



# HOW TO MISUSE AND ABUSE DORA METRICS

# BRYAN FINSTER

---



Distinguished Engineer  
Value Stream Architect



Deploy more & sleep better

# IMPROVING DELIVERY A JOURNEY OF DISCOVERY

2014

## Leadership Ask

Increase release frequency from quarterly to bi-weekly

*“Why can’t we release every day?”*

# EXPERIMENTS

Aligned teams to business domains

Loosely coupled architecture

Created a CD platform team



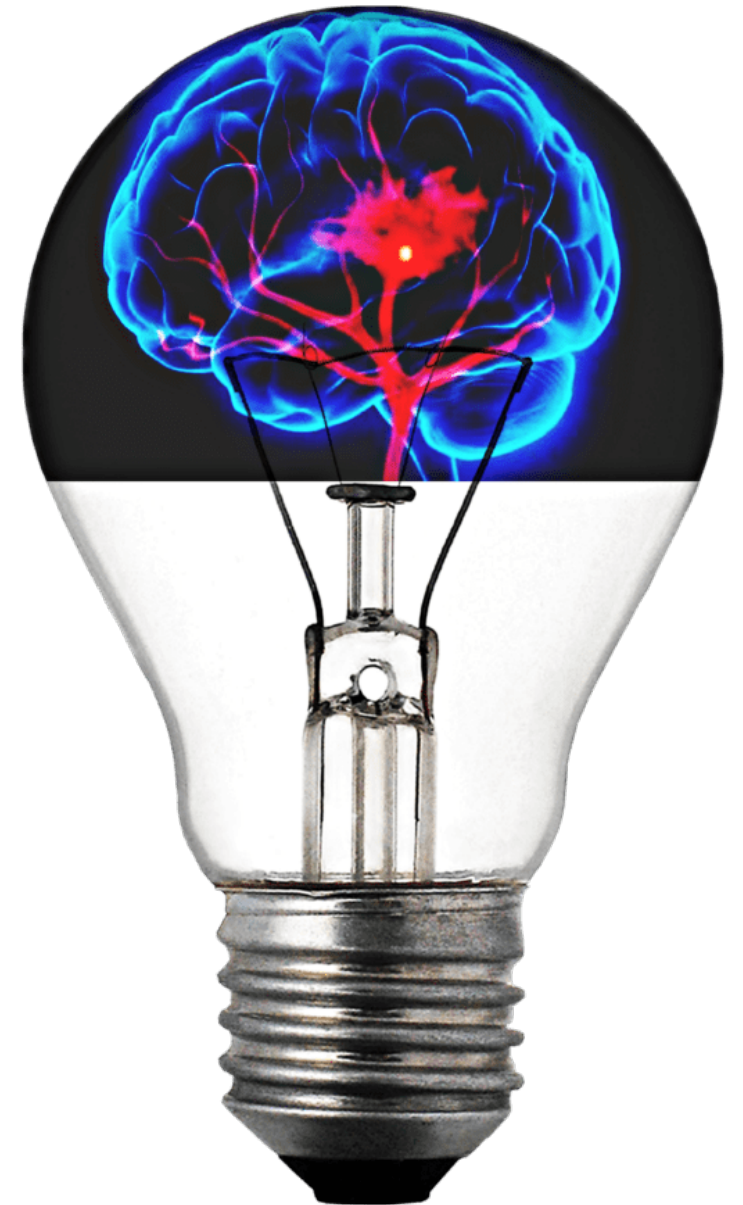
# REVELATIONS

Product teams and pipelines > scaling frameworks

*"Why can't we deliver today?":* Best tool for uncovering organizational issues.

CD improves outcomes and morale

We needed better metrics



# PERVERSE INCENTIVES

---

METRICS CHANGE BEHAVIOR



**CHOOSE WISELY**

# MISTAKES WERE MADE

## Quality

- Test coverage

## Expectation

- Improved testing

## Reality

- Increased the number of poor tests

## Outcomes not compliance

- Deploy frequency + Defect rate



# MISTAKES WERE MADE

## Predictability

- Completion rate

## Expectation

- Teams keep commitments

## Reality

- Promoted planning over delivery

## Outcomes not compliance

- Lead time + Development cycle time + defect rate





# SCALING IMPROVEMENT

2017

## Enterprise Goal

Expand CD to all teams for more efficient, effective,  
and sustainable delivery

# EXPERIMENTS

Opinionated CD platform

Gamified CD signals

- Trunk-based development
- Continuous integration
- Daily deploy
- Stable pipelines

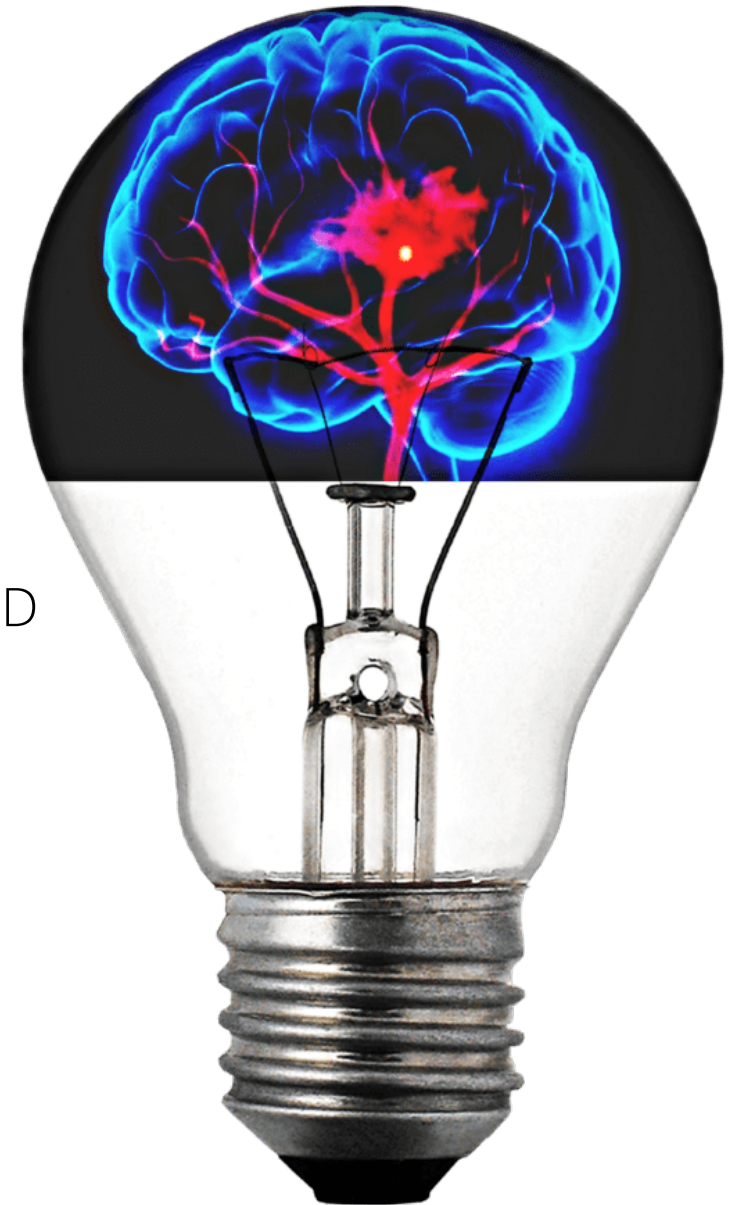


# REVELATIONS

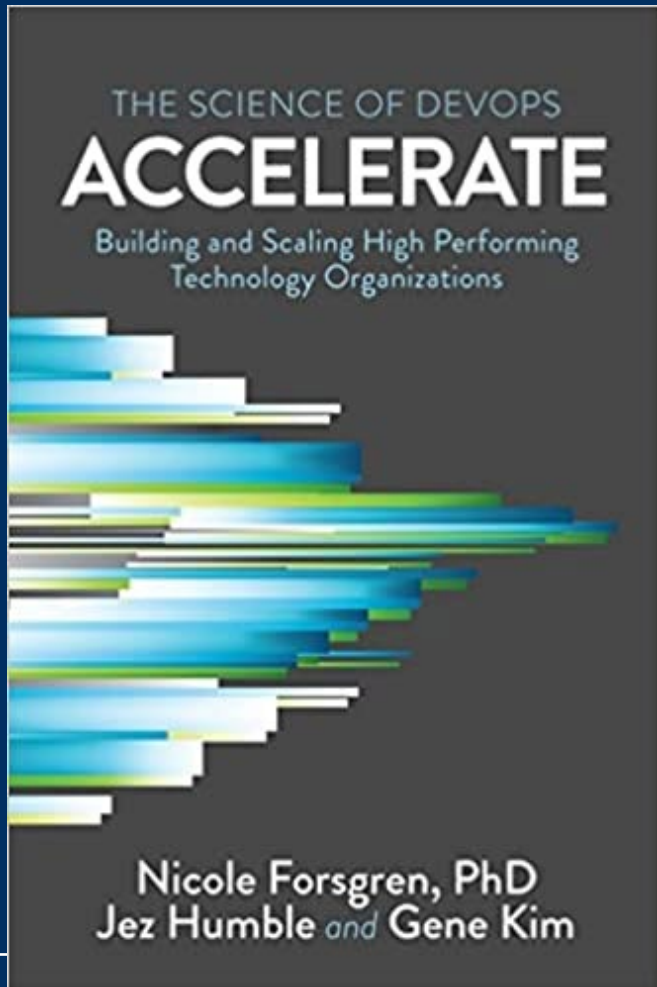
Gamified metrics helped early and middle adopters

Opinionated pipeline encouraged late adopters to explore CD

We needed better ways to communicate “why”.



# VALIDATED BY INDUSTRY DATA



2018

*“Continuous delivery improves both delivery performance and quality, and also helps improve culture and reduce burnout and deployment pain.”*

*-- Accelerate*

# The Four Key Metrics

# 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.

## 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.

# 2

# 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.

## 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.

# 4

Accelerate by Nicole Forsgren, PhD, Jez Humble, and Gene Kim

# WE CAUSED A PROBLEM

The 4 metrics make sense in the context of the rest of the book



# WE CAUSED A PROBLEM

The 4 metrics make sense in the context of the rest of the book

People don't read books

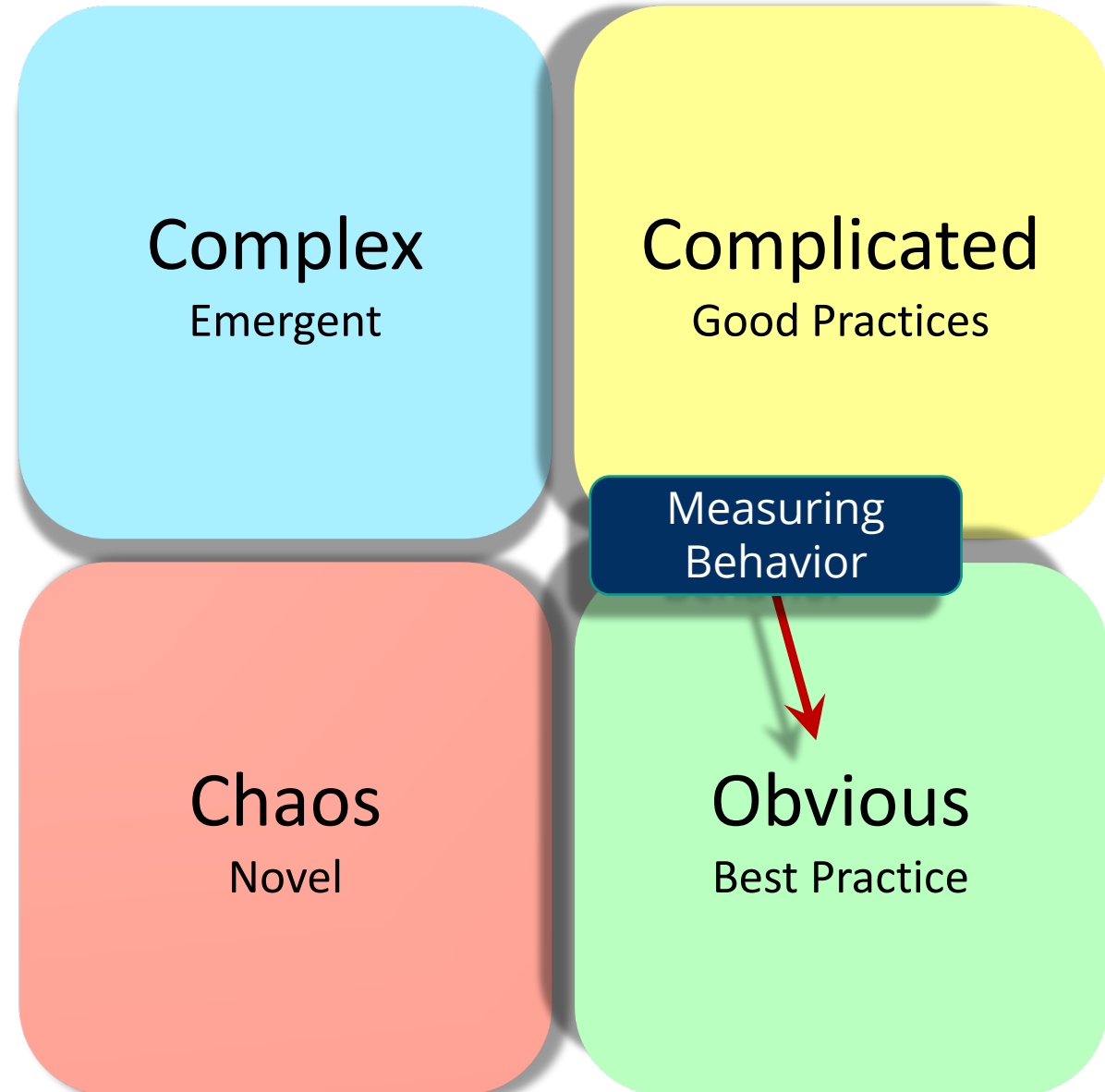


# WE CAUSED A PROBLEM

The 4 metrics make sense in the context of the rest of the book

People don't read books

We over-simplified the metrics





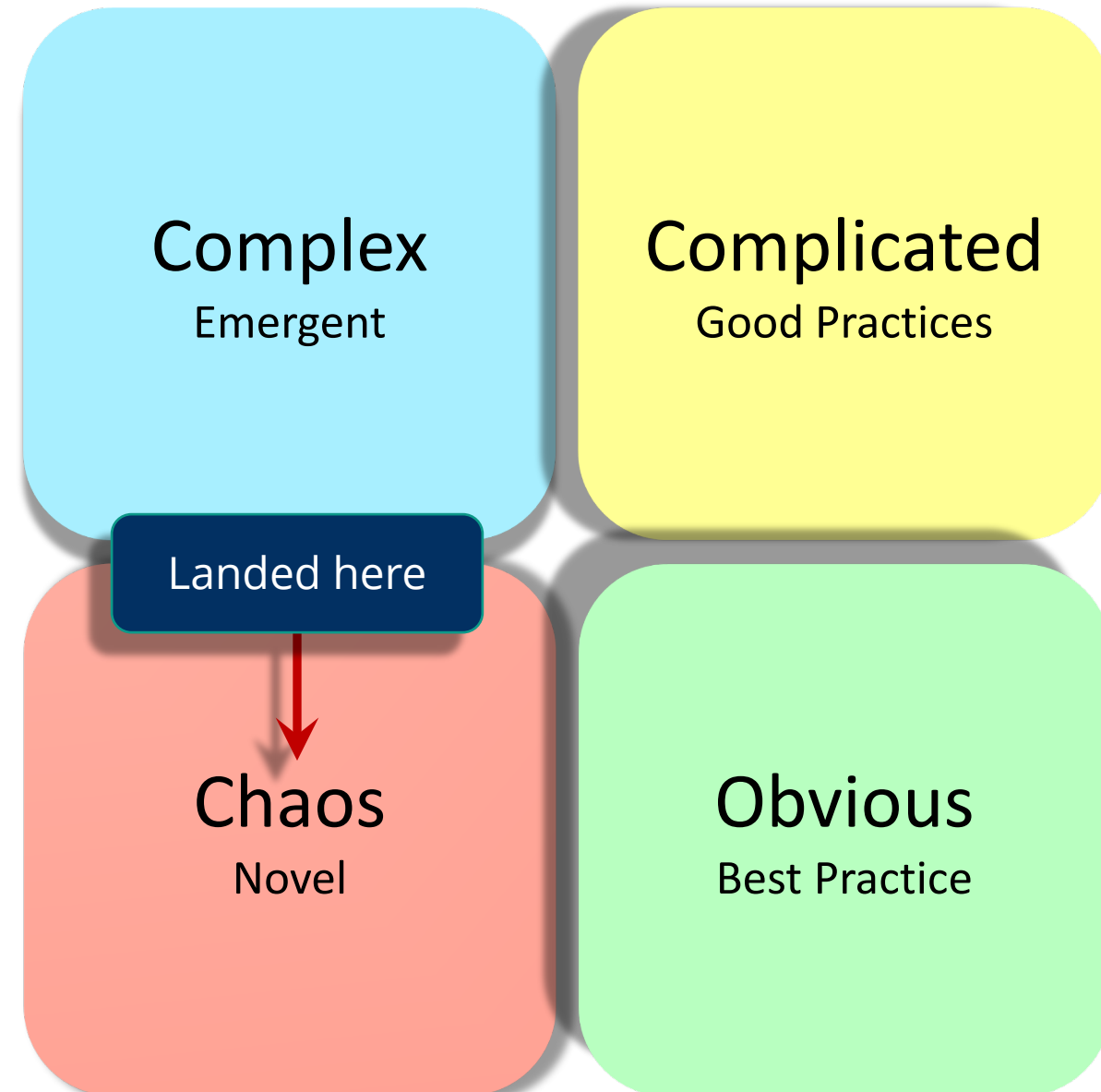
# WE CAUSED A PROBLEM

The 4 metrics make sense in the context of the rest of the book

People don't read books

We over-simplified the metrics

Their purpose got lost in translation



**YOU KEEP USING DORA METRICS**

**I DO NOT THINK THEY  
MEAN WHAT YOU THINK THEY MEAN**

# METRICS AS GOALS

---

## Fallacy

“To be a high performing organization, we need better DORA metrics. We need DORA OKRs!”

## Reality

- Correlation != Causation
  - High performing organizations focused on improving how they deliver value
  - Goals should focus on value
-

# PRODUCTIVITY

---

## Fallacy

“Now we can stack-rank the productivity of our teams.”

## Reality

- Each team has their own context
  - We are measuring health & improvement, not productivity
  - If we lack a generative culture, comparing teams is destructive to our goals.
-

# SPEED

---

## Fallacy

“We need to increase deploy frequency so we can deliver faster.”

## Reality

- Measure of batch size, not speed
  - Smaller batches delivered more frequently to expose waste and improve our quality processes...
  - Improved quality improves speed
-

# VANITY RADIATORS

How big is each org?

Time range?

What action does this inform?

Deploys	Lead Time	MTTR	Change Fail %
437	3 hours	5 hours	8%

Deploys	Lead Time	MTTR	Change Fail %
645	2.3 days	2 hours	10%

# DELIVERY HEALTH INDICATORS

A green ECG-style waveform is overlaid on a dark blue grid background. The waveform has several peaks and troughs, with the most prominent peak occurring near the top center of the image. The grid lines are light blue and form a consistent pattern across the entire background.

Reducing batch size?  
Improving quality and reliability?  
Reducing toil?  
Accelerating feedback?  
Happier customers?  
Happier teams?

# FOUR METRICS?



An incomplete view



# FOUR METRICS?

Deploy Frequency

Lead Time for Change

Change Fail %

MTTR

**Flow**

- Total lead time
- Work in progress

**Culture**

- Westrum score
- Employee NPS
- After hours work

**Continuous Integration**

- Branch duration
- Integration frequency
- Mean time to detect

**Customer Outcomes**

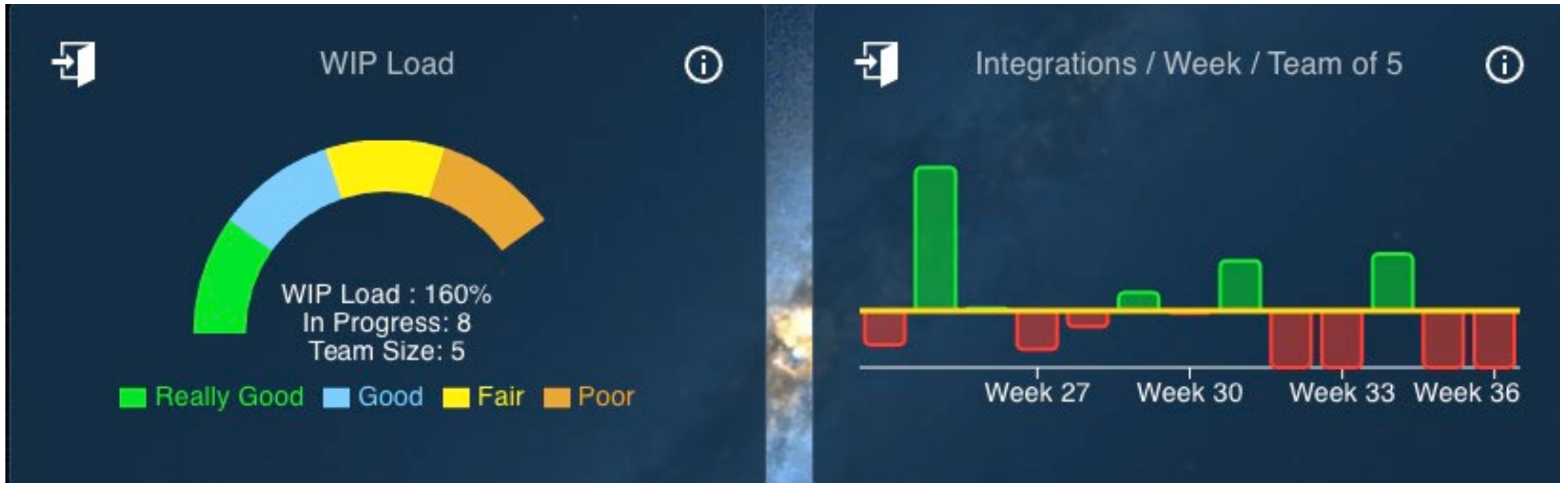
- NPS
- Downtime for deploy



# INFORMATION RADIATORS

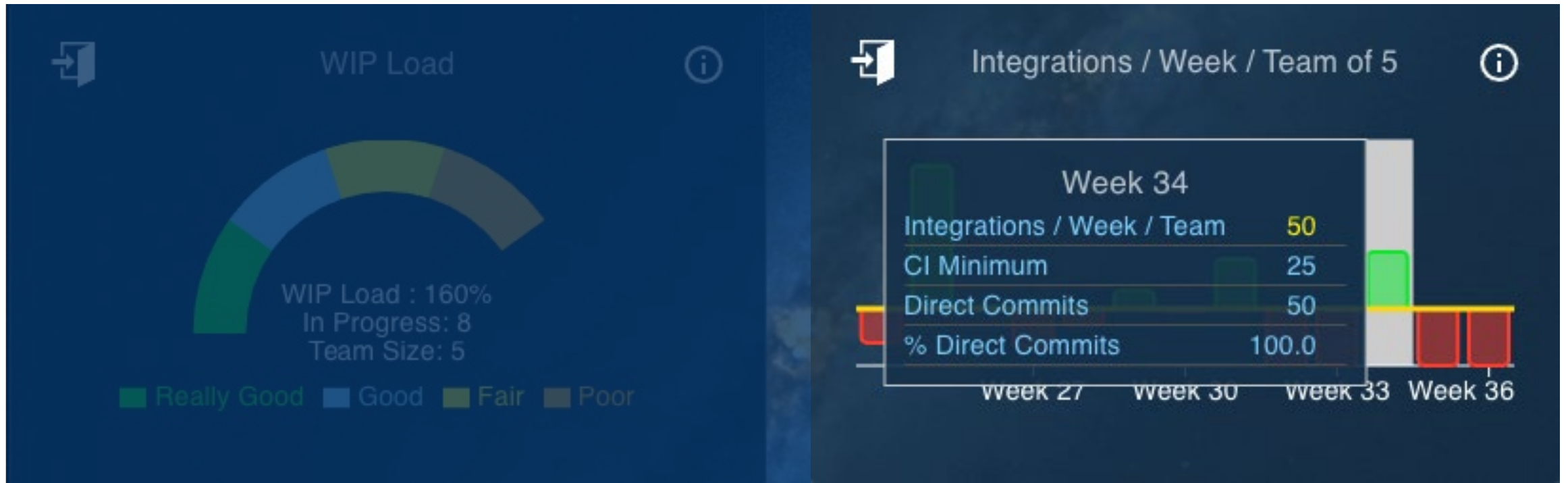
“It’s only when they’re combined with the use of visual displays... that we see a strong effect.”

-- Accelerate



# INFORMATION RADIATORS

Communicate the definition of “good.”



# INFORMATION RADIATORS

“A goal without a method is cruel.”

- W. E. Deming

## Code Integration Frequency

### Definition

[Code Integration Frequency](#) tracks how often tested, production-worthy code changes are integrated to the trunk. The minimum threshold for continuous integration is at least once per day for each developer on the team.

### Goal of Measuring

- Smaller changes improve the effectiveness of code review, give faster quality feedback, and reduce the chances of defects introduced during change conflict resolution.
- Focusing on small changesets uncovers issues with work decomposition that can help the team improve the quality of the requirements

### Tips for Improvement

- Understand the basics of [Continuous Integration](#)
- Use [Behavior Driven Development](#) and [Test Driven Development](#) to help with breaking down changes to smaller units to incrementally deliver features.
- Use [feature flags, branch by abstraction, or other techniques](#) to separate deployment from release.

### Data Source

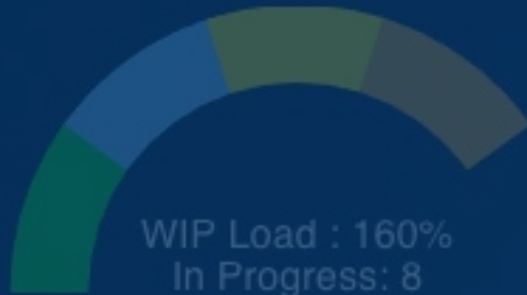
Data is aggregated from the team's source repositories.

### Possible Data Quality Issues

- Not all of the team's source repositories are properly identified and accounted for.



## WIP Load



Really Good Good Fair Poor

## Week / Team of 5



Week 27 Week 30 Week 33 Week 36

# Culture

Trust - Mission - Learning - Improving



## Business Goals, Customer Value, & Sustainability

- Profitable, Happy customers, Happy teams



## Business Objectives

- Key results



## Flow

- WIP, Lead time, Development cycle time, Throughput, Flow efficiency



## Continuous Delivery

- Deploy frequency, Pipeline cycle time, Defect rate, MTTR



## Continuous Integration

- Code integration frequency, Branch duration

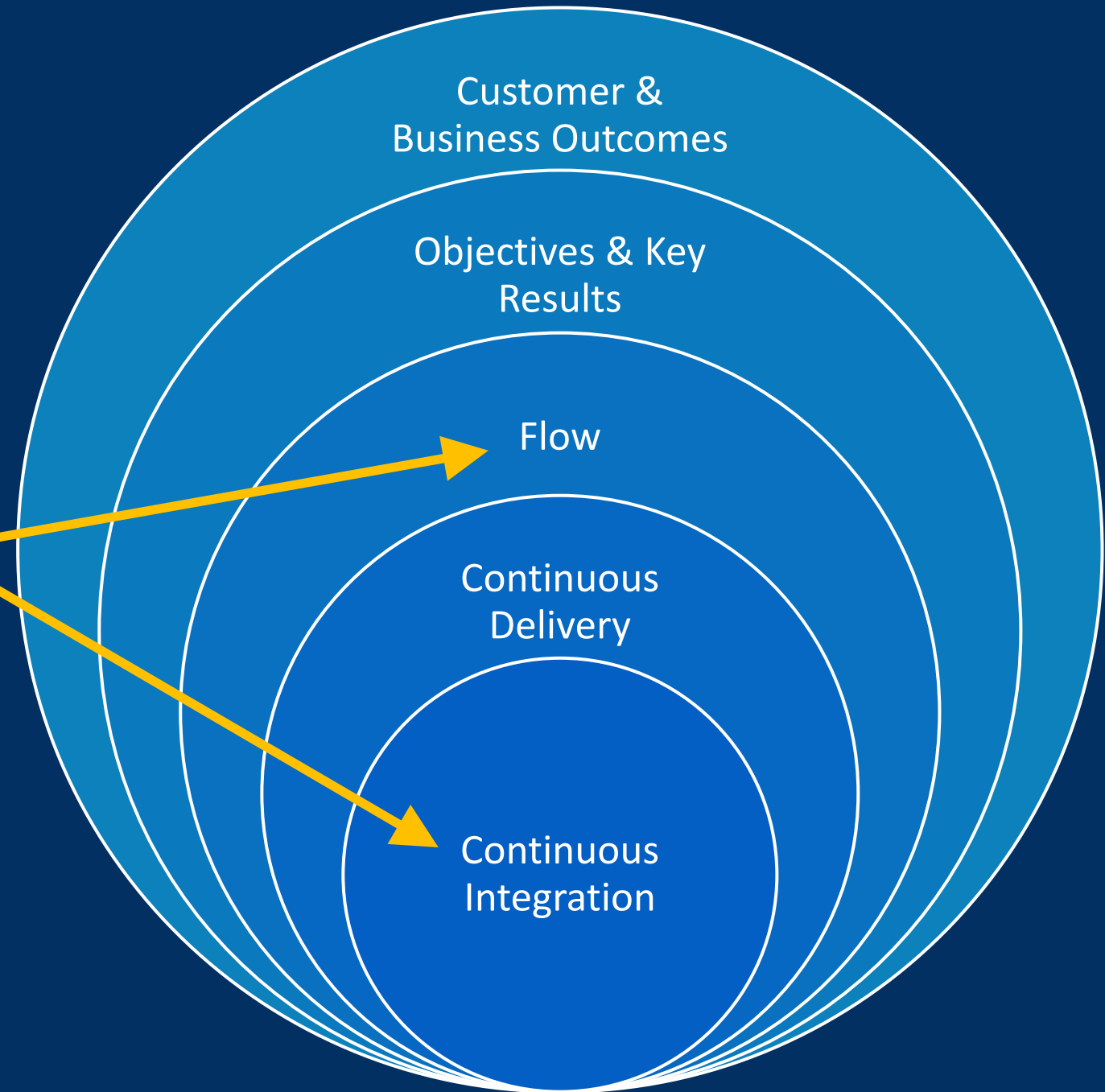
Low

Data Fidelity

High

# UNDERLYING METRICS CONSTRAIN IMPROVEMENT

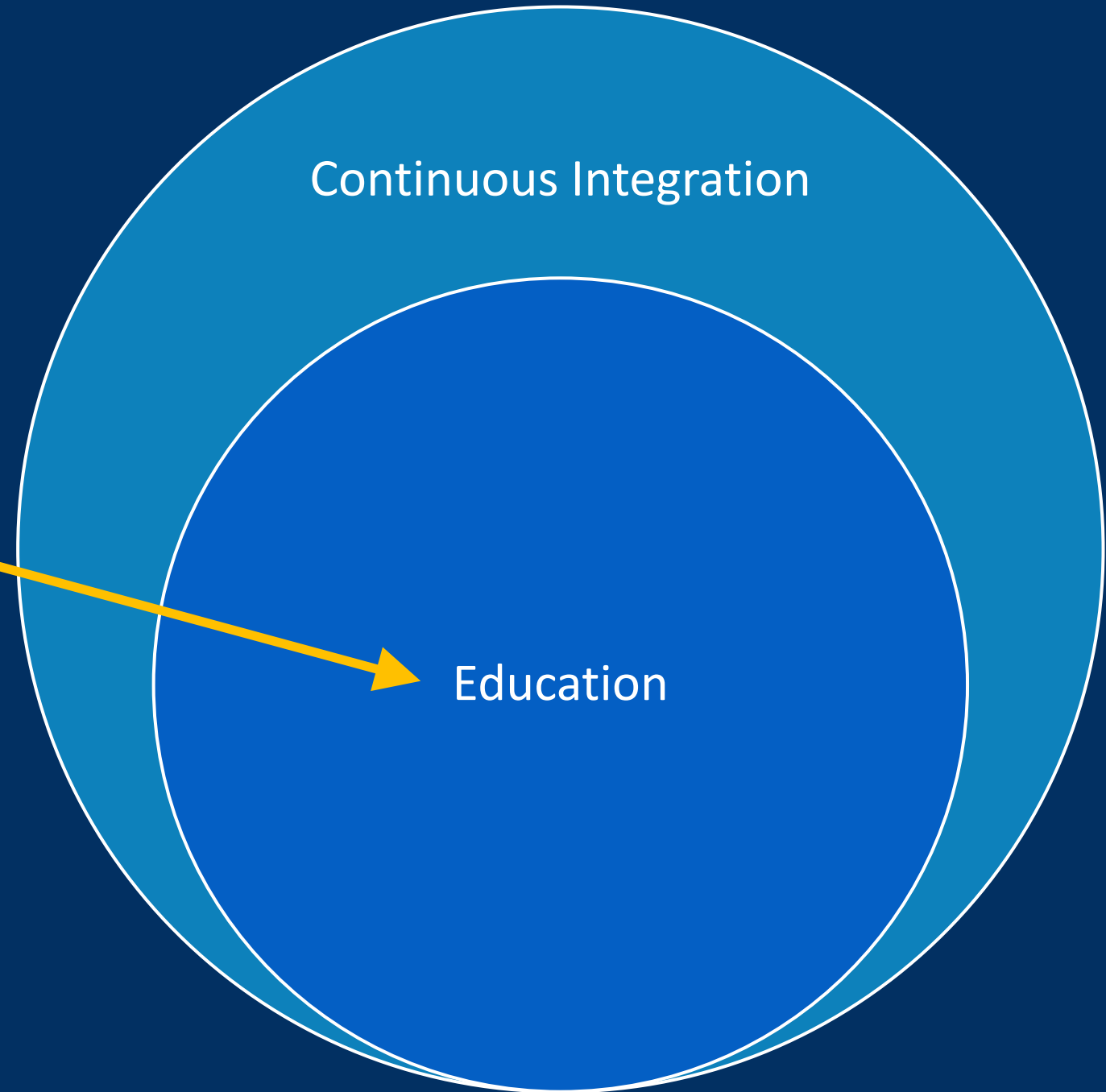
To improve flow, we must  
improve CI



# INVESTMENT CONSTRAINS METRICS

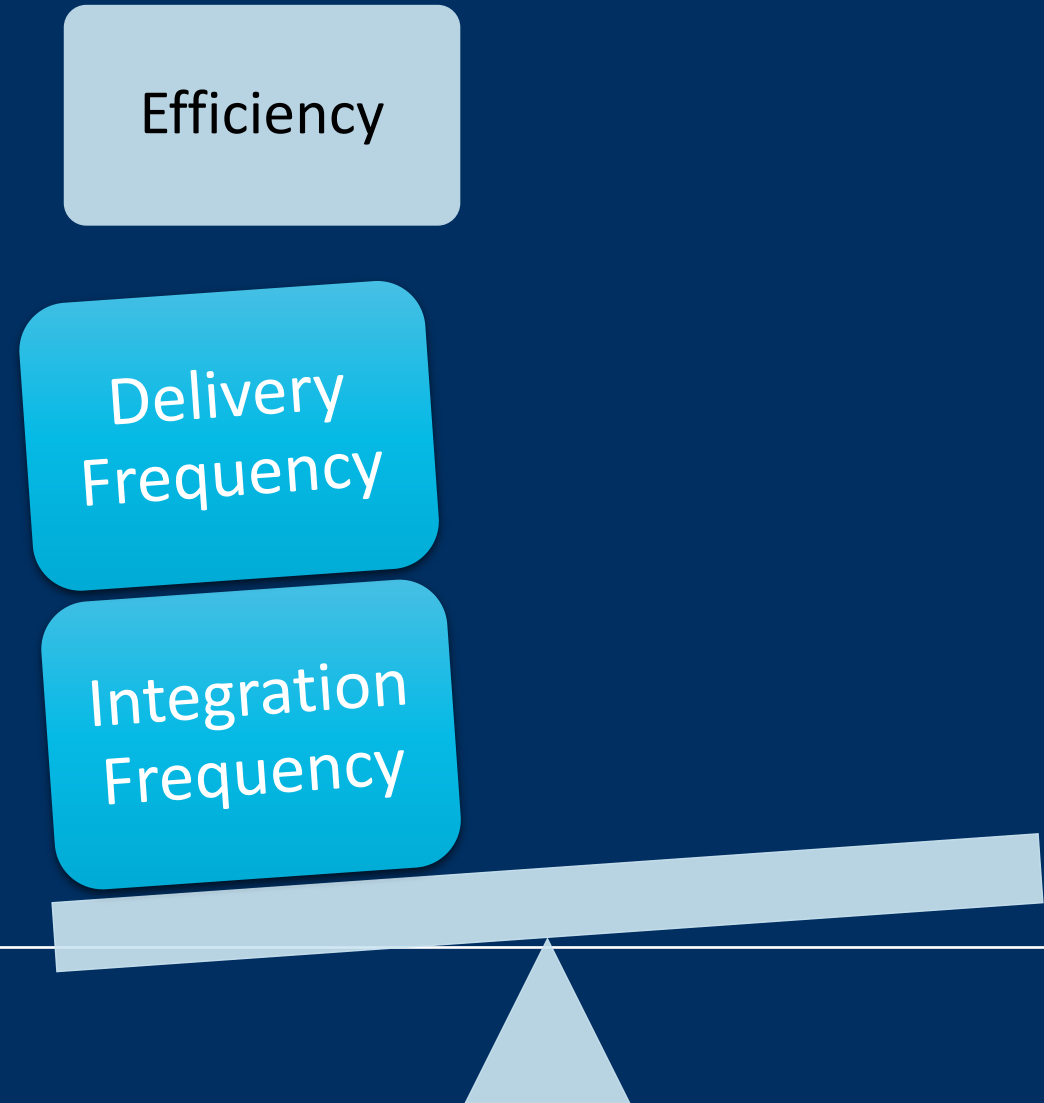
We cannot wager our goals on hope that teams will find the right information and self-train.

---



# METRICS NEED BALANCE

Smaller batches uncover pain

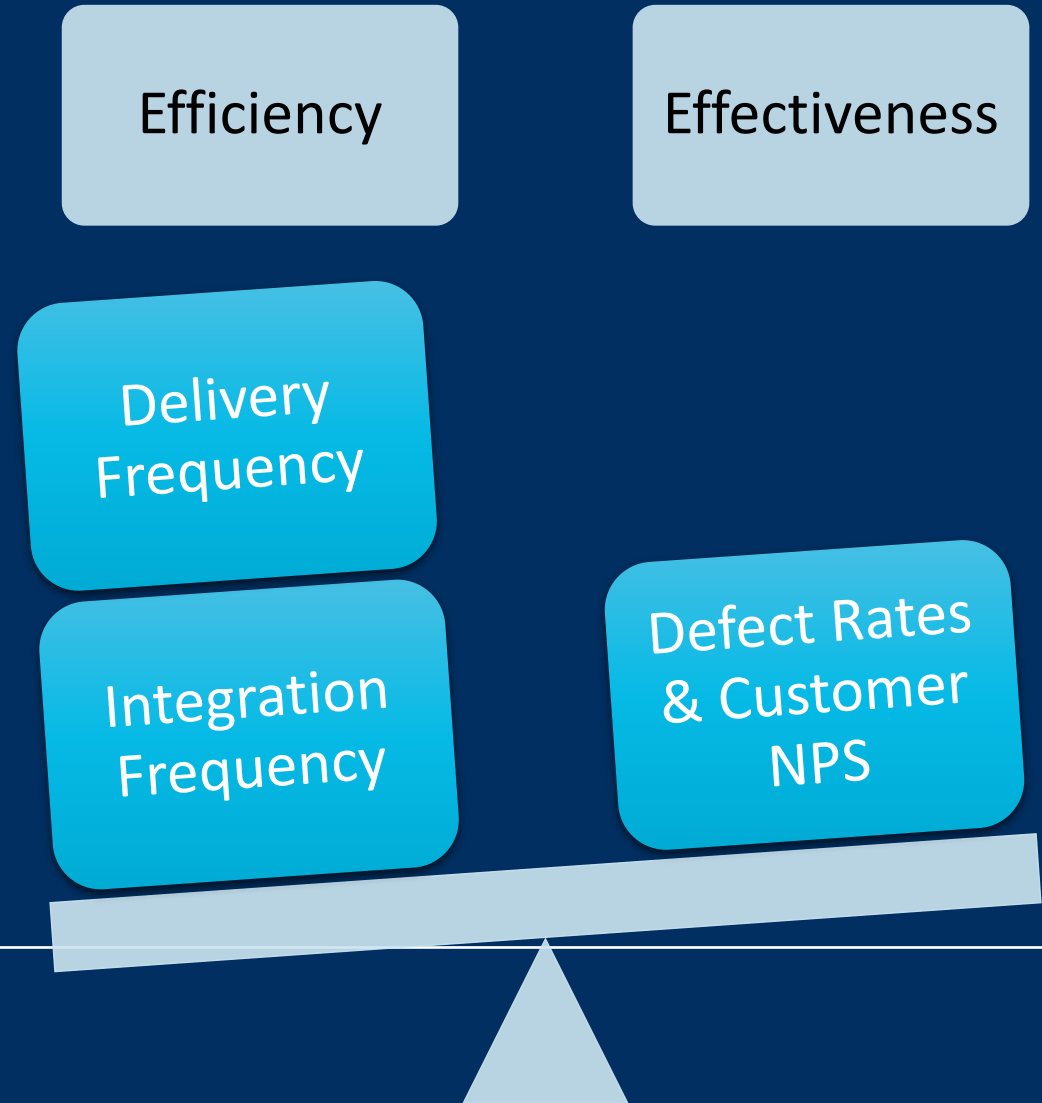




# METRICS NEED BALANCE

Smaller batches uncover pain

Quality is a guardrail against prioritizing speed



# METRICS NEED BALANCE

Smaller batches uncover pain

Quality is a guardrail against prioritizing speed

Team feedback is guardrail against burnout

Efficiency

Effectiveness &  
Sustainability

Delivery  
Frequency

Employee  
NPS

Integration  
Frequency

Defect Rates  
& Customer  
NPS



# CLOSING THOUGHTS

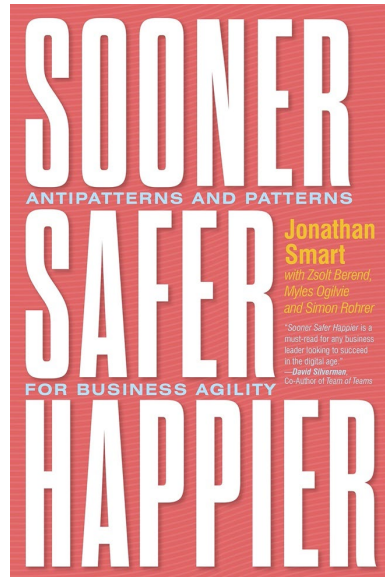
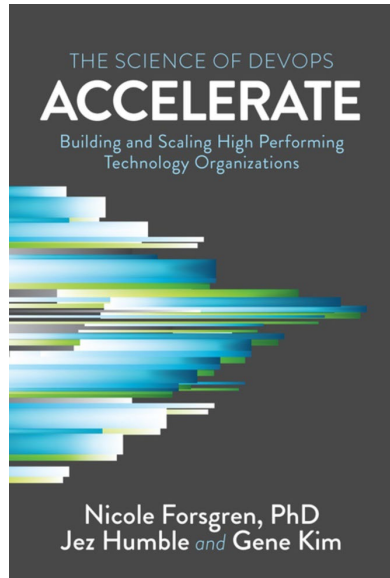
- The 4 outcome metrics are only the tip of the iceberg.
- Product development is a complex interaction of people, process, and products. There are no simple metrics.
- Measures require guardrails to avoid perverse incentives.

# CLOSING THOUGHTS

- Metrics are a critical part of the improvement toolbox, but...
  - ✓ We cannot measure our way to improvement.
  - ✓ We use them to monitor and inform the next improvement experiment.
- Don't measure people, invest in them. They are our most valuable asset.

# RESOURCES

---



[bit.ly/continuous-videos](https://bit.ly/continuous-videos)

# LET'S TALK ABOUT IMPROVING!



*Defense*  
*Unicorns*

DefenseUnicorns.com

Medium: [bdfinst.medium.com](https://bdfinst.medium.com)

LinkedIn: [bryan-finster](#)

eMail: [Bryan.Finster@DefenseUnicorns.com](mailto:Bryan.Finster@DefenseUnicorns.com)