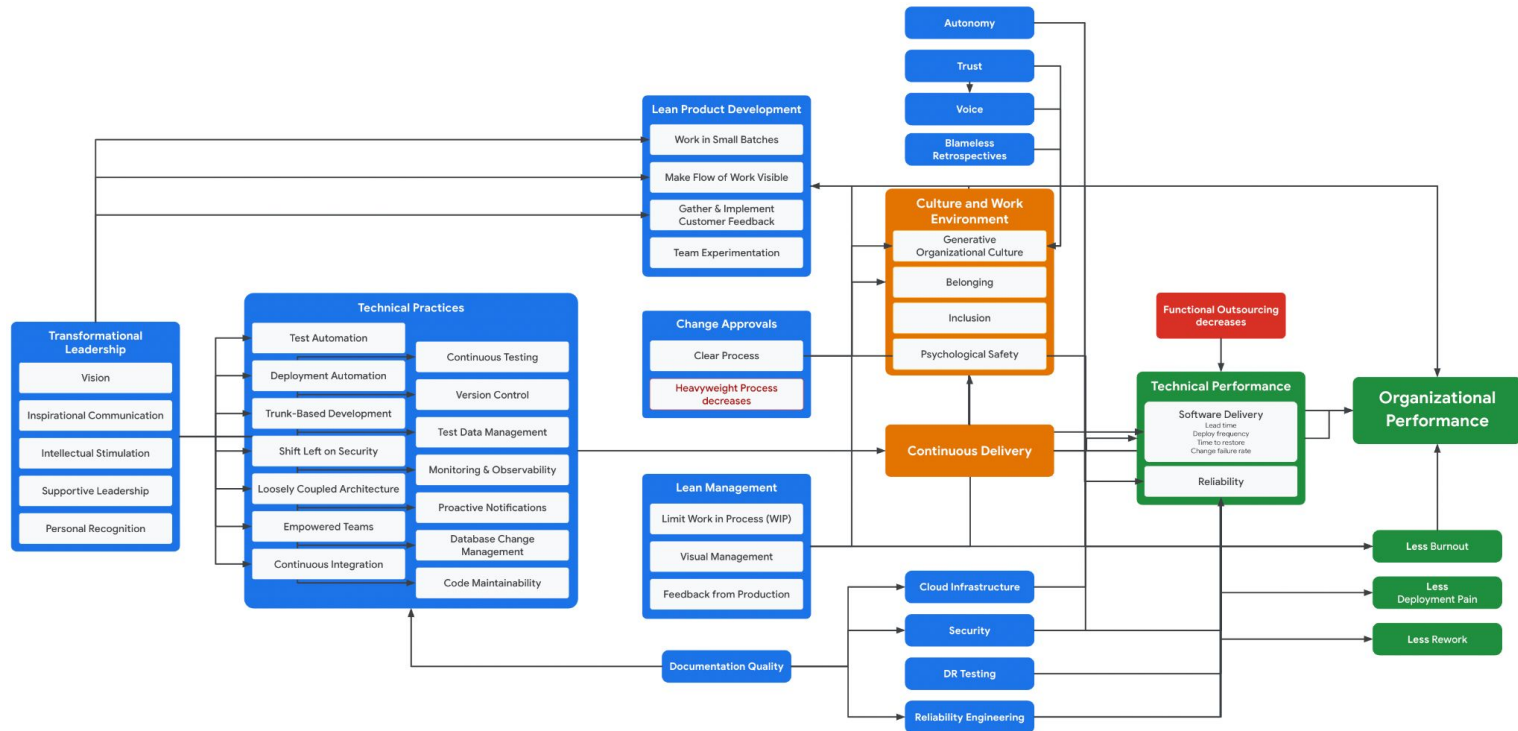
What did we learn?

SRE is widely practiced

A majority of respondents reported the use of SRE practices

(to some degree)

SRE is good for systems and humans



SRE is good for...

Humans

- SRE mitigates burnout
- SRE enables balance between coding and “ops” work

Systems

- “Shared responsibility” for operations predicts better r9y outcomes
- SRE predicts higher reliability

Business

- Higher reliability predicts better business outcomes

Teams that excel at reliability engineering are

1.8x

more likely to meet or
exceed organizational goals

Reliability is a force multiplier

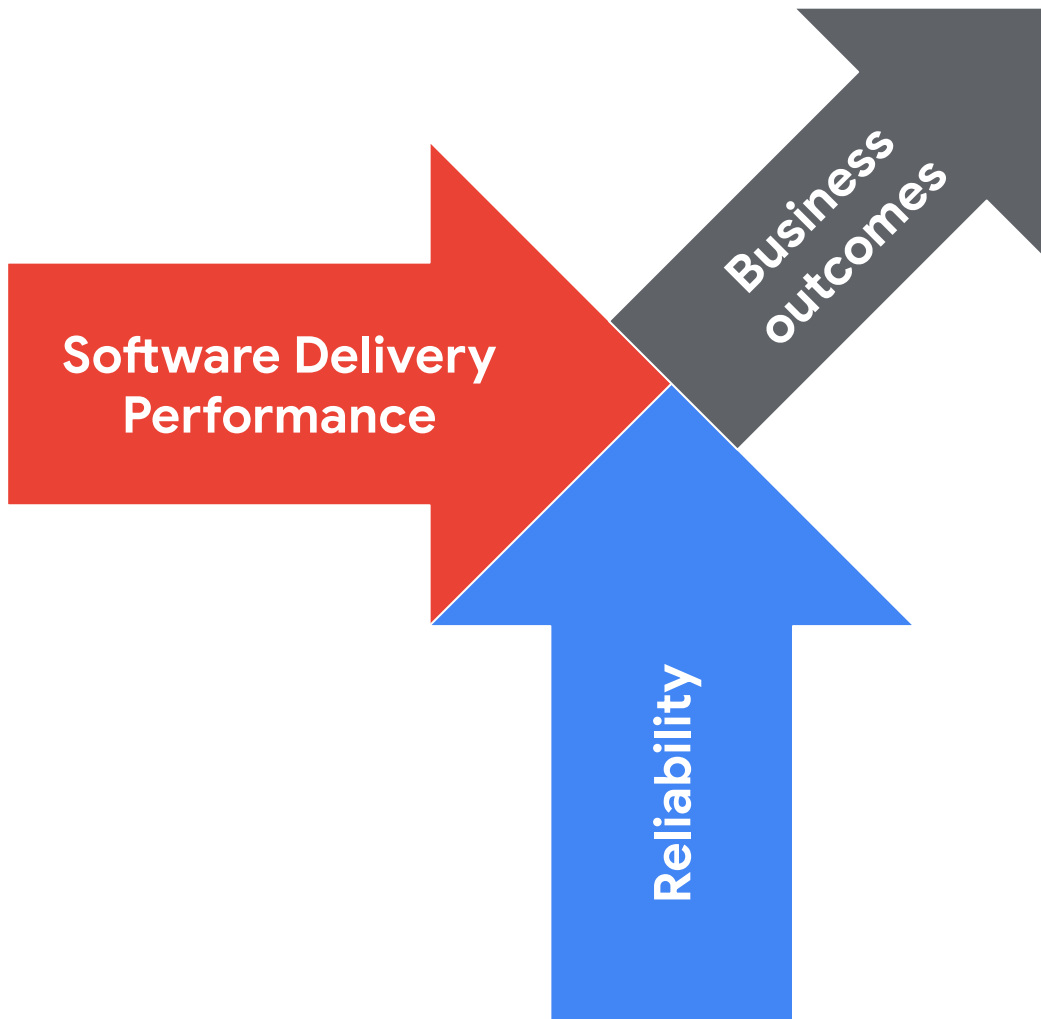


Software Delivery Performance



**Software Delivery
Performance**

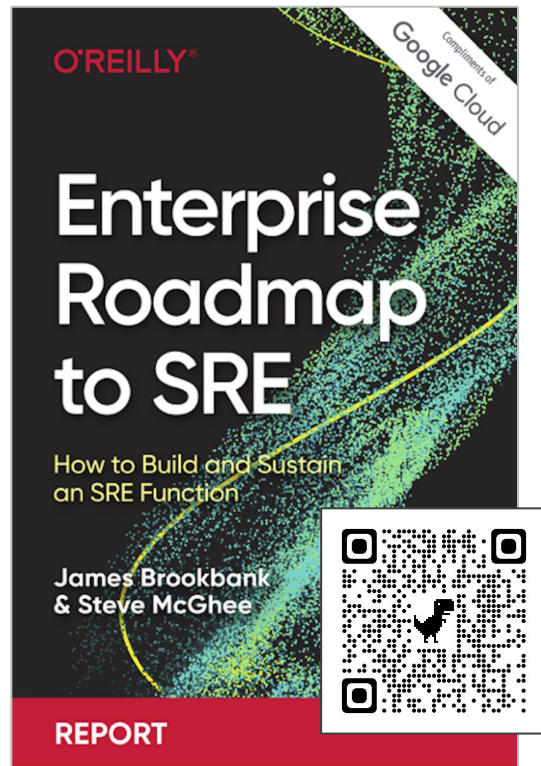
Reliability



Okay, that's nice and all, but...

How do *I* do the SRE?

- Enthusiasm > Successful Adoption of SRE
- Reliability **isn't the most important thing** for everything
- SRE is often seen as expensive or difficult to achieve (usually both)
- Not everyone wants the Google SRE way
 - but they usually still want something that is better than today



Why the SRE approach to Reliability?

- ✓ Product differentiation
- ✓ Risk mitigation
- ✓ Hyperscale services

However! – **not every service** needs SRE

Why the SRE approach to Reliability?

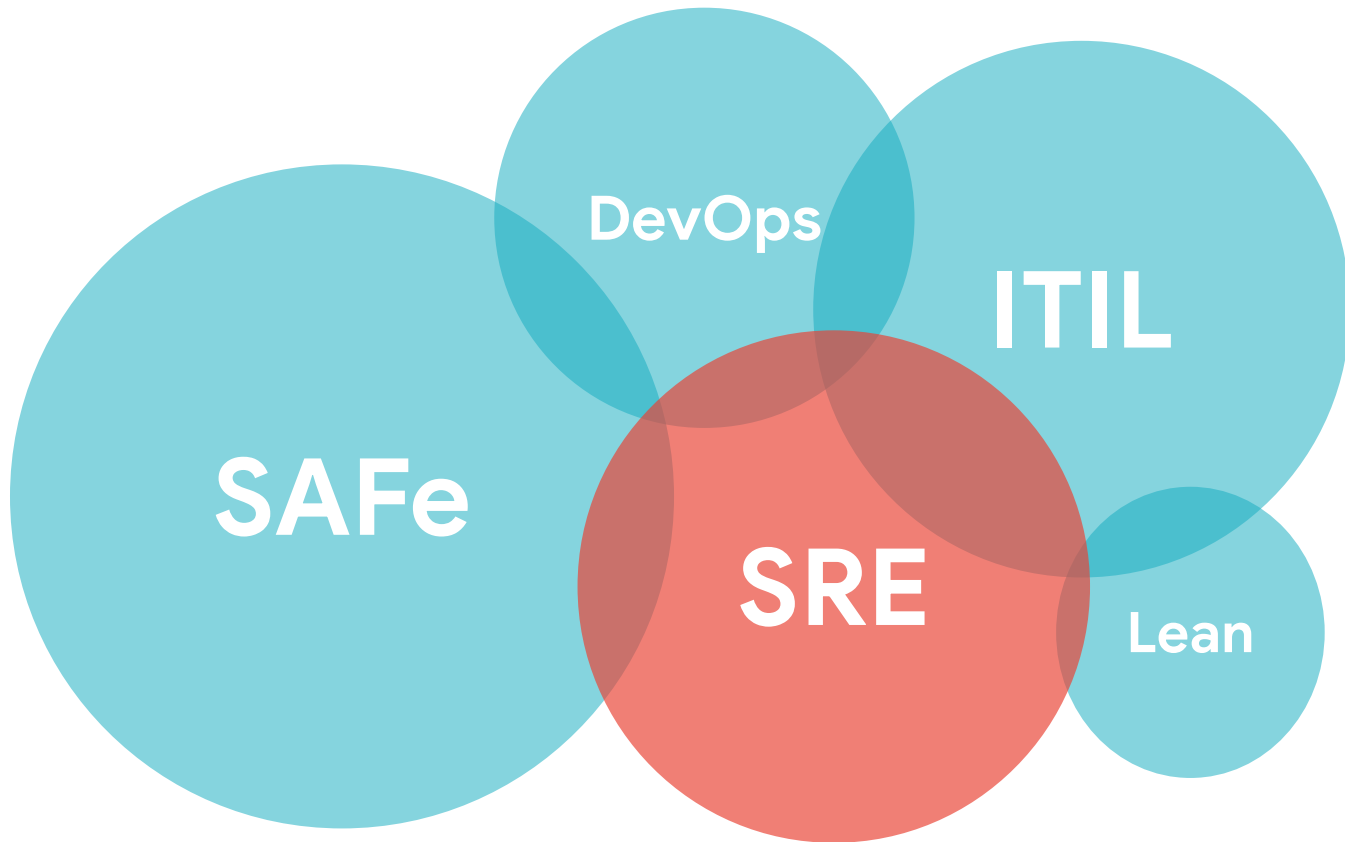
Cost reduction...?

Yes! ... But also no.

SRE is a **strategic investment** (\$↑)
in long-term operational efficiency (\$↓)

Cost optimization is **global**, not **local**.

Getting Started with Enterprise SRE



Approaches to culture change



Communities of practice



Bottom-up or grassroots



Training centers



Centers of excellence



A big bang approach

“A seat at the table”

Consider a Chief Reliability Officer (CRO)

- Compare with a Chief Information Security Officer (CISO)
 - "Security is everyone's responsibility"
- Enterprises also have CISOs to nurture and champion those efforts

Do not let reliability become event driven (i.e. only invest after an outage)

Is it working?

Your SRE is working if...

- Users are happy with your reliability
- Engineers are happy and productive
- You feel good about your ROI

You're on the right track if...

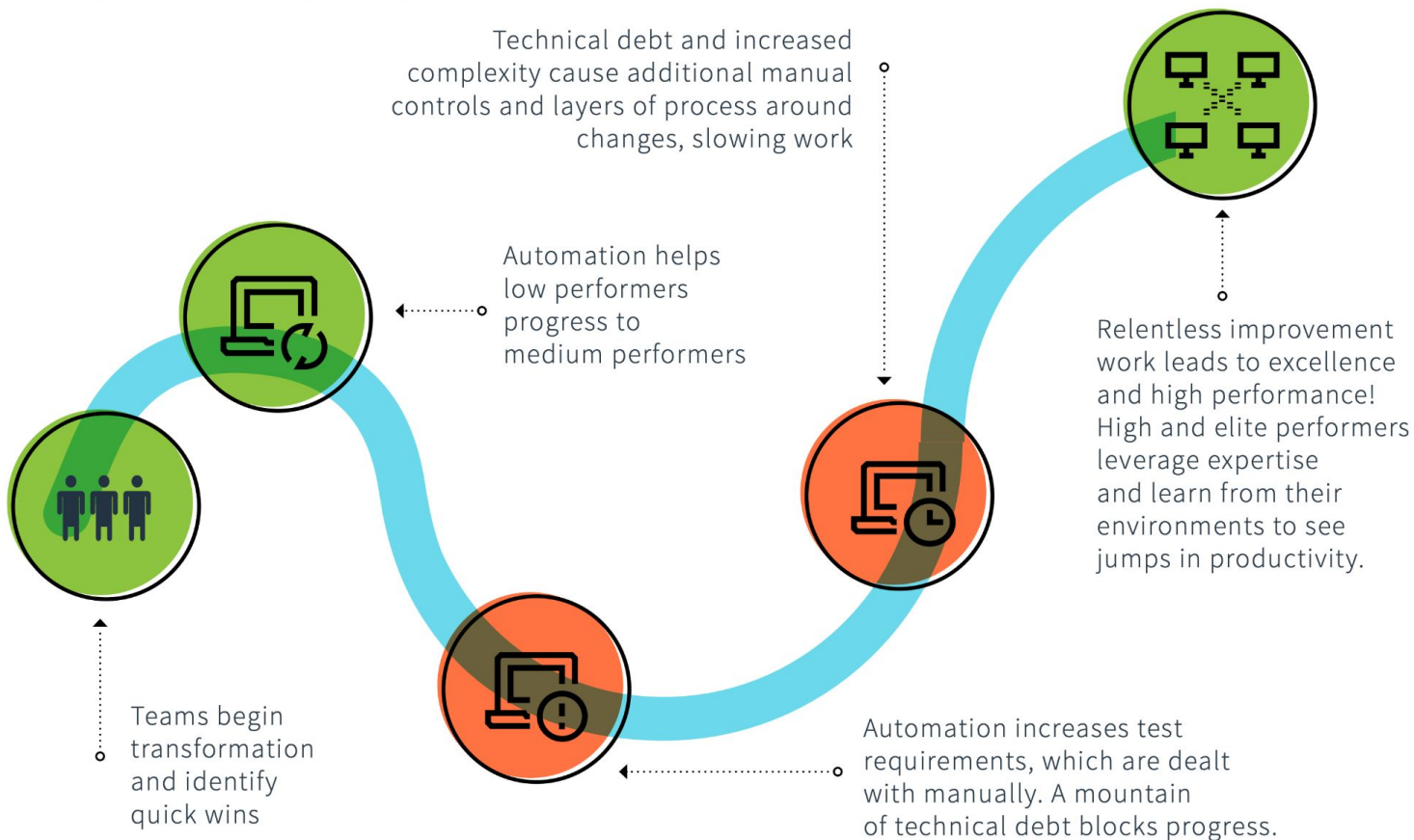
- You know what matters to your users
- Delivery is slowed or halted when error budget is gone
- Individual heroism isn't celebrated
- Funding is proactive and strategic
- Learning is incentivized (failures and successes)



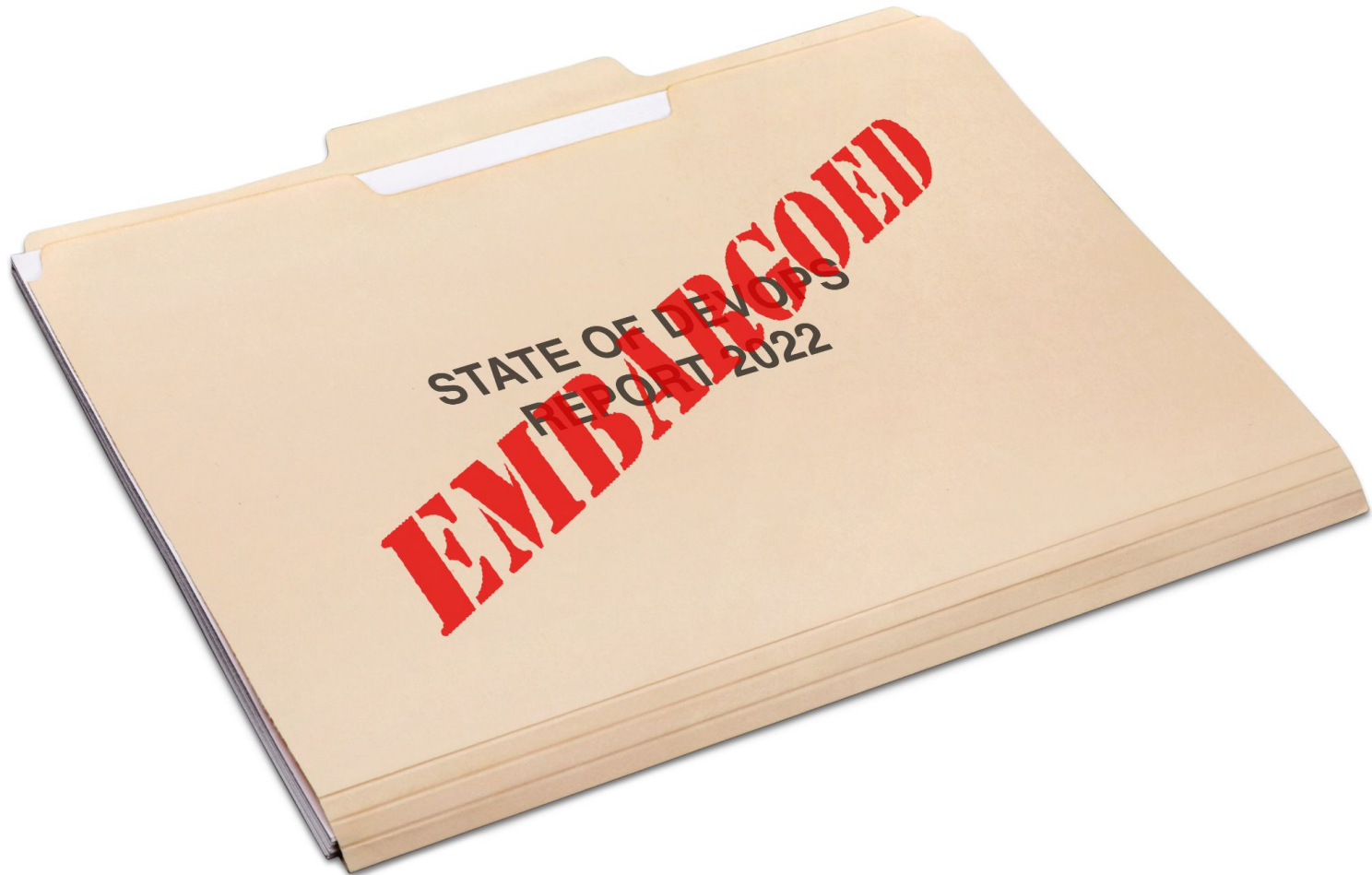
Actively Nurturing Success

- Strive for sublinear scaling
- Building and retaining sustainable teams (grow your teams organically)
- SRE is dynamic and evolves over time
- High reliability levels take (much) longer than you think
- Understand the dedicated org model isn't supposed to be a silo
- Promotion/training/compensation matches other roles (especially dev)
- Communities need water and sunlight to thrive

J-CURVE OF TRANSFORMATION



Shhhhhh!



Remember, this is science

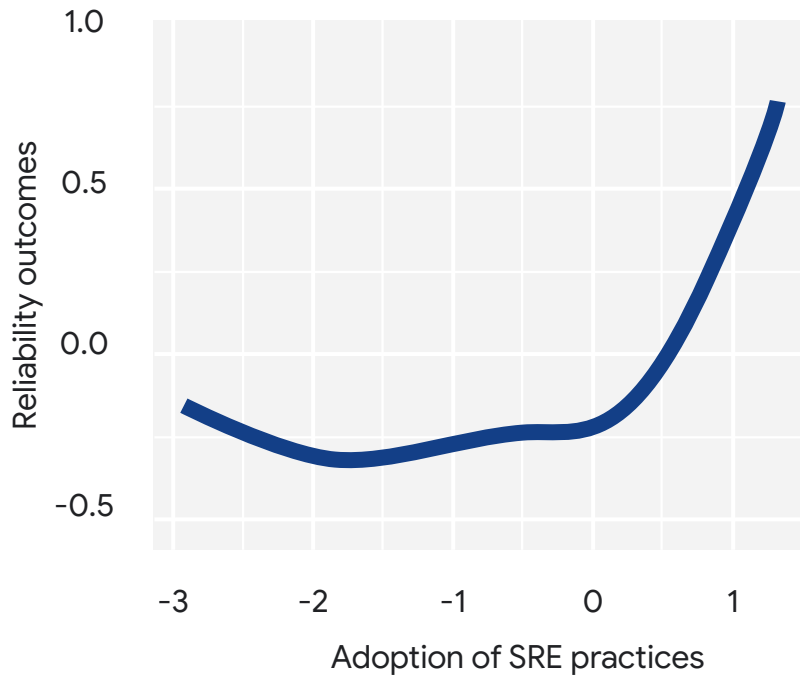
Reliability is a necessary ingredient

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.**

Generative team culture predicts better reliability

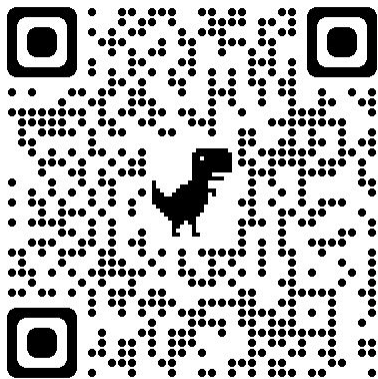
Pathological (power oriented)	Bureaucratic (rule oriented)	Generative (performance oriented)
Low cooperation	Modest cooperation	High cooperation
Messengers shot	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to inquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

You gotta stick with it



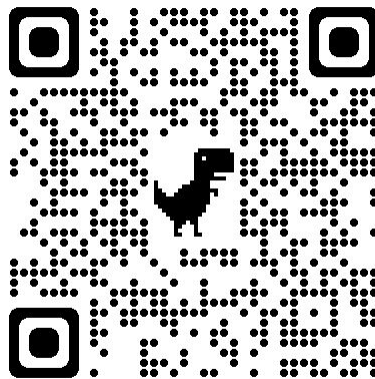
Teams that persist beyond initial steps of SRE adoption see increasing improvement in reliability outcomes.

Discuss.



bit.ly/r9y-discuss

SRE Discussion Group



dora.community

DORA Discussion Group