

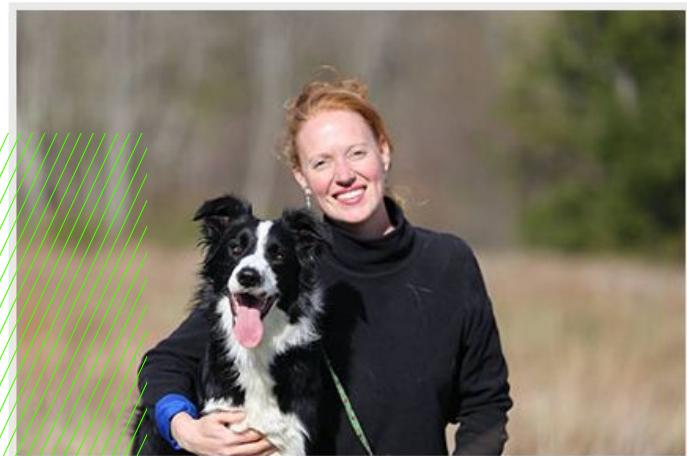
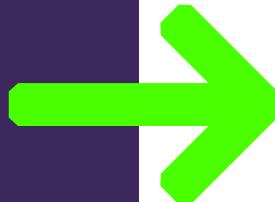
# AGILE NUMBERS —

The Signposts on Your Agile Journey

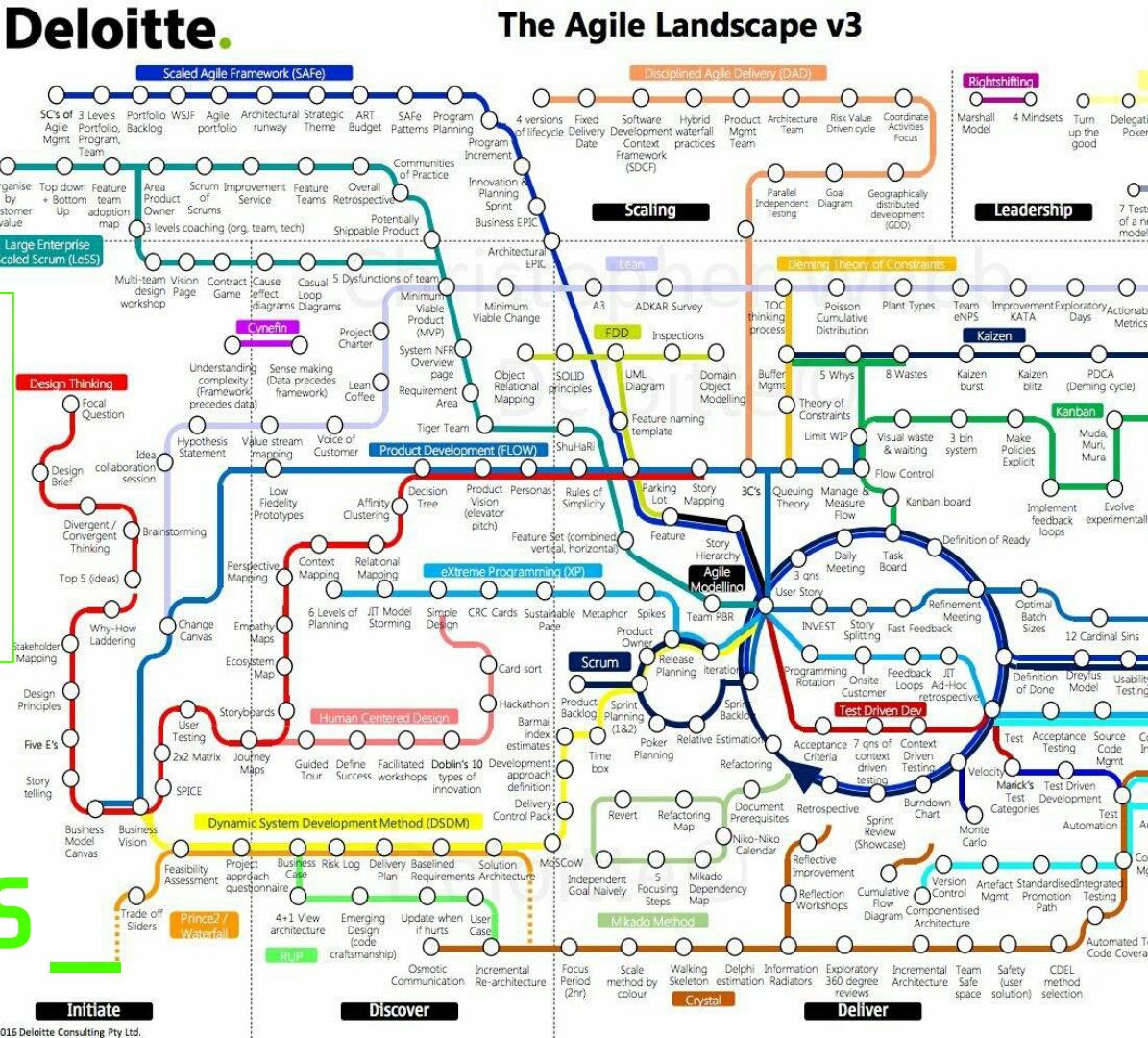
By Akrem Saed



The **name**  
that I  
really  
wanted\_



# Current State of Agile Practices



## Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck  
Mike Beedle  
Arie van Bennekum  
Alistair Cockburn  
Ward Cunningham

James Grenning  
Jim Highsmith  
Andrew Hunt  
Ron Jeffries  
Jon Koske

Robert C. Martin  
Steve Mellor  
Ken Schwaber  
Jeff Sutherland  
Dawn Thomas

# Agile didn't start as a

[agilemanifesto.org](http://agilemanifesto.org)

# "landscape"

# Manifesto for Agile Software Development



We are **uncovering better ways** of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools  
Working software over comprehensive documentation  
Customer collaboration over contract negotiation  
Responding to change over following a plan

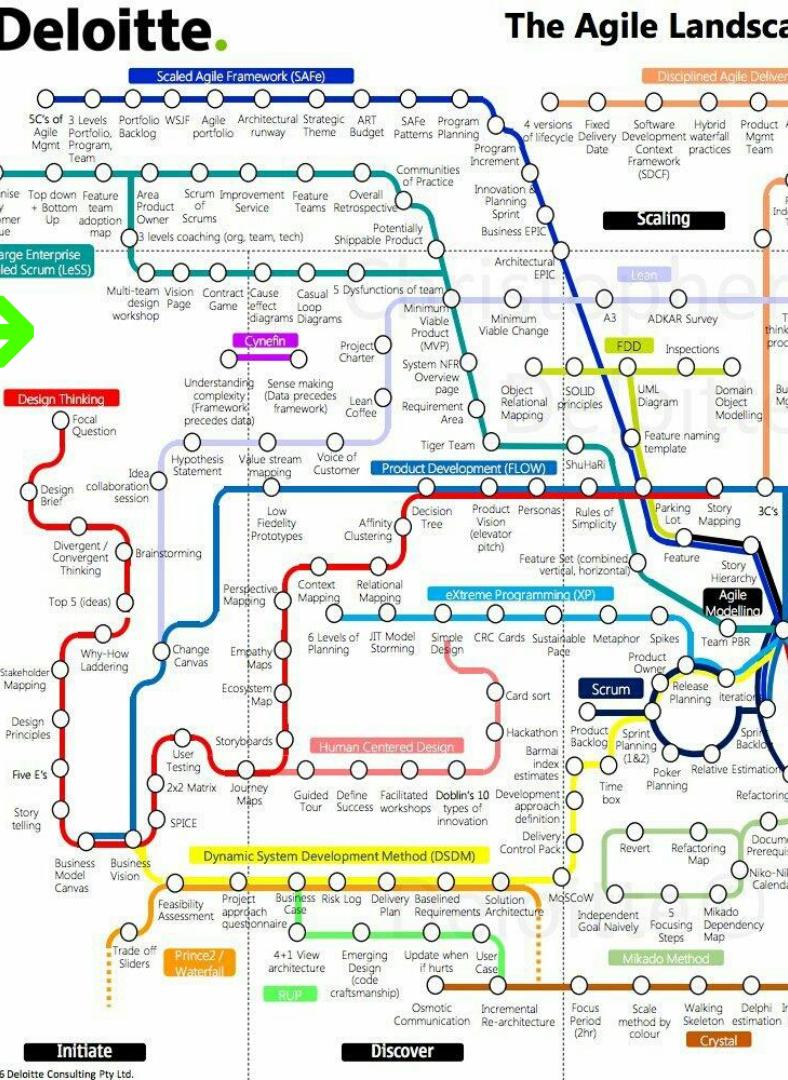
That is, while there is value in the items on

## Why Data? —

[agilemanifesto.org](http://agilemanifesto.org)

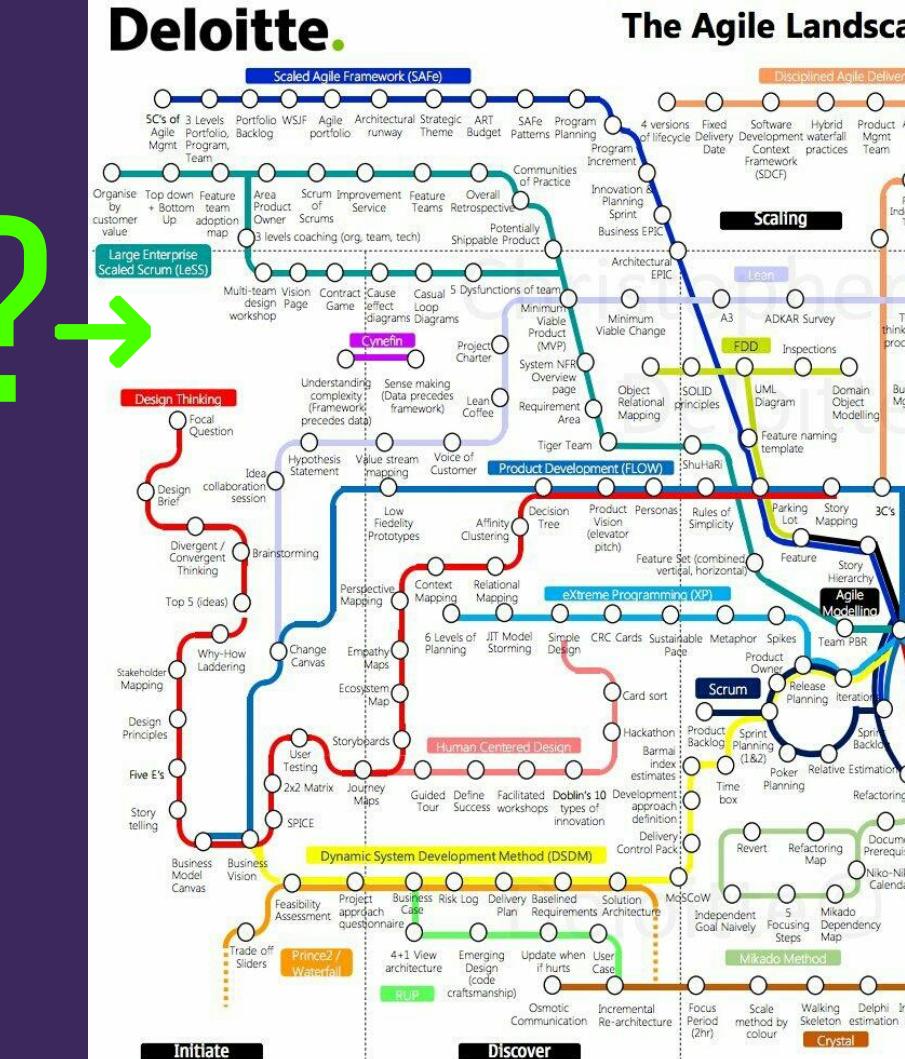
# Good Questions\_?

- Which one is more popular?
- Which one is endorsed by my agile coach?
- Which one did Netflix use?
- Which one can scale to the size of my organization?
- Which one is easy to pick up?



# Better Questions -

## Which ones have data behind them?



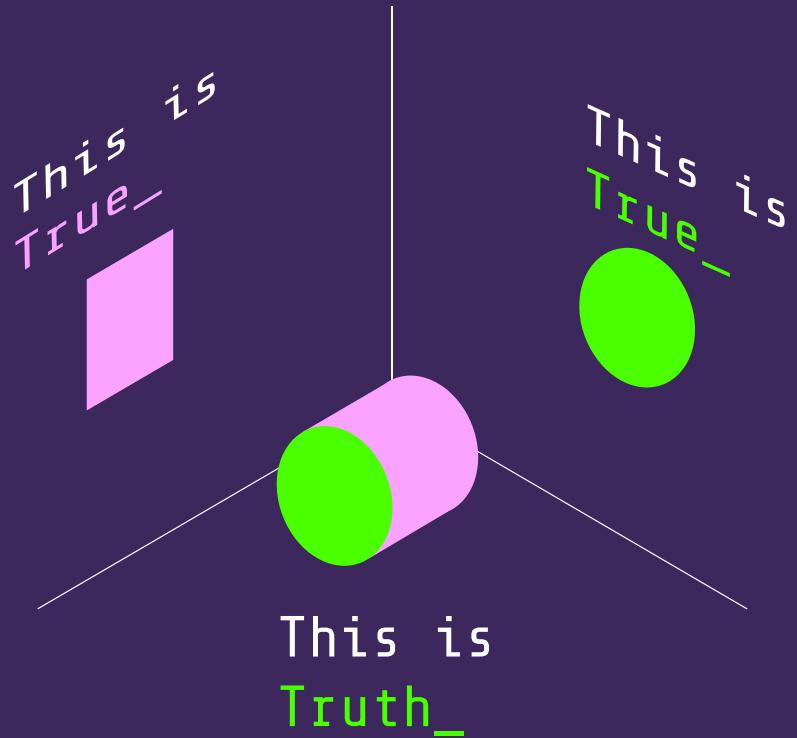


Current  
State\_  
of Discourse



# Subjective Measures\_

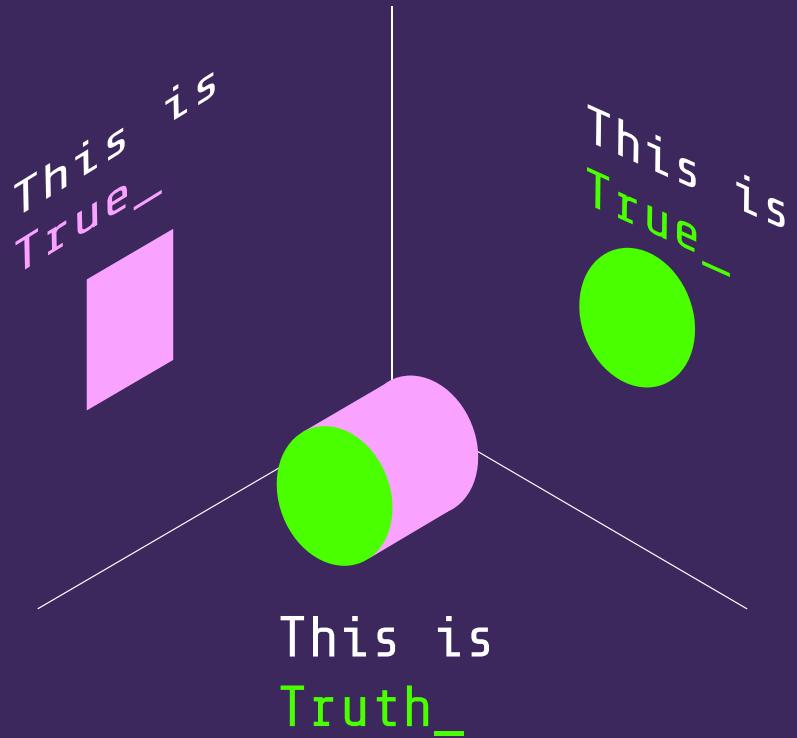
- “Google has been using it for years”
- “I’ve used it in a past project and worked wonders”
- “Our competitors are using it”
- “I’ve read this blog post”
- “It’s trending on Twitter/publications/conferences”
- “It made sense to me!”



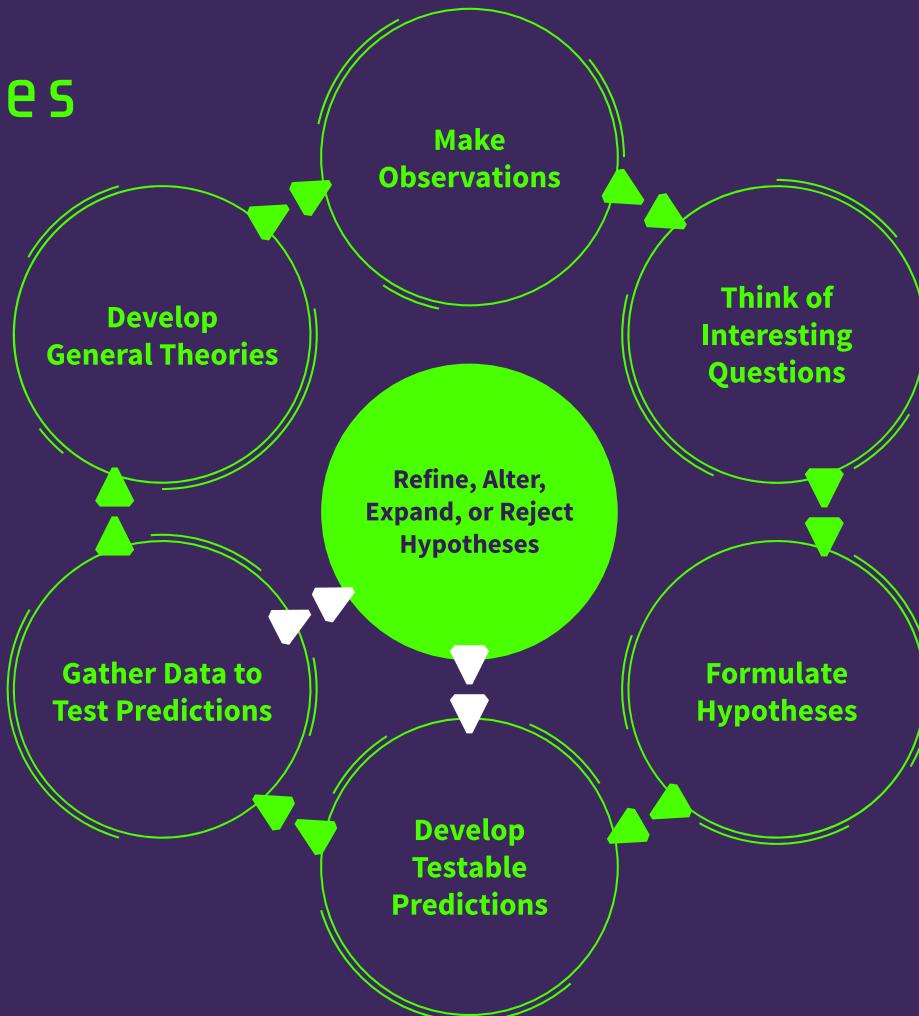
# Objective Measures\_

(a.k.a Agile Numbers)

- Controlled Scientific Studies
- Statistical Analysis of IT Behavior
- Comparative Case Studies
- Large-Scale Experiments
- Small-Scale Experiments



# Scientific Studies & Research\_



# What We Know About Software Engineering

The image shows a video player interface. On the left is a slide titled "Oh, the Irony" about James Lind's 1747 experiment. On the right is a video frame of Greg Wilson speaking at a podium.

**“Oh, the Irony”**

James Lind (1716-94)

1747: (possibly) the first-ever controlled medical experiment

✗ cider	✗ sea water
✗ sulfuric acid	✓ oranges
✗ vinegar	✗ barley water

No-one paid attention until a proper Englishman repeated the experiment in 1794...

“Bits of Evidence:  
What We Actually Know About Software Development, and Why We Believe It’s True”

- Greg Wilson, January 2010

<https://vimeo.com/9270320>

# Standard Of Proof\_



“ ”

“.. the debate is hampered because not enough people know how to develop DSLs effectively”

- **Martin Fowler, 2009**



“ ”

“One of the smartest guys in our industry made two substantive claims in an academic journal without a single citation ..... I think the debate is hampered by low standards of proof”

- **Greg Wilson, 2010**

# Raise the Standard: Studies\_

## Anchoring and Adjustment in Software Estimation

(Aranda and Easterbrook 2005)



### Finding:

the anchor mattered more than experience,  
how formal the estimation method was,  
or anything else.

# Raise the Standard: Studies\_

The two biggest causes of project failure are poor estimation and unstable requirements.

(**van Genuchten 1991 and many others**)

For every 25% increase in problem complexity, there is a 100% increase in solution complexity.

(**Woodfield, 1979**)

If more than 20-25% of a component has to be revised, it's better to rewrite it from scratch.

(**Thomas et al, 1997**)

# Raise the Standard: Studies\_

Rigorous inspections can remove 60-90% of errors before the first test is run.

(Fagan 1975)

Physical distance does not affect post-release fault rates. Distance in the organization chart does.

(Nagappan et al (2007) & Bird et al (2009))

# Want more studies? \_

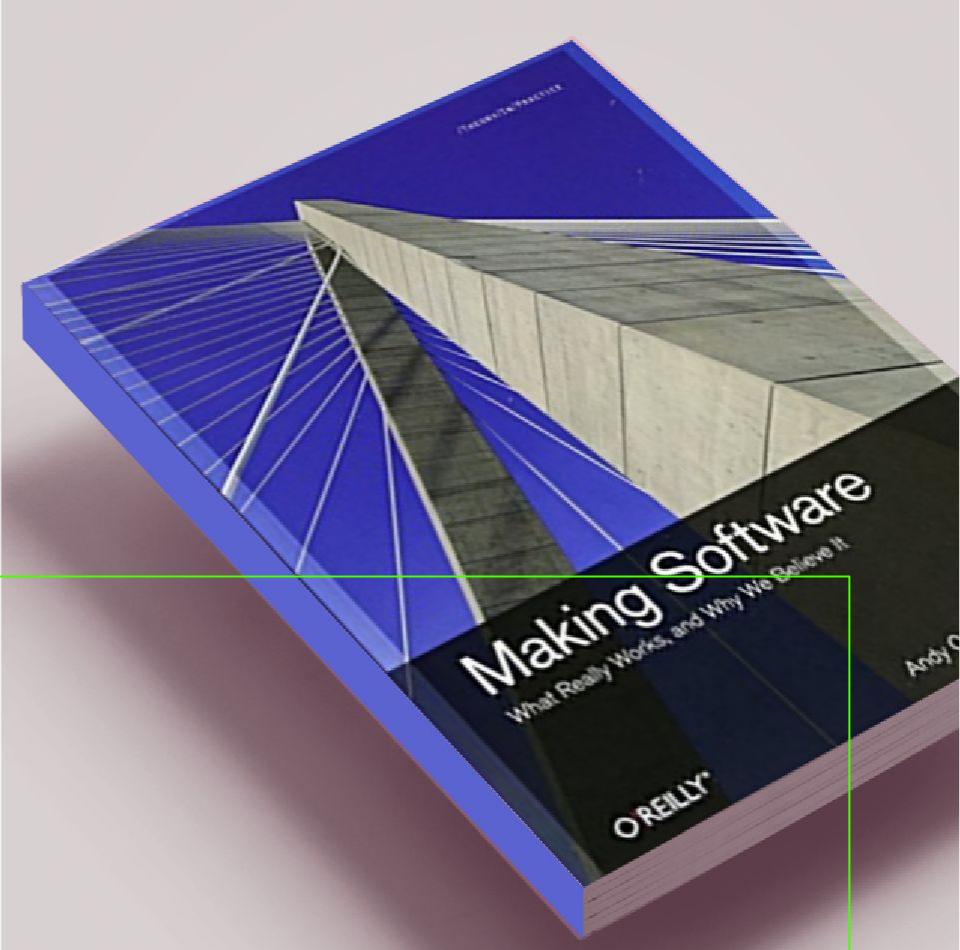


Making Software: What Really Works, and Why We Believe it

By Andy Oram & Greg Wilson

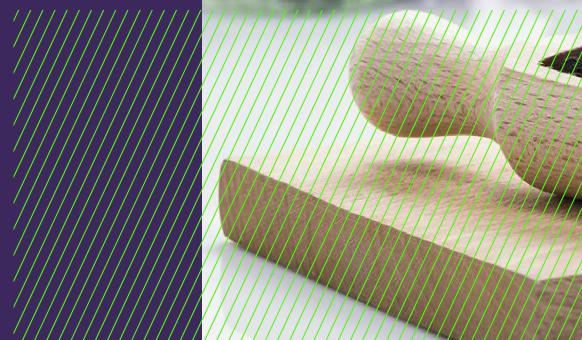


<https://goo.gl/aPUzGo>



# Folk Medicine in IT\_

Study What Already  
Works in the Wild





# Project Aristotle\_

Quest to Build  
The Perfect Team

A black and white photograph of the rock band The Who. The four members—Roger Daltrey, Pete Townshend, John Entwistle, and Keith Moon—are wrapped in a large Union Jack flag. The band's name, "THE WHO", is printed in a bold, sans-serif font at the top left of the image.

# The Who

## Aristotle: Test the commonly-held myths\_

“... there was nothing showing that a mix of specific personality types or skills or backgrounds made any difference. **The ‘who’ part of the equation didn’t seem to matter.**”

- Abeer Dubey, leader of Project Aristotle



New York Times > <https://goo.gl/z1jjac>



# Aristotle: The Safety Hint\_

What distinguished the “**good**” teams from the dysfunctional groups was **how teammates treated one another.**



Evidence for a Collective Intelligence Factor in the Performance of Human Groups

Woolley et al. (2010)



# Aristotle: The Safety Hint\_

1

## Equal Talk Time

“As long as everyone got a chance to talk, the team did well, But if only one person or a small group spoke all the time, the collective intelligence declined.”

- Anita Williams Woolley



2

## Aristotle: The Safety Hint\_

### **High average social sensitivity**

Teammates were skilled at figuring out how others felt based on their tone of voice, expressions and other nonverbal cues.



# Aristotle: 1+1 Psychological Safety\_

“[Psychological safety is] a sense of confidence that the team will not embarrass, reject or punish someone for speaking up,”

- **Amy Edmondson (1999)**



TEDx Talk > <https://goo.gl/eXbxXn>



## Aristotle: The Finding\_

“There were other behaviors that seemed important as well — like making sure teams had clear goals and creating a culture of dependability. But Google’s data indicated that **psychological safety, more than anything else, was critical to making a team work.**”

# More About Project Aristotle



The laptop screen shows the re:Work website, which provides a summary of the 5 key findings from the research:

- 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 & Clarity**  
Team members have clear roles, plans, and goals.
- 4 Meaning**  
Work is personally important to team members.
- 5 Impact**  
Team members feel their work matters and creates change.

Below the findings, there is a section titled "Introduction" and a note about the research being published in the New York Times Magazine.



New York Times > <https://goo.gl/z1jac>



Google re:Work > <https://goo.gl/19k4u7>

2017

# State of DevOps Report

Presented by:



Sponsored by:



## Surveys & Statistical Models\_



# DevOps Report: The Data\_



27,000

Survey responses



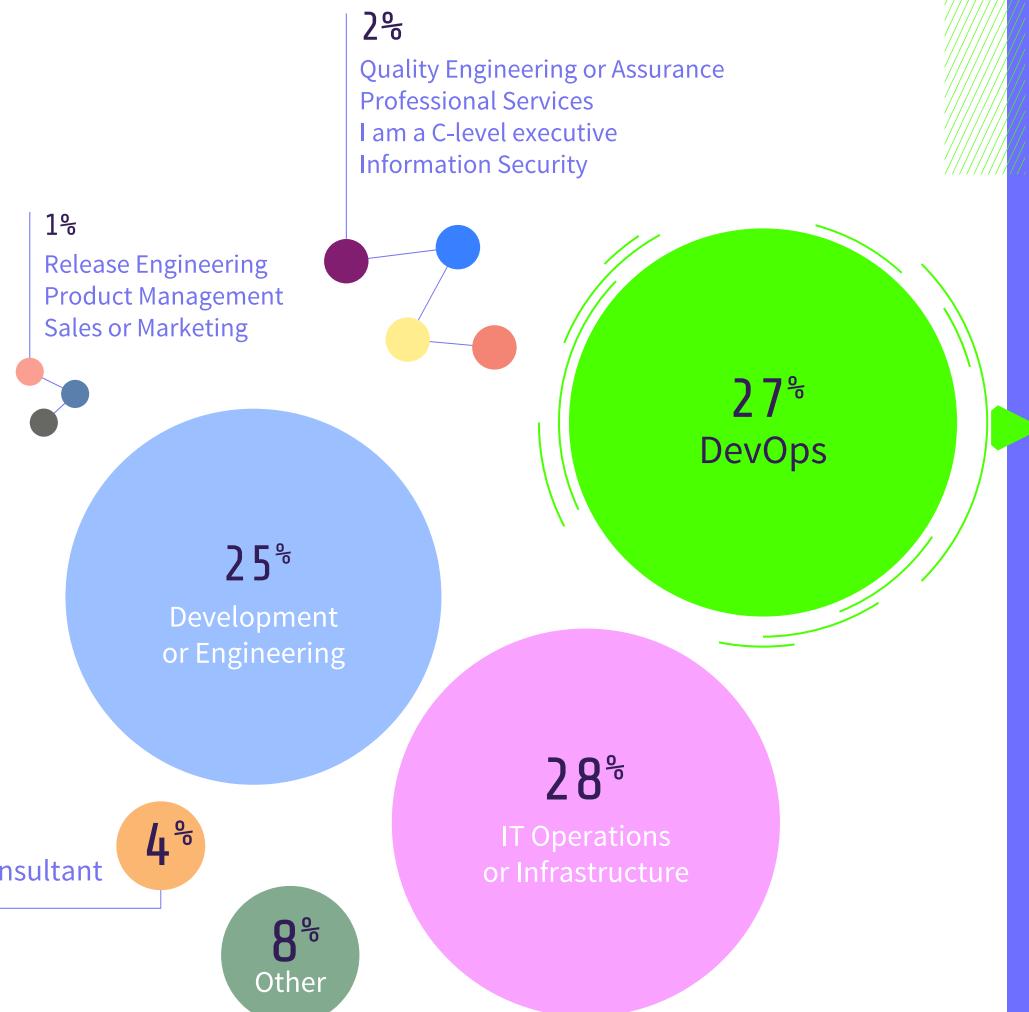
3,200

People



6

Years



## DevOps Report: Diverse Data Set\_

### Demographics

DevOps teams increased

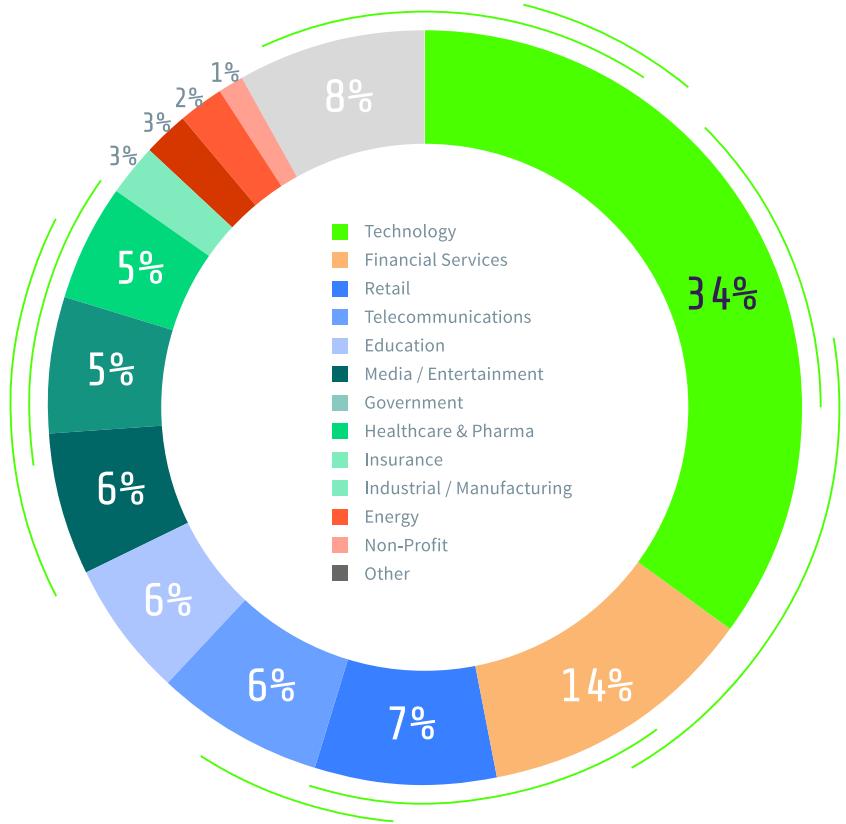
- from 16% in 2014
- to 19 % in 2015
- to 22% in 2016
- to 27% in 2017.

DevOps Report 2017 > <https://goo.gl/yqTHvz>



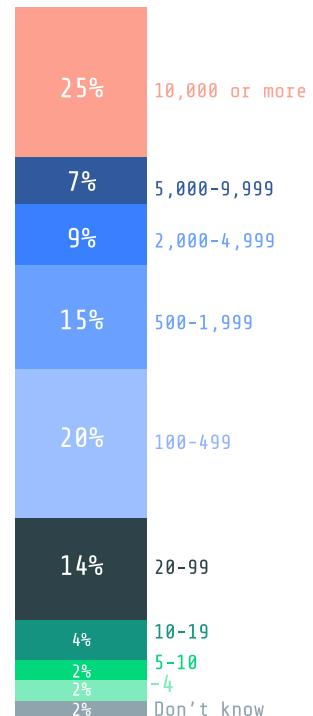
## Industry

---



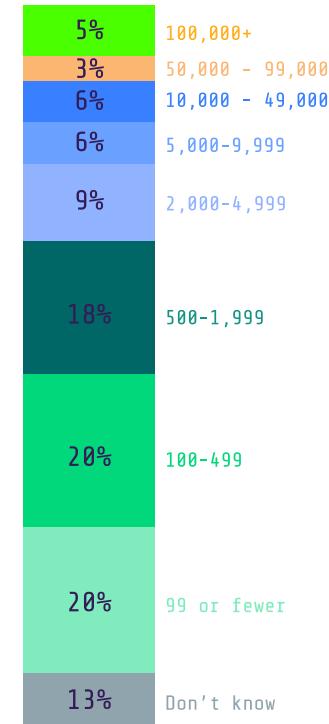
## Number of employees

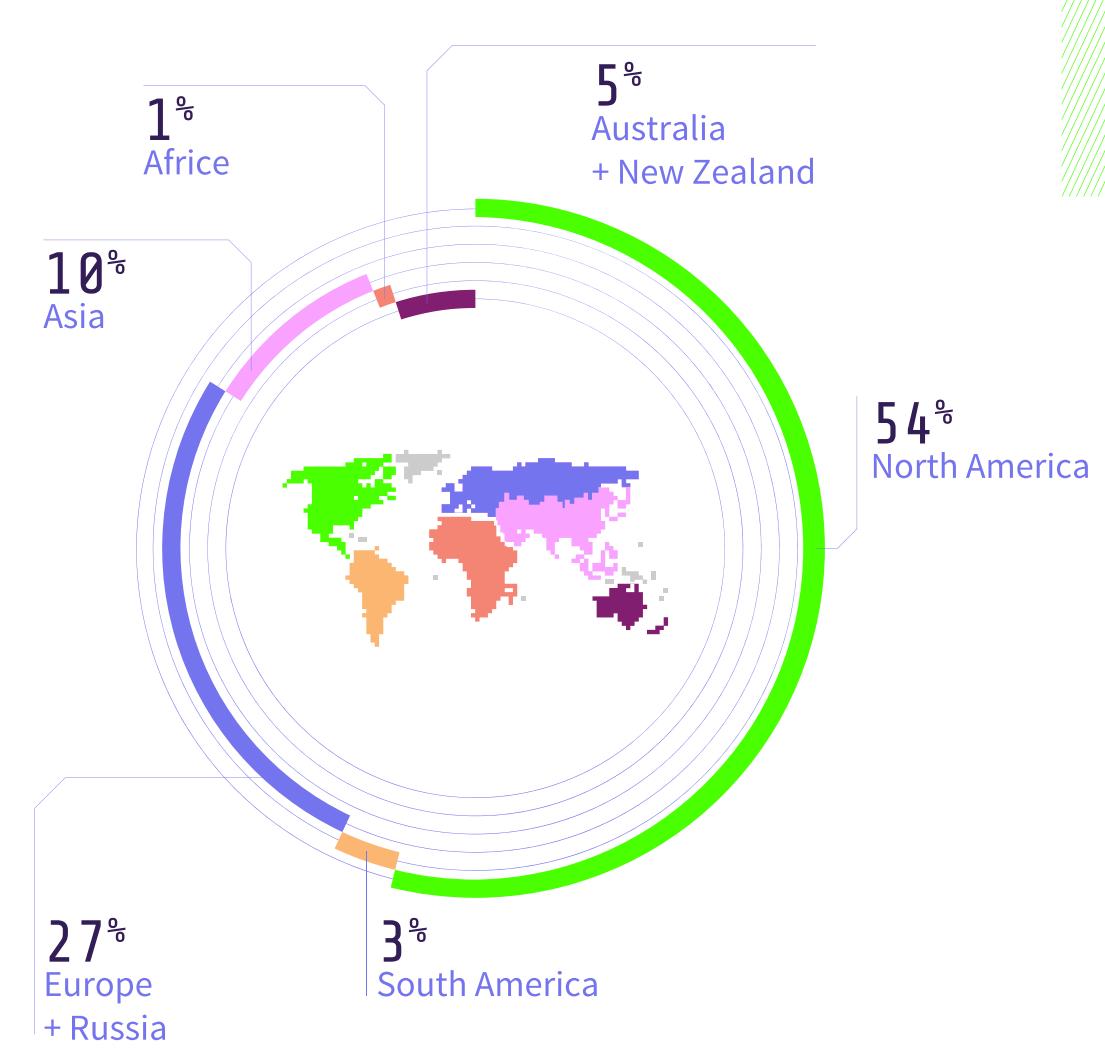
---



## Number of servers

---





DevOps Report:  
Diverse Data Set\_

## Respondent by Regions



A speedometer with a digital display showing 120 km/h. The needle is red and points to the 120 mark. The background is dark with green diagonal stripes on the left side.

# DevOps Report: IT Performance\_



# DevOps Report: IT Performance Gap

Throughput gap ↓

Stability gap ↑

High  
Performing Teams

Vs  
Low  
Performing Teams



# DevOps Report: IT Performance Gap

IT Performance Metrics	2016	2017
Deployment frequency	200x more frequent	<b>46x more frequent</b>
Lead time for changes	2,555x faster	<b>440x faster</b>
Mean time to recover (MMTR)	24x faster	<b>96x faster</b>
Change failure rate	3x lower (1/3 as likely)	<b>5x lower (1/5 as likely)</b>

Table 1: Changes in IT performance of high performers, 2016 to 2017



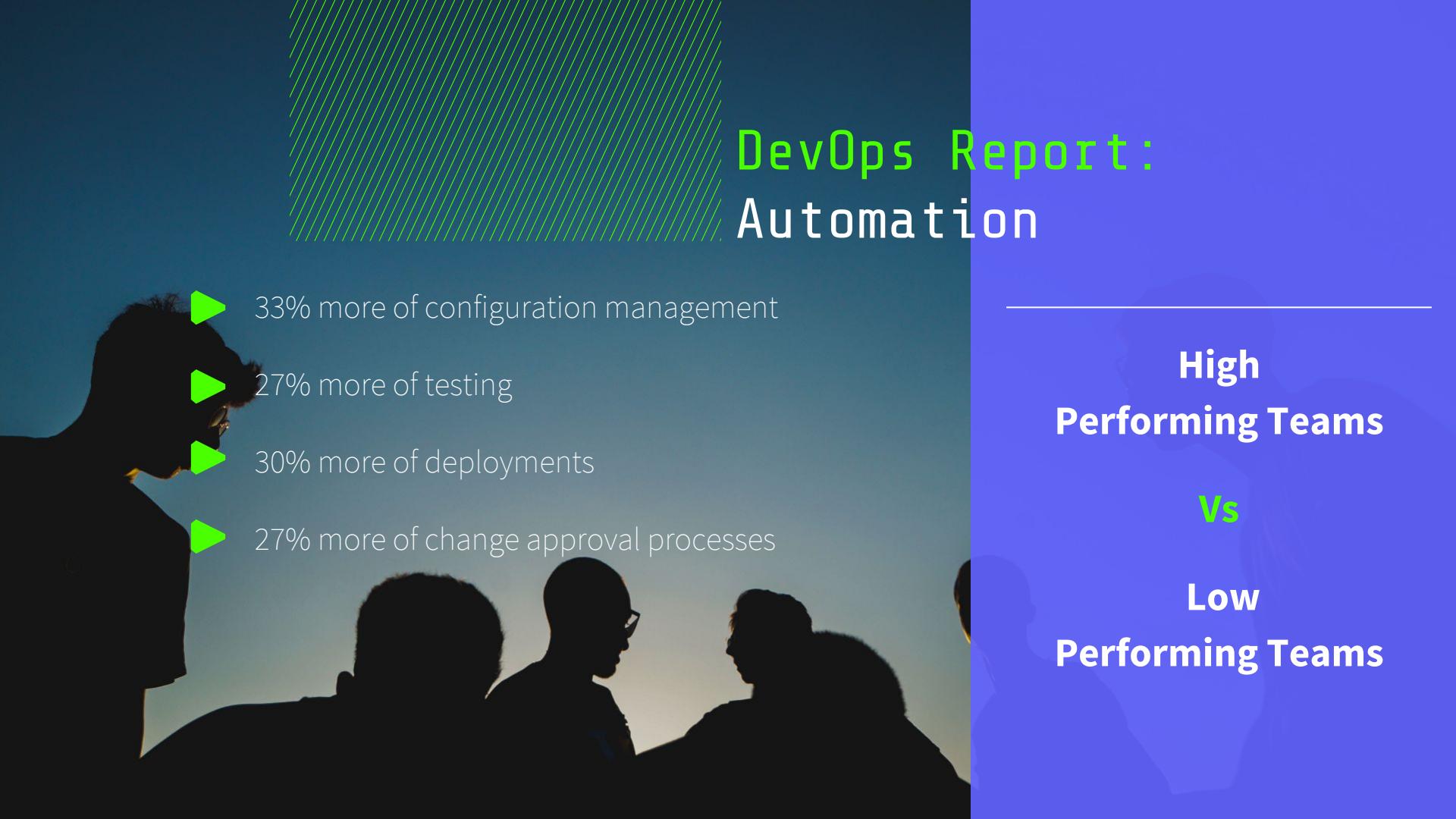
# DevOps Report: IT Performance Gap

The High  
Performing Teams'  
Advantages



# DevOps Report: Technical Practices\_





# DevOps Report: Automation

- ▶ 33% more of configuration management
- ▶ 27% more of testing
- ▶ 30% more of deployments
- ▶ 27% more of change approval processes

**High  
Performing Teams**

**vs**

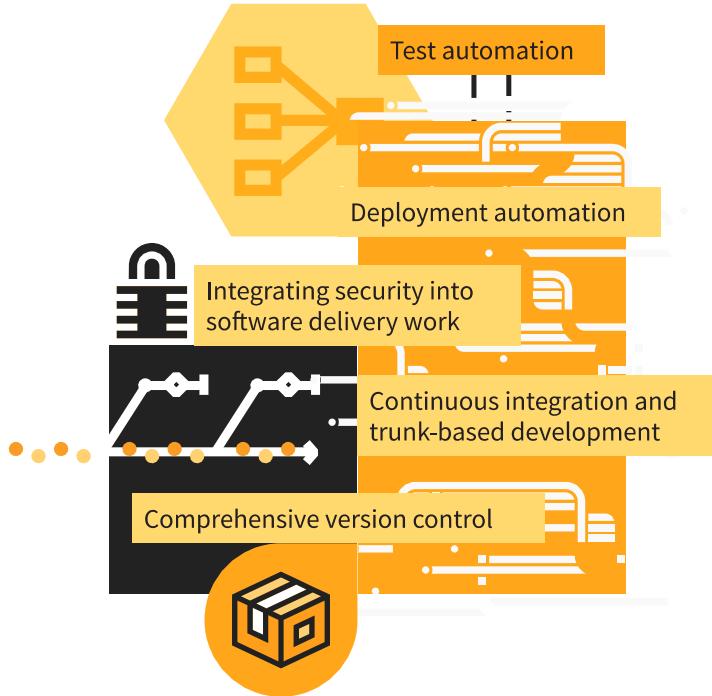
**Low  
Performing Teams**

# DevOps Report: Automation

- ▶ Medium- were doing more manual work than low-performing teams in



### Factors that positively contribute to continuous delivery:



# DevOps Report: Continuous Delivery

*“Continuous Delivery significantly contributes to both lower deployment pain and higher IT performance”*

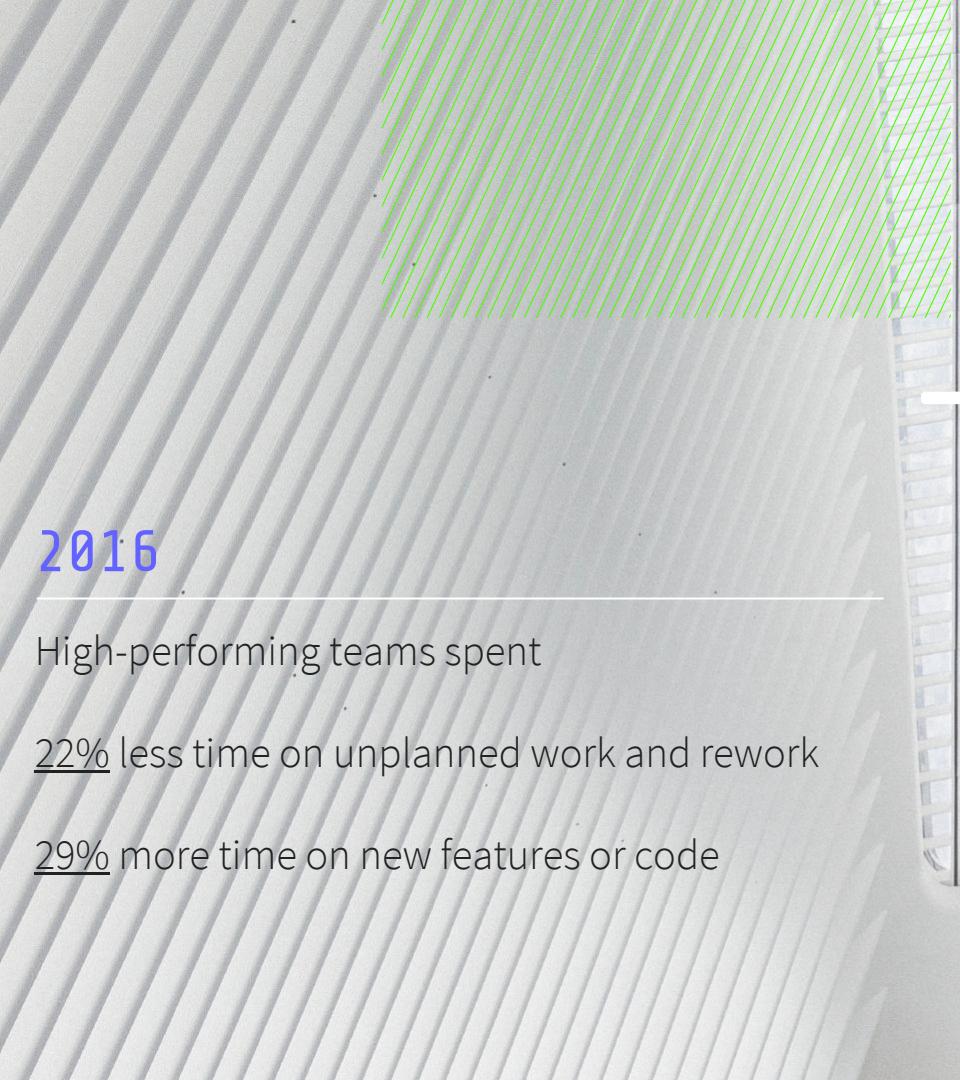


# DevOps Report: Architecture

- ▶ In 2015, **high-performing teams** were more likely to have **loosely coupled architecture**.
- ▶ In 2017,
  - **Loosely Coupled Teams** contribute to better IT performance.
  - **Loosely Coupled Architecture** derives IT performance.
    - Bounded Contexts (Domain Driven Design)
    - APIs
    - Test doubles for testing

# DevOps Report: Continuous Delivery & Teams

- 
- Biggest contributor to Continuous Delivery (bigger than test and deployment automation) is if a team can do:
- Make design changes without permission or dependency
  - Complete its work without involving other teams
  - Deploy and release on demand
  - Do most of testing on demand without integrated test environment
  - Deploy during business hours with negligible downtime



2016

High-performing teams spent

22% less time on unplanned work and rework

29% more time on new features or code

## DevOps Report: Quality



2017

High-performing teams spent

**21%** less time on unplanned work and rework

**44%** more time on new features or code

In 2016, higher software delivery performance contributed by:

Merging code into trunk on daily basis

Having short-lived branches and forks (less than 1 day)

Having fewer than three active branches

## DevOps Report: Trunk-Based Dvlpmnt.



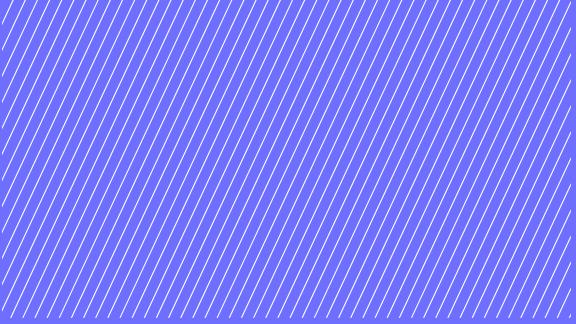
In 2017, report analyzed how long branches live and time to integrate branches into trunk

High-performing teams spend only **hours** on both measures

Low-performing teams spend **days**



# DevOps Report: Lean Product Management\_



# DevOps Report: Lean Product Management\_

In 2016

---

lean product management practices predicted:

- Higher IT performance
  - Lower deployment pain
- 

In 2017

---

the report found IT performance predicts lean product management practices

---

→ 2016 + 2017

---

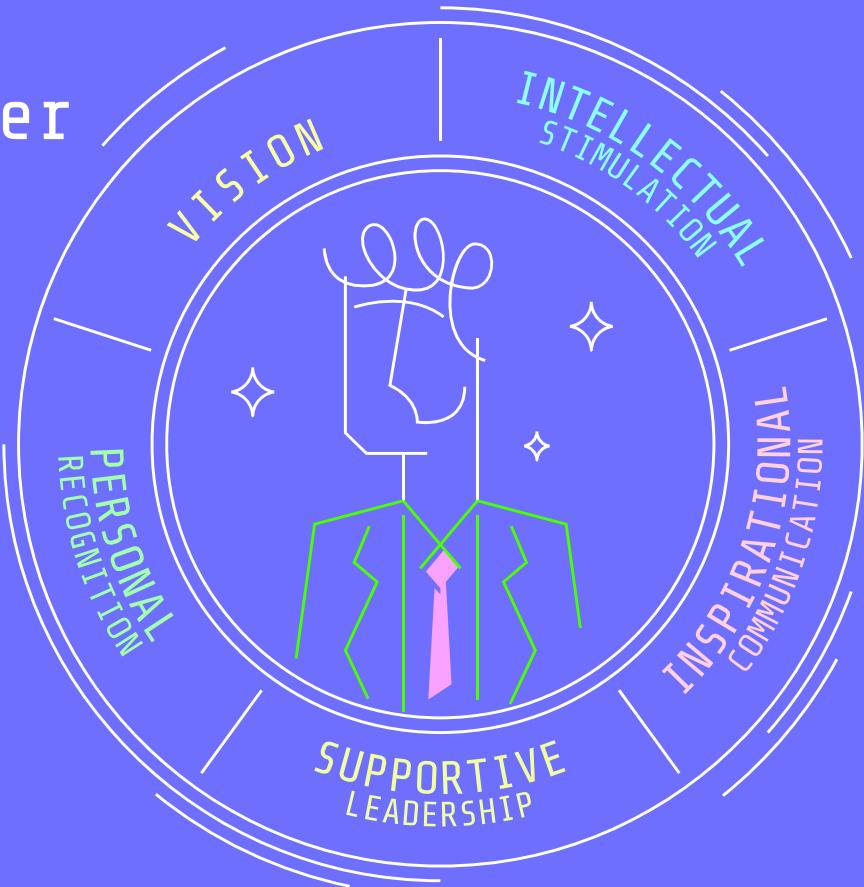
**A virtuous cycle between LPM and software delivery pipeline**



# DevOps Report: Impact of Leadership\_

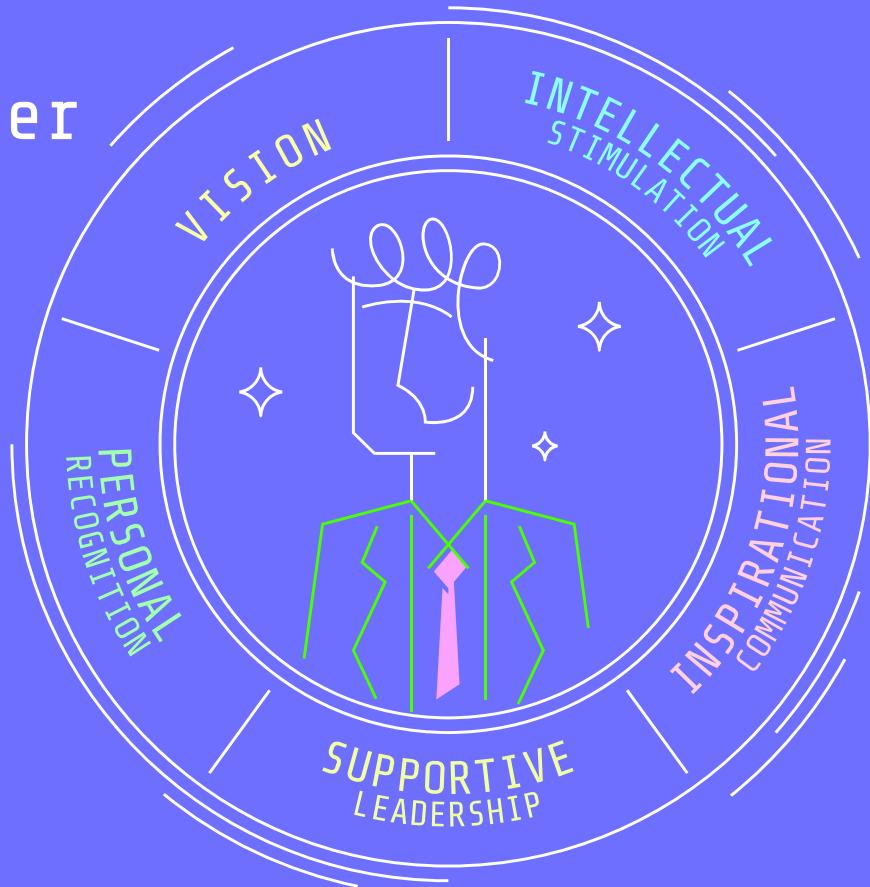
# DevOps Report: Transformational Leader

Statistically significant differences between high-performing and low-performing teams when it comes to **5 leadership characteristics**



# DevOps Report: Transformational Leader

Teams with **least transformative leaders** are 50% less likely to exhibit high performance



# DevOps Report: Transformational Leader

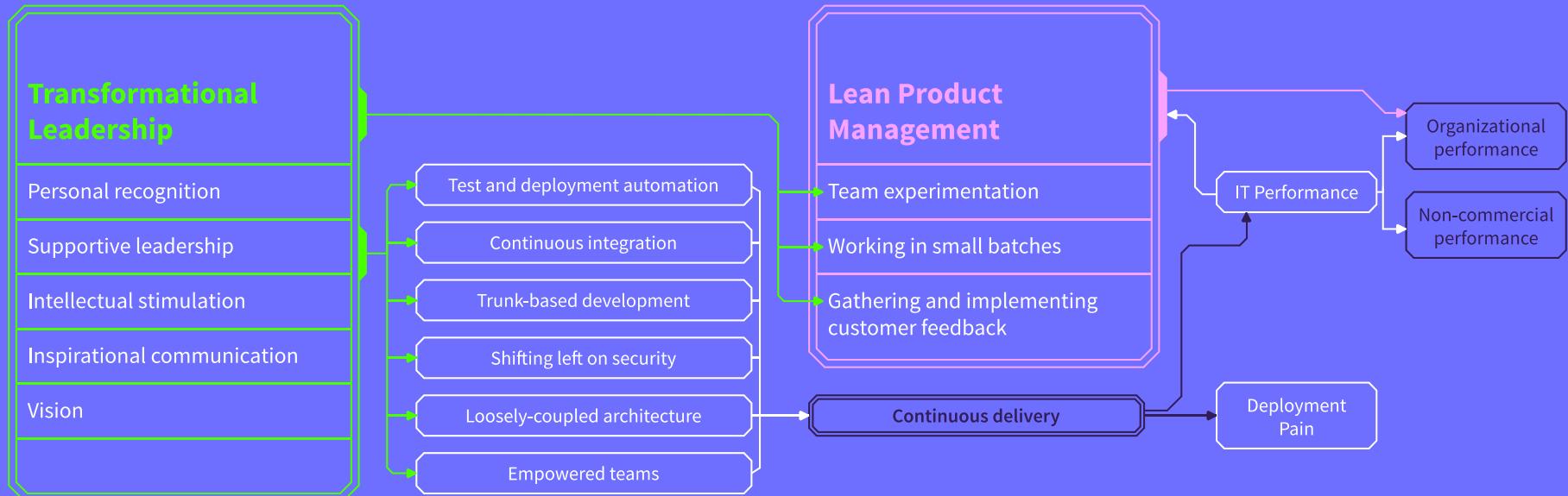


Figure 1. Structured equation model showing relationships between constructs



Objective Team Transformation  
→ How should your team evolve?



1

# → Dig Deeper

Objective Team Transformation



# Pick a Practice or → Insight from Here

2

Objective Team Transformation



# Favor Objective → Measures

3

Objective Team Transformation



# Measure Impact on Your Team

4

Objective Team Transformation



# Act on → Measured Impact

5

Objective Team Transformation

Who was this  
Akrem dude ?



★Designed by TownMapsUSA.com



@AkremSaed

WHO WHERE  
WHEN WHY  
WHAT HOW\_?