The best time to start Continuous Delivery is *Now*

Josh Reed – Release Engineering – Aiven


CI/CD

What is Continuous Delivery?

- Always stay release-ready
- Validate by releasing often
- Make the release process mundane
- Get quick feedback about changes to code
- Achieve quality at speed

Four Key Metrics

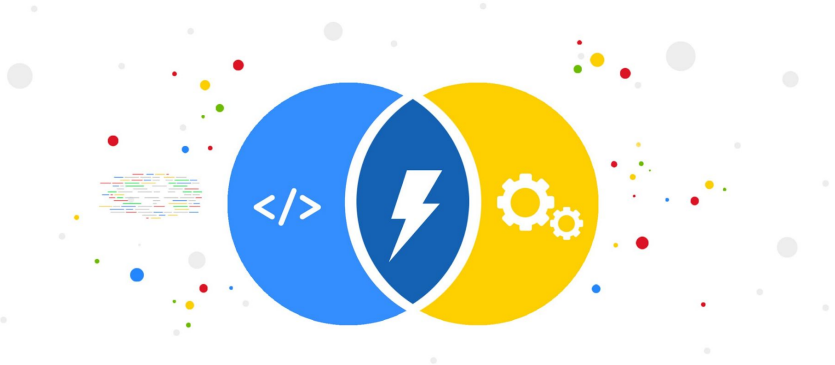
Lead Time

Deployment Frequency

Change Failure Rate

Recovery Time

Compare to Industry



See the State of DevOps Report from Google DORA

“But I don’t need it!”

That’s the perfect time to start!

One Exception

If you know you’re going to be making software, you can benefit from continuous delivery

Early Adoption

Delivery vs Deployment

- SaaS bias in the literature
- Delivery can mean many things
- Applies to remotely installed software
- Can even apply to internal deliveries
- Don’t forget about R&D!


Ship it!

“But I already have a process!”

How’s that working out for you?

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%

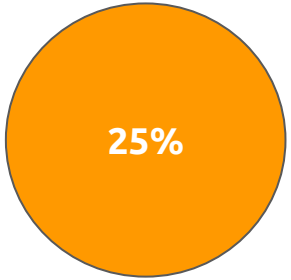
Source: https://cloud.google.com/blog/products/devops-sre/using-the-four-keys-to-measure-your-devops-performance

Quarterly Release

Lead Time

Deployment Frequency

Change Failure Rate



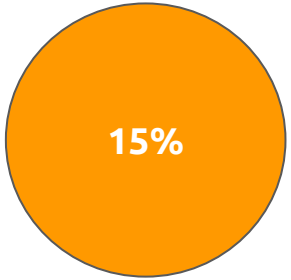
Recovery Time

End of the Sprint

Lead Time

Deployment Frequency

Change Failure Rate



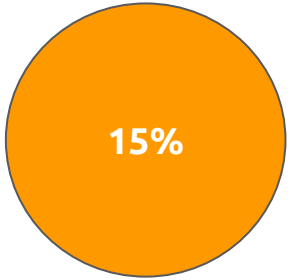
Recovery Time

Process is King

Lead Time

Deployment Frequency

Change Failure Rate



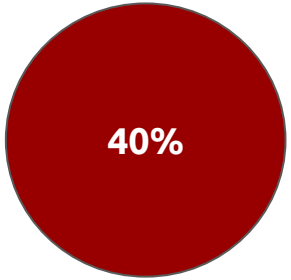
Recovery Time

YOLO into Prod

Lead Time

Deployment Frequency

Change Failure Rate



Recovery Time


Measure!

“But change is hard!”

It’s worth it.

Someone is Paying the Price

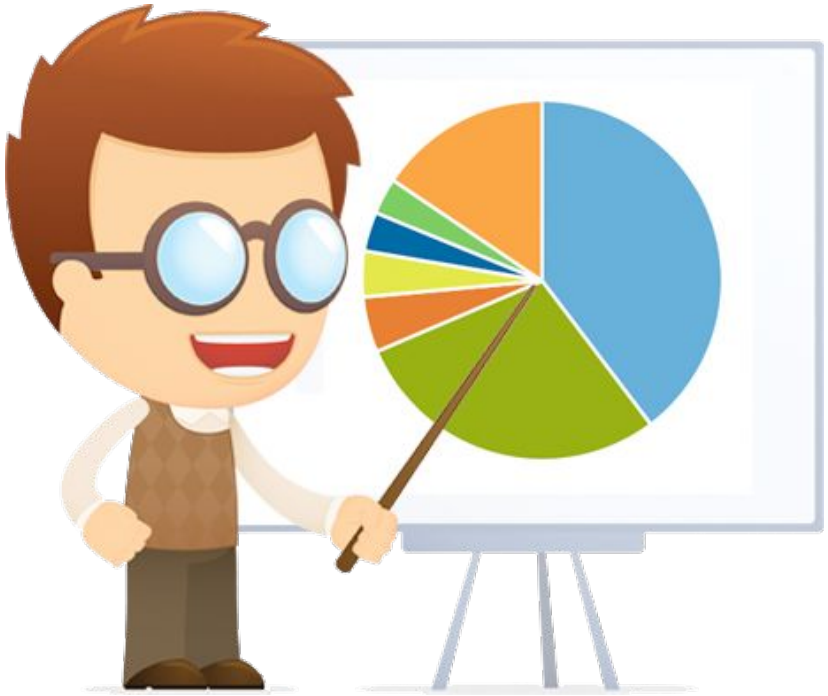
Product

Developers

Customers

Operations

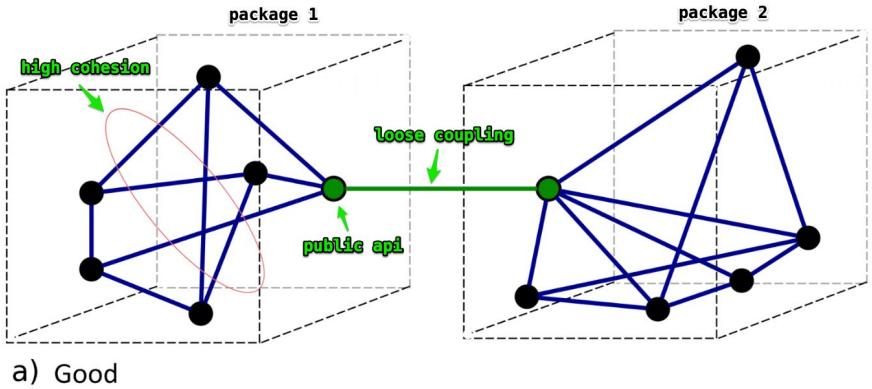
Analyze and Iterate



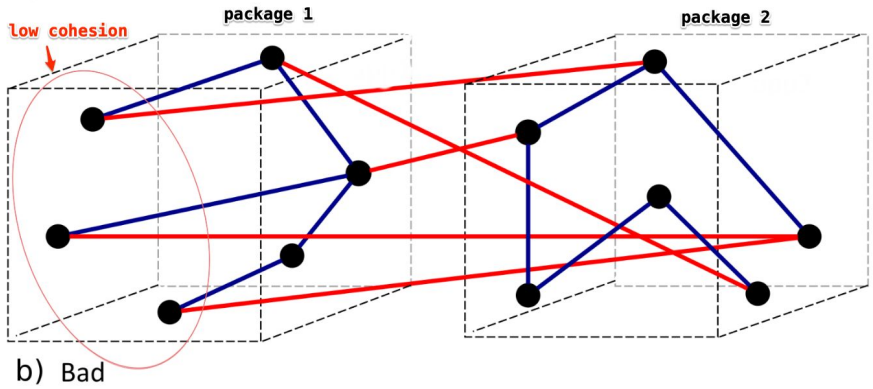
Technical Approaches

- Automate
- Decouple
- Refactor

Architecture



a) Good

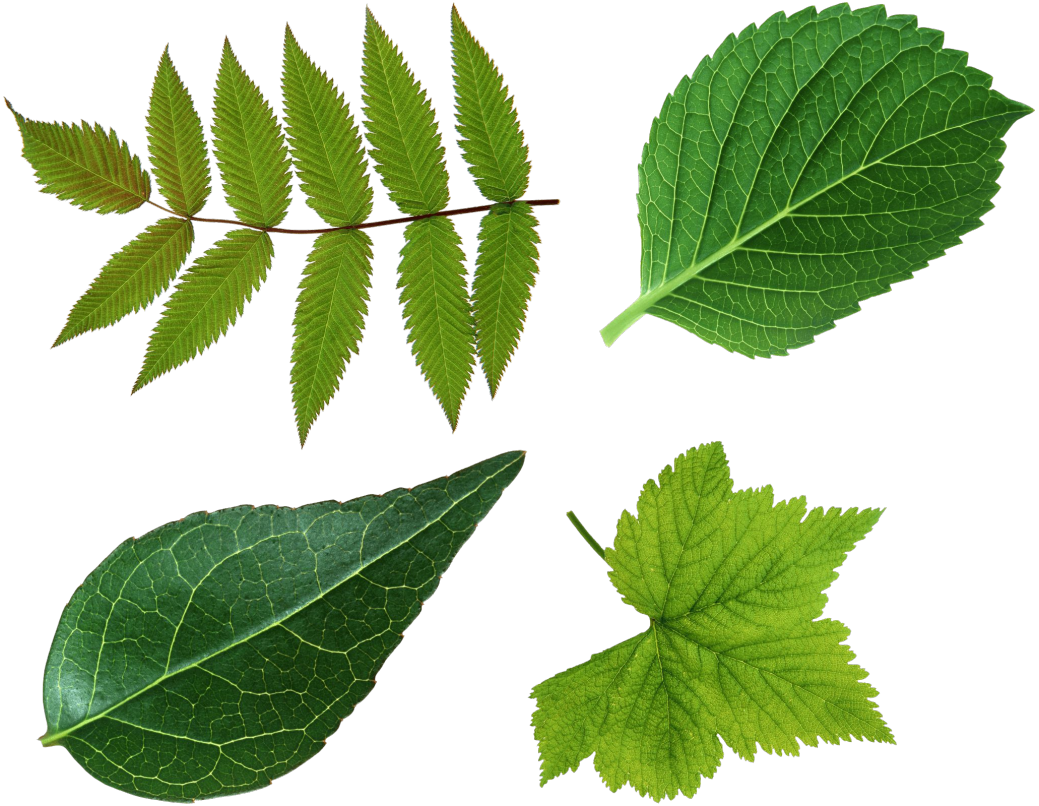


b) Bad

https://github.com/Shopify/packwerk/blob/main/USAGE.md

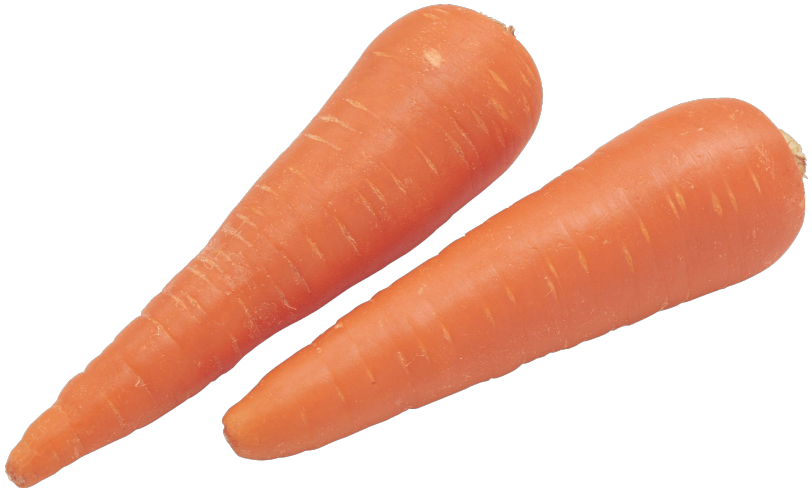
Tooling

Start at the edges

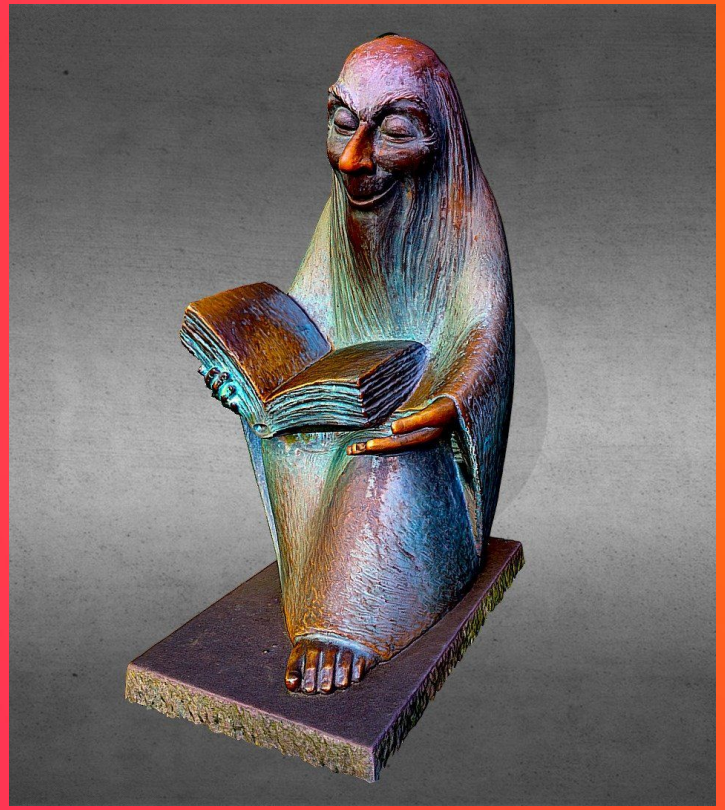


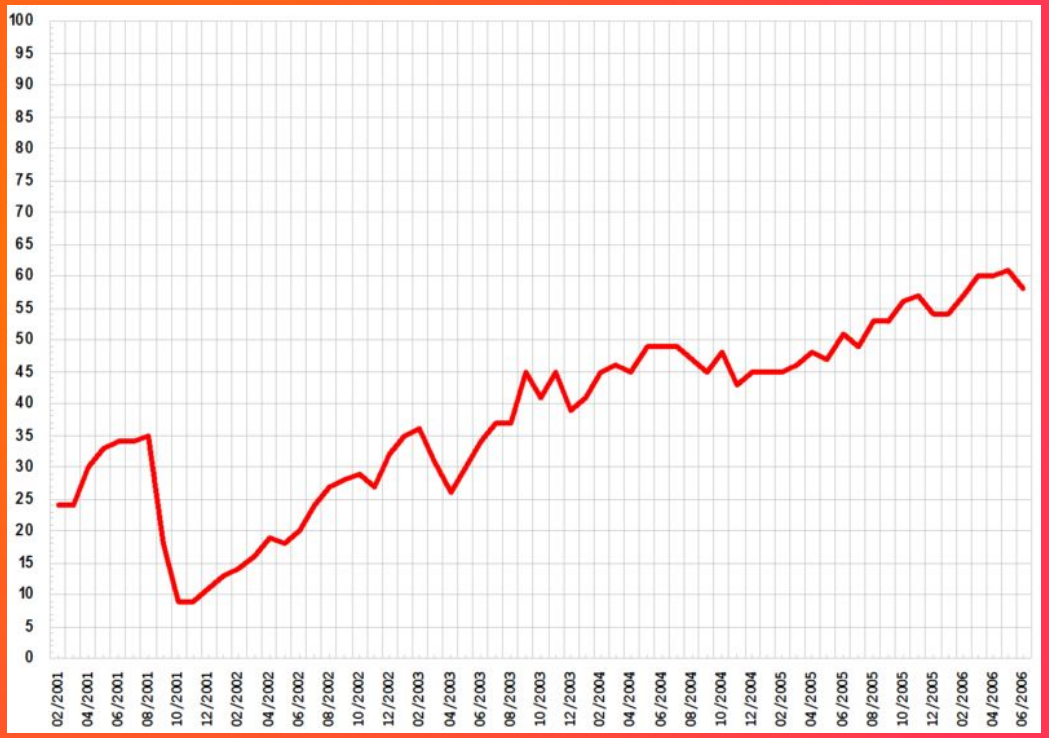
Build Paved Roads

Carrot and Stick



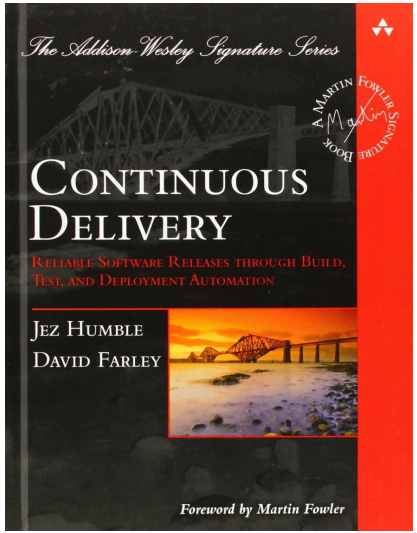
Dealing with People

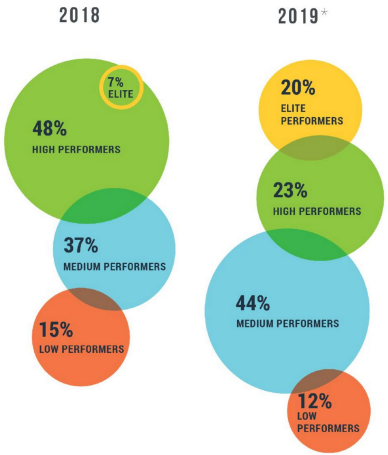


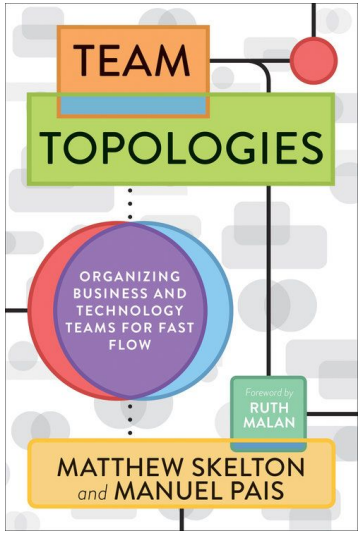


Keep Going!

Resources


The Addison-Wesley Signature Series
A Martin Fowler Signature Book
CONTINUOUS DELIVERY
RELIABLE SOFTWARE RELEASES THROUGH BUILD, TEST, AND DEPLOYMENT AUTOMATION
JEZ HUMBLE
DAVID FARLEY
Foreword by Martin Fowler


2018
2019*
7% ELITE
48% HIGH PERFORMERS
37% MEDIUM PERFORMERS
15% LOW PERFORMERS
20% ELITE PERFORMERS
23% HIGH PERFORMERS
44% MEDIUM PERFORMERS
12% LOW PERFORMERS


TEAM
TOPOLOGIES
ORGANIZING BUSINESS AND TECHNOLOGY TEAMS FOR FAST FLOW
Foreword by
RUTH MALAN
MATTHEW SKELTON
and MANUEL PAIS

Contact

Josh Reed

@jriddycuz

https://www.linkedin.com/in/josh-reed-3469383a/

Release Engineering @ aiven.io

aiven