The Latest in DevOps: Elite Performance, Productivity, & Scaling

Dr. Nicole Forsgren & Dr. Dustin Smith

DORA
DEVOPS RESEARCH & ASSESSMENT

Google Cloud

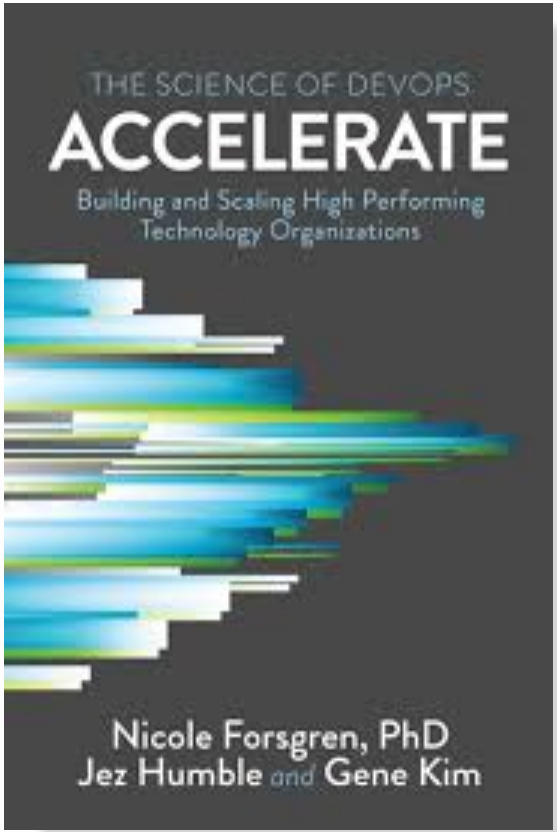
Dr. Nicole Forsgren
Research and Strategy
Google

Dr. Dustin Smith
User Experience
Researcher
Google

@nicolefv

DORA DEVOPS RESEARCH & ASSESSMENT | Google Cloud


2018
Accelerate:
State of DevOps
Strategies for a New Economy
Presented by
DORA
Diamond Sponsor
Google Cloud
Gold Sponsors
Deloitte.
XebiaLabs
pagerduty
redgate
cloudbees
Datical
sumologic
TRICENTIS
Electric Cloud
Microsoft Azure
Pivotal
aws


THE SCIENCE OF DEVOPS
ACCELERATE
Building and Scaling High Performing Technology Organizations
Nicole Forsgren, PhD
Jez Humble and Gene Kim

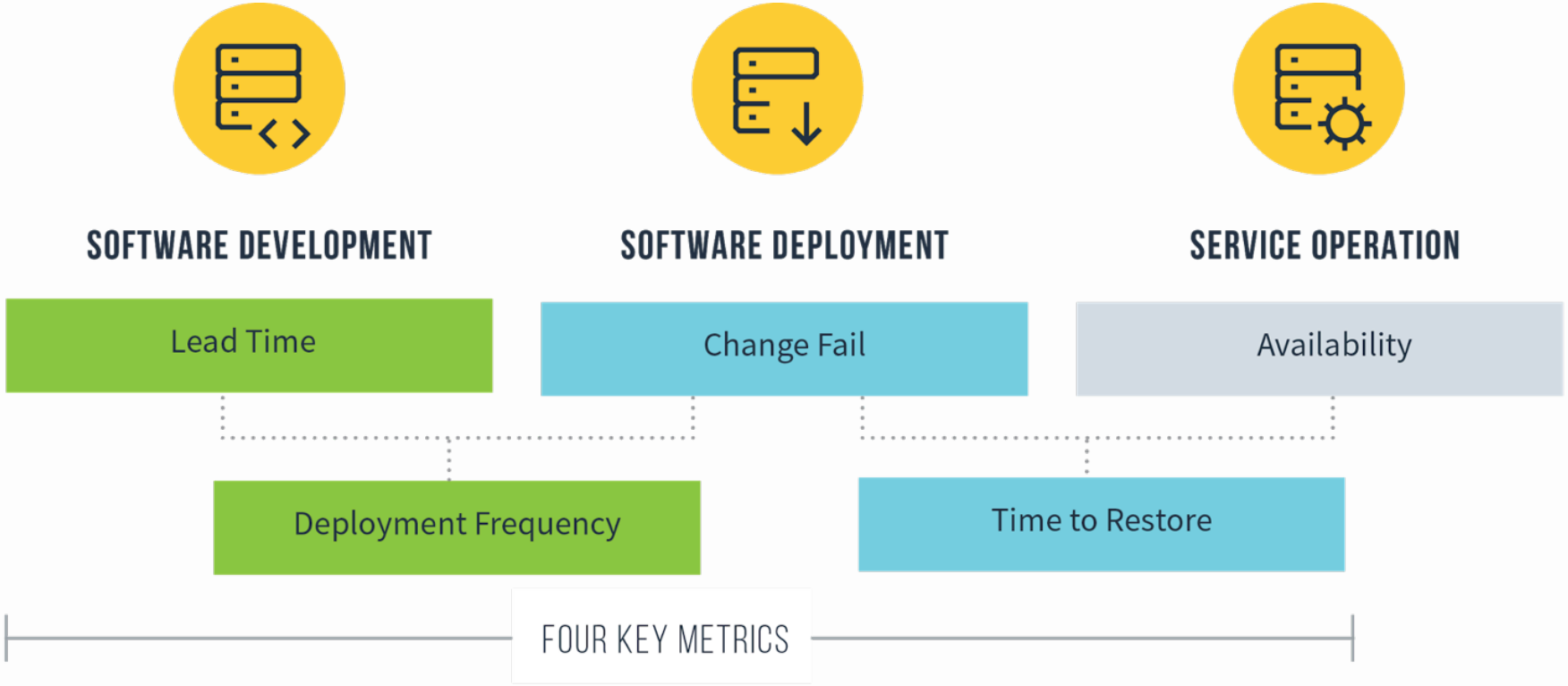

ACCELERATE
State of DevOps
2019
Sponsored by
DORA
DEVOPS RESEARCH & ASSESSMENT
Google Cloud
Pivotal
Deloitte.
sumologic
redgate
cloudbees.
armory
Datical
New Relic.

Today!

- Is this DevOps thing even “A Thing”?
- Getting better (aka Choose your own adventure)
- Performance - *speed and stability!*
- Productivity - *getting work done!*
- Culture
- Open source
- Scaling your transformation
- Fin (and more!)

DevOps?

Let’s get on the same page.



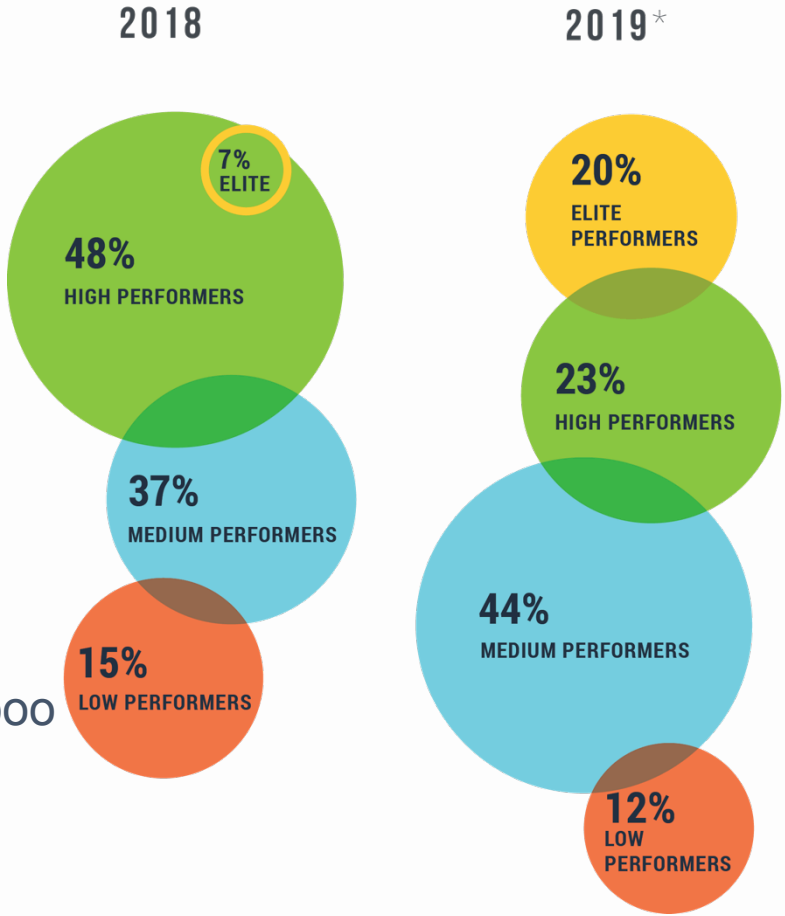
@nicolefv

DORA DEVOPS RESEARCH & ASSESSMENT | Google Cloud

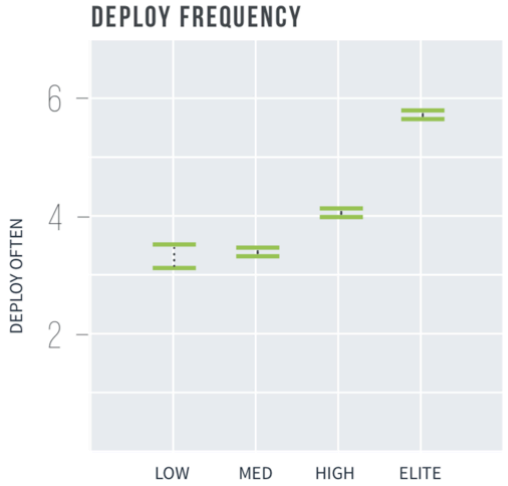
We have movement

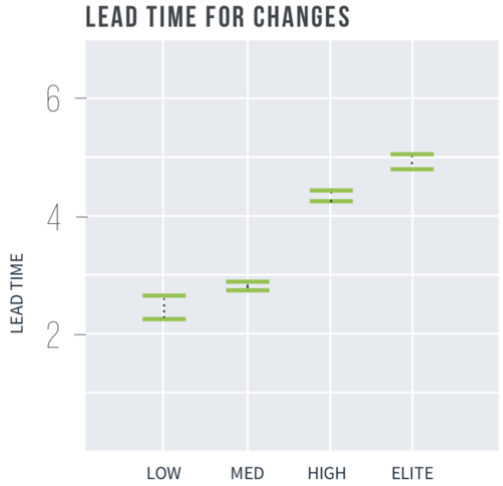
- Look at the elite performers. Yay!
- Now look at the low performers. Yay!
- Check out medium performers…
- Now look at the medium vs. high. Oooo

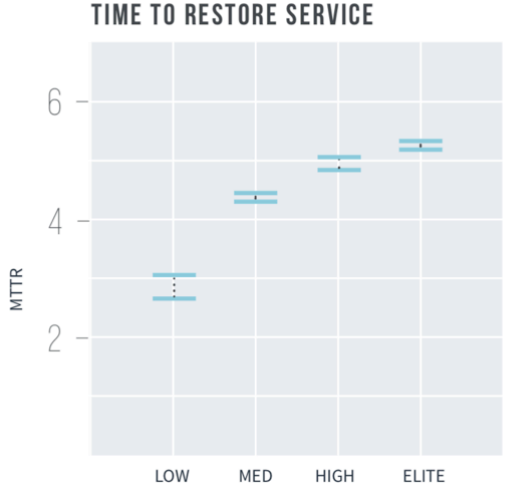
...But what does it all *mean?*

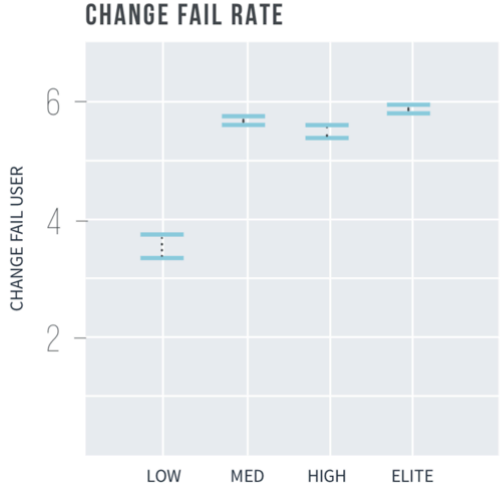


Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day[a]	Less than one day[a]	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15%[b,c]	0-15%[b,d]	0-15%[c,d]	46-60%









DEPLOY FREQUENCY:

1 = Fewer than once per six months
2 = Between once per month and once every 6 months
3 = Between once per week and once per month
4 = Between once per day and once per week
5 = Between once per hour and once per day
6 = On demand (multiple deploys per day)

LEAD TIME FOR CHANGES

1 = More than six months
2 = Between one month and six months
3 = Between one week and one month
4 = Between one day and one week
5 = Less than one day
6 = Less than one hour

TIME TO RESTORE SERVICE

1 = More than six months
2 = Between one month and six months
3 = Between one week and one month
4 = Between one day and one week
5 = Less than one day
6 = Less than one hour

CHANGE FAIL RATE

1 = 76%-100%
2 = 61%-75%
3 = 46%-60%
4 = 31%-45%
5 = 16%-30%
6 = 0%-15%

How do they compare? (And what does it all mean?)

208
TIMES MORE
frequent code deployments

106
TIMES FASTER
lead time from
commit to deploy

2,604
TIMES FASTER
time to recover from incidents

7
TIMES LOWER
change failure rate
(changes are 1/7 as likely to fail)

Availability

Availability is about promises we make and keep to our customers and end users…

The measure: How well teams

- Define their availability targets
- Track their current availability
- Learn from any outages

Availability

Availability is about promises we make and keep to our customers and end users…

The measure: How well teams

- Define their availability targets
- Track their current availability
- Learn from any outages

Day one is short, day two is ***long***

So… We do tech because it’s fun?

So… We do tech because it's fun?

yes, and...

Technology delivers value

Elite performers are

2

as likely

to meet or exceed their

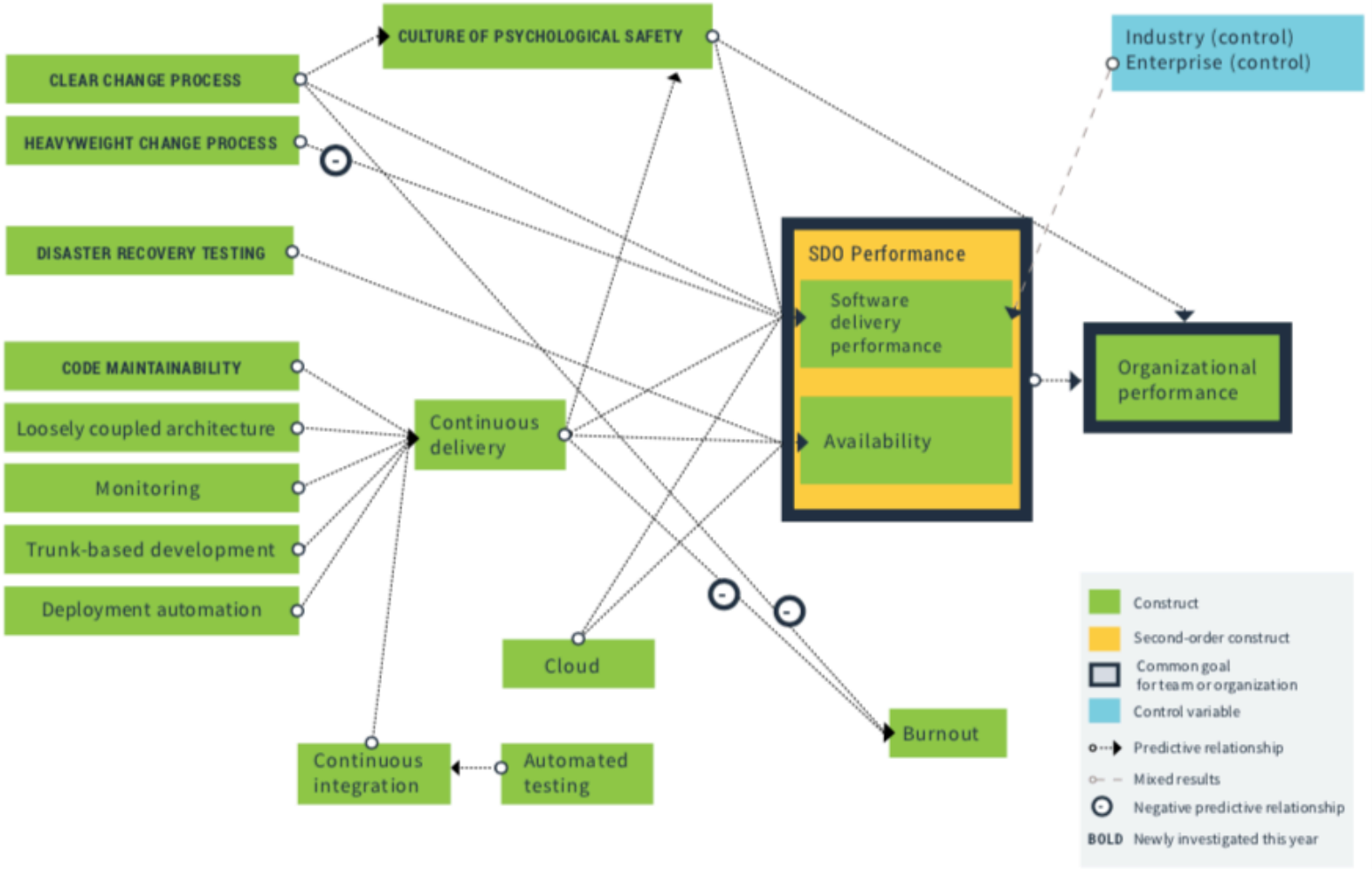
organizational performance goals

DORA DEVOPS RESEARCH & ASSESSMENT | Google Cloud

Okay… so it matters. *How do we get better?*

Okay… so it matters. *How do we get better?*

Improving Performance

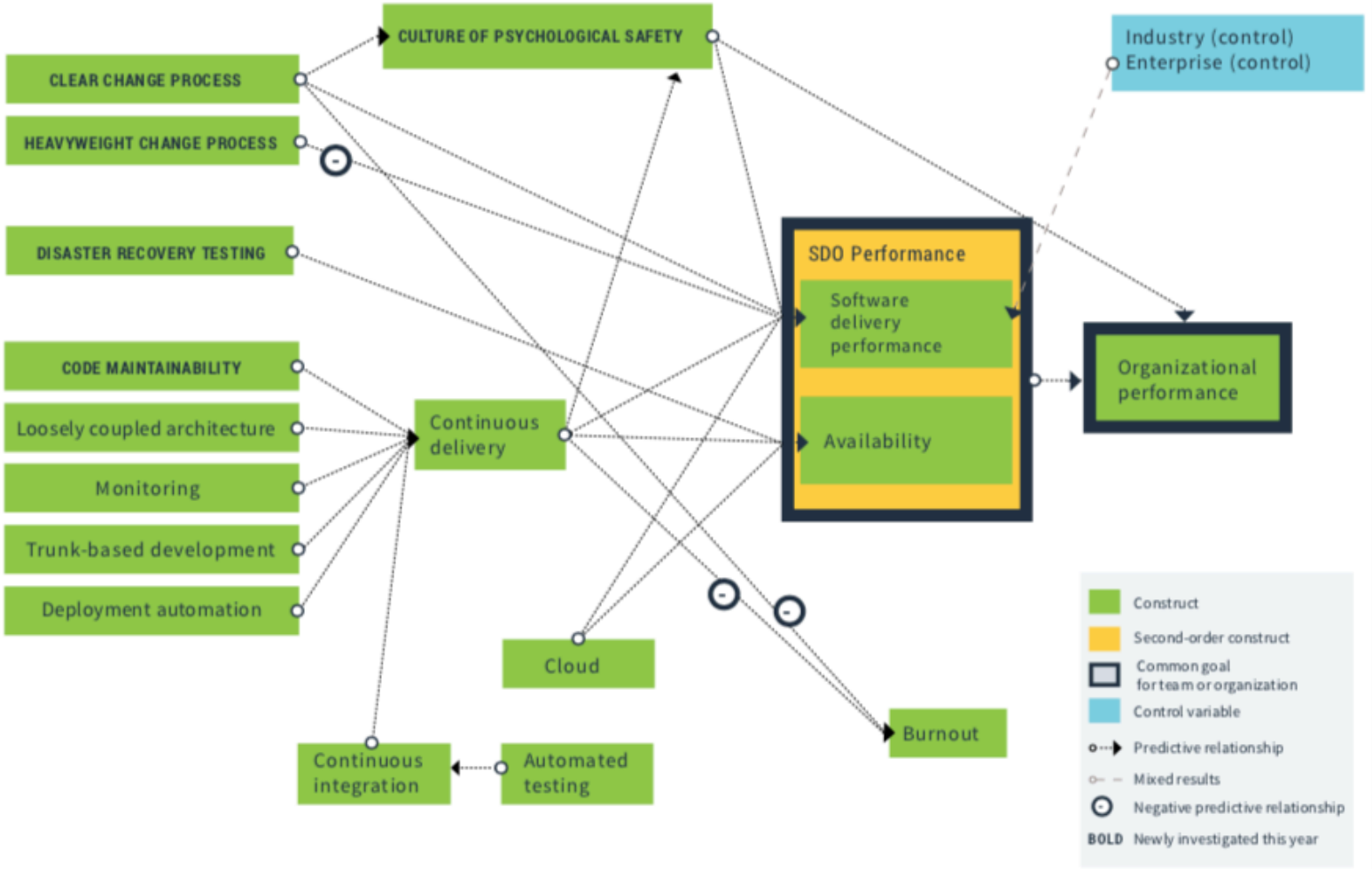


@nicolefv

DORA DEVOPS RESEARCH & ASSESSMENT | Google Cloud

girl. whoa.

Improving Performance



@nicolefv

DORA DEVOPS RESEARCH & ASSESSMENT | Google Cloud

Cloud is a differentiator for elite performance…

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

Cloud is a differentiator for elite performance…

But **only 29% of respondents** met all five characteristics of cloud computing

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

Cloud is a differentiator for elite performance…

But **only 29% of respondents** met all five characteristics of cloud computing

Elite performers were **24 times more likely** to have met cloud characteristics than low performers

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

Code maintainability

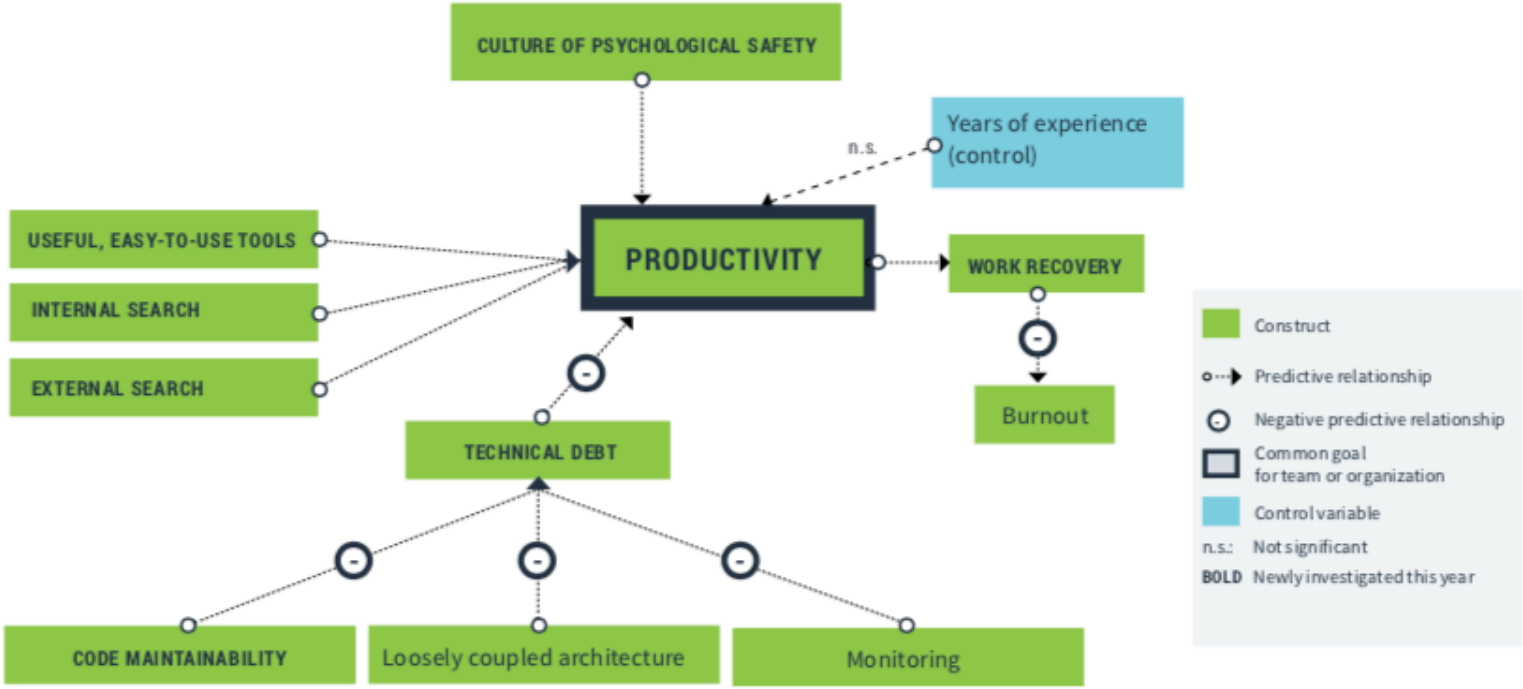
Contributes to CD and helps reduce technical debt (stay tuned!)

Systems and tools that make it easy to:

- Change code maintained by other teams
- Find code in the codebase
- Reuse other people's code
- Add, upgrade, and migrate to new versions of dependencies without breaking code

Yeah, yeah. Read the report. What *else* is new?

Improving Productivity



@nicolefv

What is productivity?

Productivity is the ability to get complex, time-consuming tasks completed with minimal distractions and interruptions

What is productivity?

Productivity is the ability to get complex, time-consuming tasks completed with minimal distractions and interruptions

This kind of productivity helps us leave work at work and reduce burnout

Technical debt was introduced in 1992 by Ward Cunningham to describe what happens when we fail to maintain “immature code”

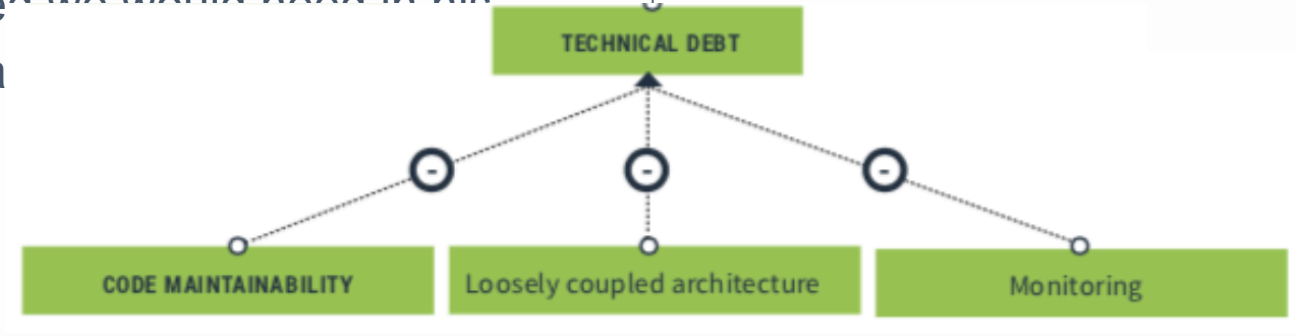
It is a problem for many of us and includes code or systems with:

- Known bugs that go unfixed in favor of new features
- Insufficient test coverage
- Problems related to low code quality or poor design
- Code or artifacts that aren’t cleaned up when no longer used
- Implementations the team doesn’t have expertise in, and therefore can’t debug or maintain
- Incomplete migration
- Obsolete technology
- Incomplete or outdated documentation or missing comments

Reducing Technical Debt

This helps us maintain a mental model of our systems, something Ward Cunningham suggested we would need in his original a

- Maintainable code
- Loosely coupled architecture
- Monitoring



Culture

I keep hearing culture matters.

*What does that even **mean?***

Westrum organizational culture

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

The Westrum organizational typology model: How organizations process information (Source: Ron Westrum, "A typology of organisation culture)," BMJ Quality & Safety 13, no. 2 (2004)

Westrum organizational culture

Pathological *(Power oriented)*	**Bureaucratic** *(Rule oriented)*	**Generative** *(Performance oriented)*
Failure leads to scapegoating	Failure leads to justice	Failure leads to inquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

The Westrum organizational typology model: How organizations process information (Source: Ron Westrum, "A typology of organisation culture)," BMJ Quality & Safety 13, no. 2 (2004)


1
PSYCHOLOGICAL SAFETY
Team members feel safe to take risks and be vulnerable in front of each other
2
DEPENDABILITY
Team members get things done on time and meet Google's high bar for excellence
3
STRUCTURE AND CLARITY
Team members have clear roles, plans, and goals
4
MEANING
Work is personally important to team members
5
IMPACT
Team members think their work matters and creates change
re:Work

Culture is important

- A culture of trust and psychological safety has a positive impact on:
  - Software delivery performance
  - Organizational performance
  - Productivity
- These results indicate teams with this culture see significant benefits in teams in many contexts

Open Source

Is open source still a thing

Community is an asset

- **High and Elite performers continue to use open source software most**
  - Elite performers **1.75 more likely to make extensive use of open source** components, libraries, and platforms than low performers (2018 ASODR)
  - This also **benefits recruiting**
  - Low performers use fully proprietary software most

- **External search contributes to productivity**
  - Engineers who used external information **1.67 times more likely to be productive**.

Faster is better - even in open source

Open source projects are community-driven, with contributors from around the world, with different skill sets, contributing when their schedule allows. What this means for open source:

- **Commiting code sooner is better.** Merging patches faster to prevent rebases helps developers move faster.

- **Working in small batches is better.** Large “patch bombs” are harder and slower to merge into a project than smaller, more readable patchsets since maintainers need more time to review the changes.

Scaling the DevOps

But I have a whole ORGANIZATION TO CHANGE

Executing for maximum impact

- **Start with foundations**
- **Leverage concurrent efforts for maximum impact**
- **Focus on constraints to accelerate your transformation**

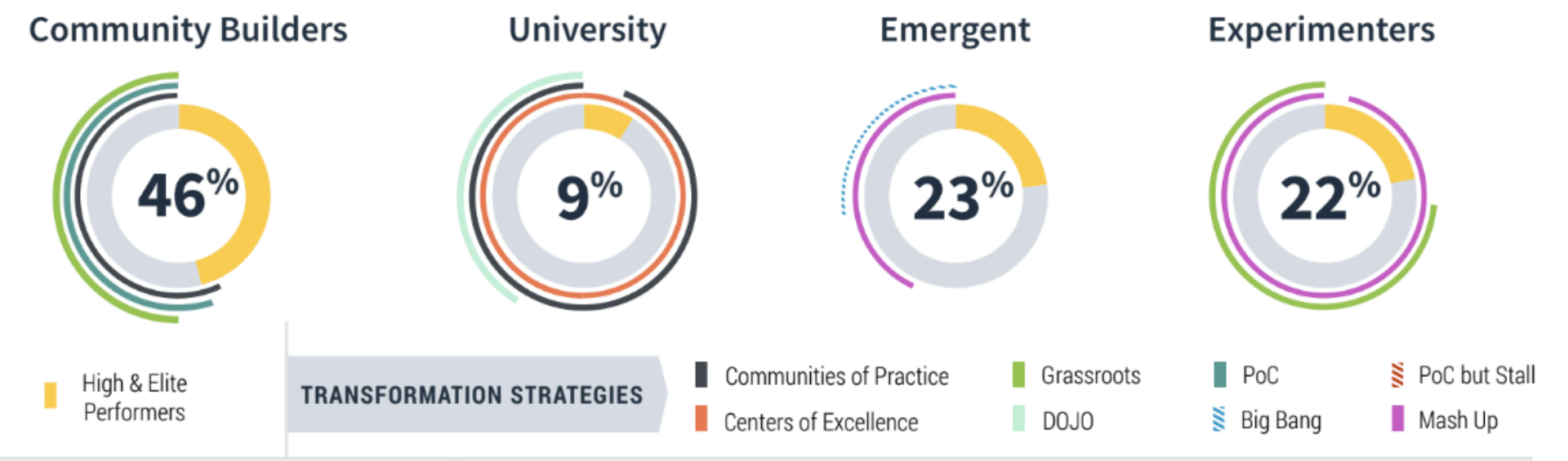
CAPABILITIES RESEARCHED IN 2019[16]	
Organization level	• Loosely coupled architecture • Clear change process • Code maintainability
Team level	• Continuous integration • Automated testing • Deployment automation • Monitoring • Trunk-based development
Both team and organizational level	• Use of cloud services • Disaster recovery testing

Scaling Your Transformation - Overall Patterns

HEATMAP OF DEVOPS TRANSFORMATION STRATEGIES BY PERFORMANCE PROFILE

	Low	Medium	High	Elite
Training Center	27%	21%	18%	14%
Center of Excellence	34%	34%	20%	24%
Proof of Concept but Stall	41%	32%	20%	16%
Proof of Concept as a Template	16%	29%	29%	30%
Proof of Concept as a Seed	21%	24%	29%	30%
Communities of Practice	24%	51%	47%	57%
Big Bang	19%	19%	11%	9%
Bottom-up or Grassroots	29%	39%	46%	46%
Mashup	46%	42%	34%	38%

Scaling Your Transformation - High and Elite Performers



TL;DR

- Is this DevOps thing even “A Thing”?
- Getting better (aka Choose your own adventure)
- Performance
- Productivity
- Culture
- Open source
- Scaling your transformation
- Fin! (there’s so much more!)
  - Disaster Recovery Testing
  - Change Approvals (the “right” way)
  - More open source!
  - More cloud (and costs!)
  - Executive summary (coming soon!)

Go check it out! cloud.google.com/devops

Thank you!

DORA
DEVOPS RESEARCH & ASSESSMENT

Google Cloud