

A hand in a plaid shirt holds a white paper boat on a wet, reflective surface. The boat and the hand are reflected in the water. The background is a blurred, wet pavement with puddles.

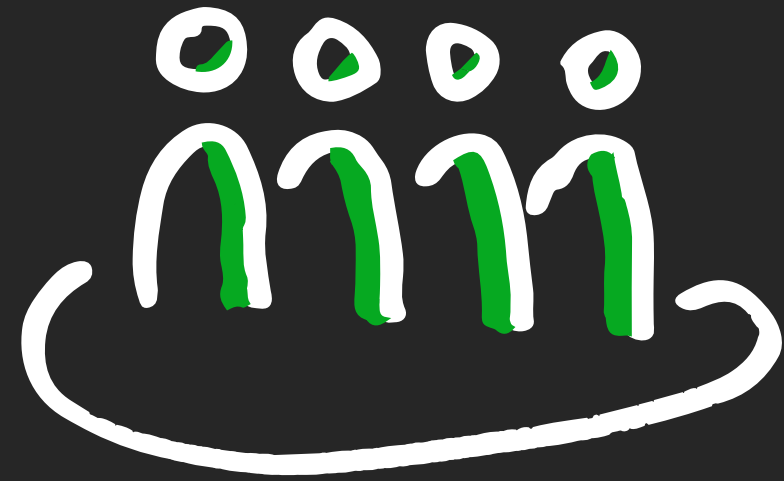
You build it,
but you ain't gonna run it

DevOps practices for IT providers

Konstantin Diener | konstantin.diener@cosee.biz | [@coseeaner](https://twitter.com/coseeaner)







Development

- as many Features as possible
- quick response
- Scrum: Shippable Product

>> throughput



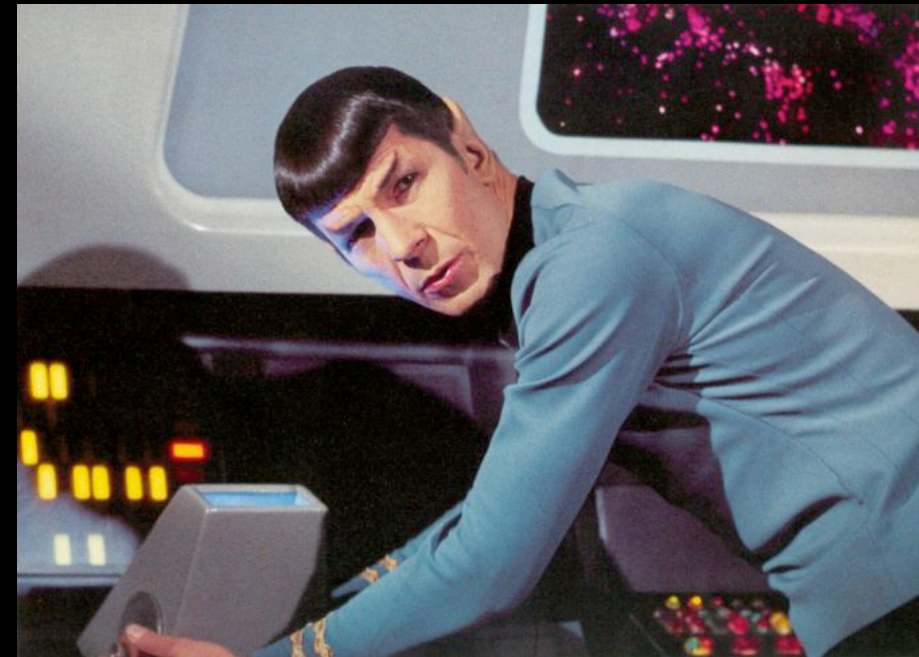
Operations

- stable application
- no Incidents or major bugs
- no Downtimes

>> stability



Dev and Ops



Little bit weird
Sits closer to the boss
Thinks too hard



Pulls levers & turns knobs
Easily excited
Yells a lot in emergencies

Ops' job is to **enable** the business
(this is dev's job too)

**“ You build it,
you run it.”**

Werner Vogels

Maybe that's true for

product

organisations, but...

... when an **IT supplier**
comes into play it's often ...

You build it,

**but you ain't
gonna run it.**

Konstantin Diener

CTO and Co-Founder of
cosee



Start



Scale



product development

But, what's the

difference?



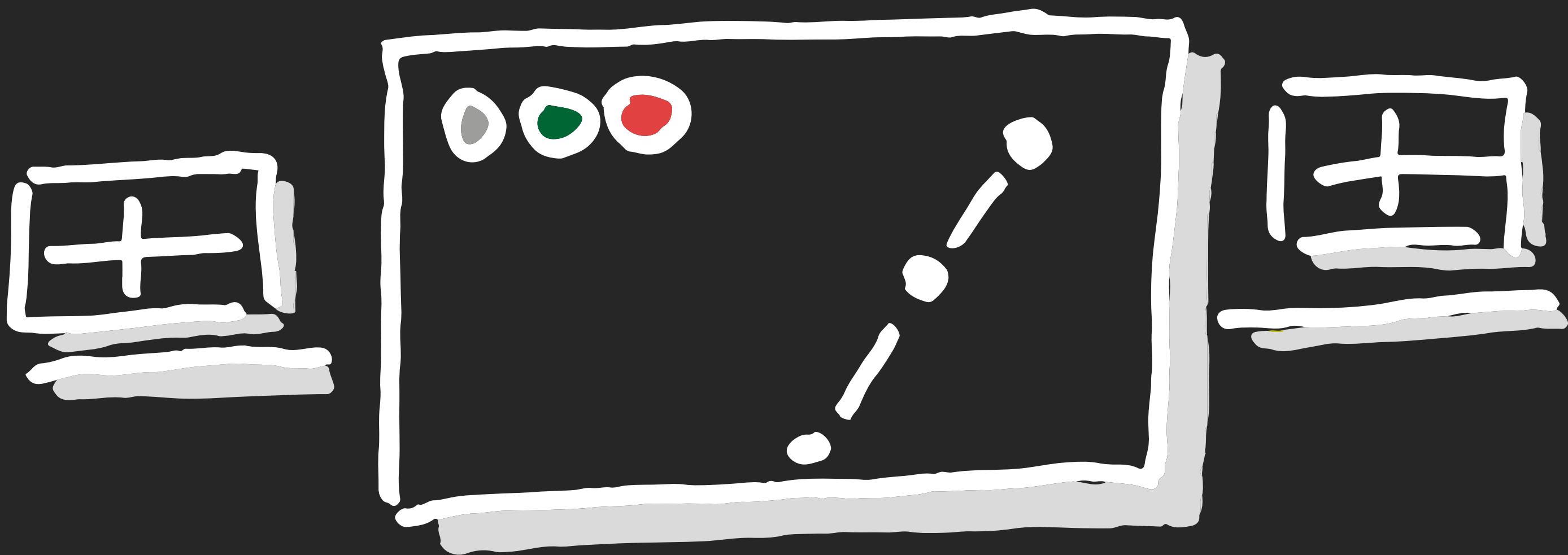
Supplier

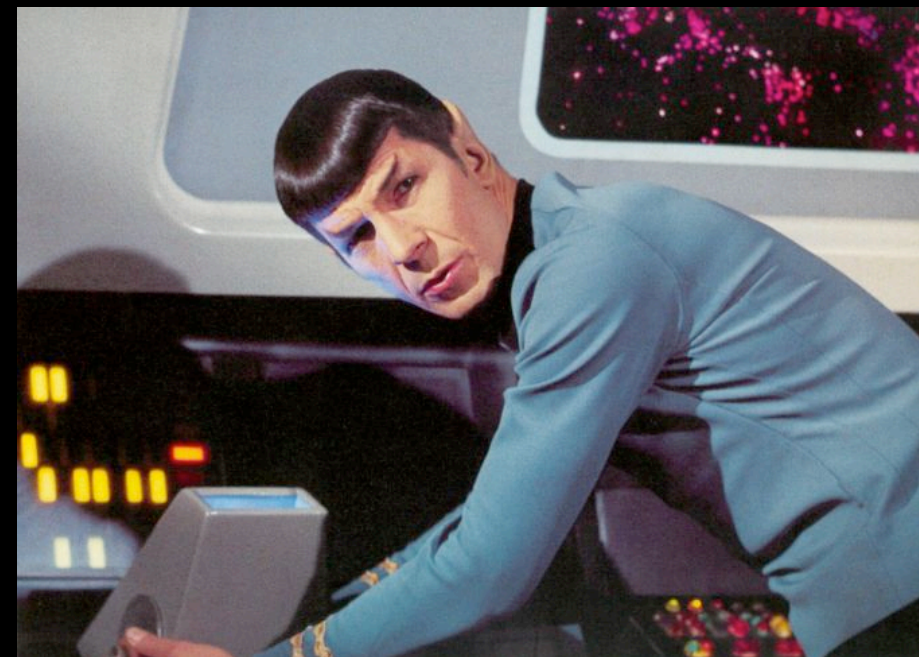


Client



Customers





Little bit weird
Sits closer to the boss
Thinks too hard

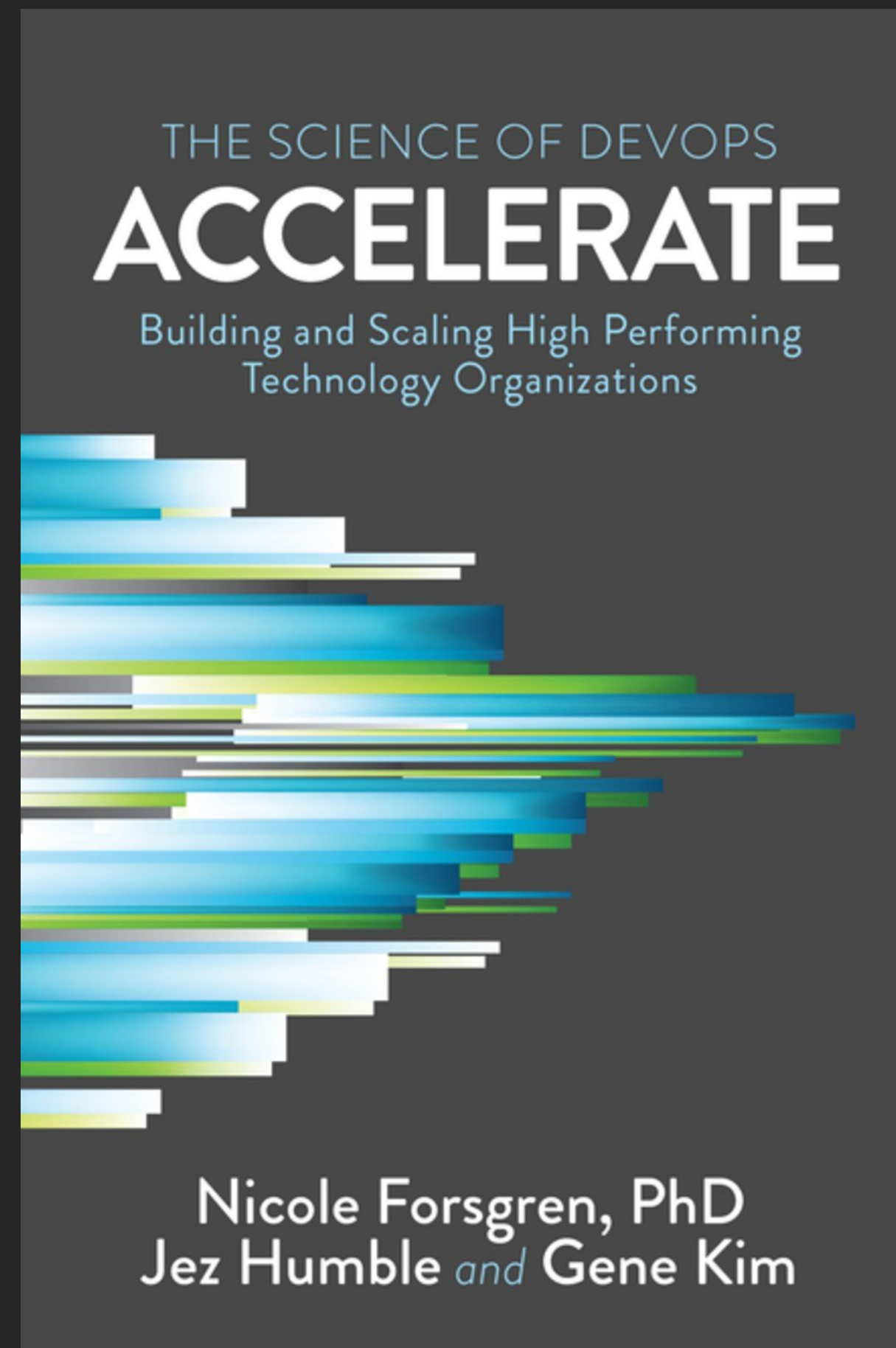


Pulls levers & turns knobs
Easily excited
Yells a lot in emergencies



What's the

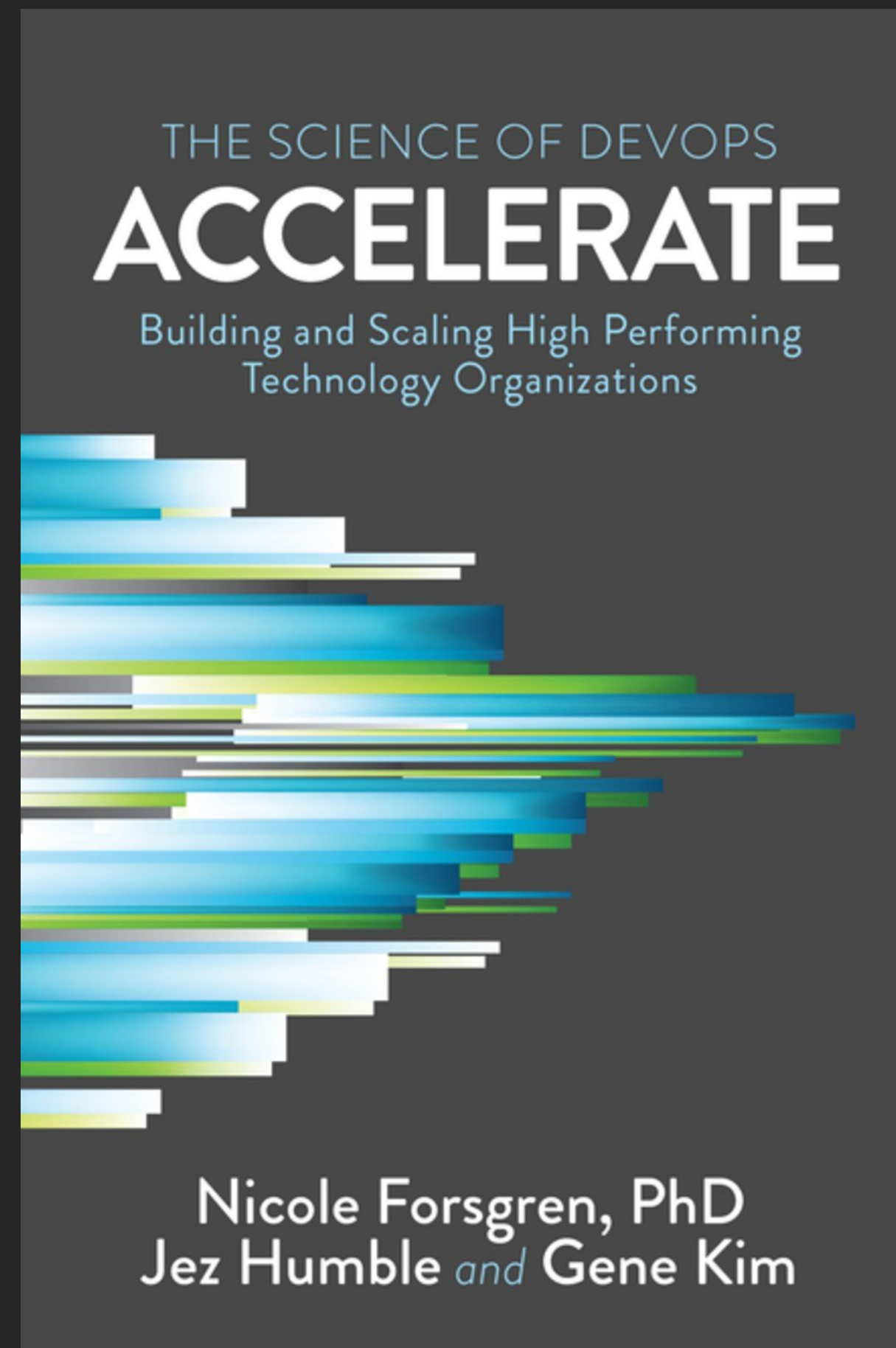
alternative?



Accelerate

Nicole Forsgren, Jez Humble,
Gene Kim

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



Generative
organizations focus on
the **mission**. How do we
accomplish our **goal**?

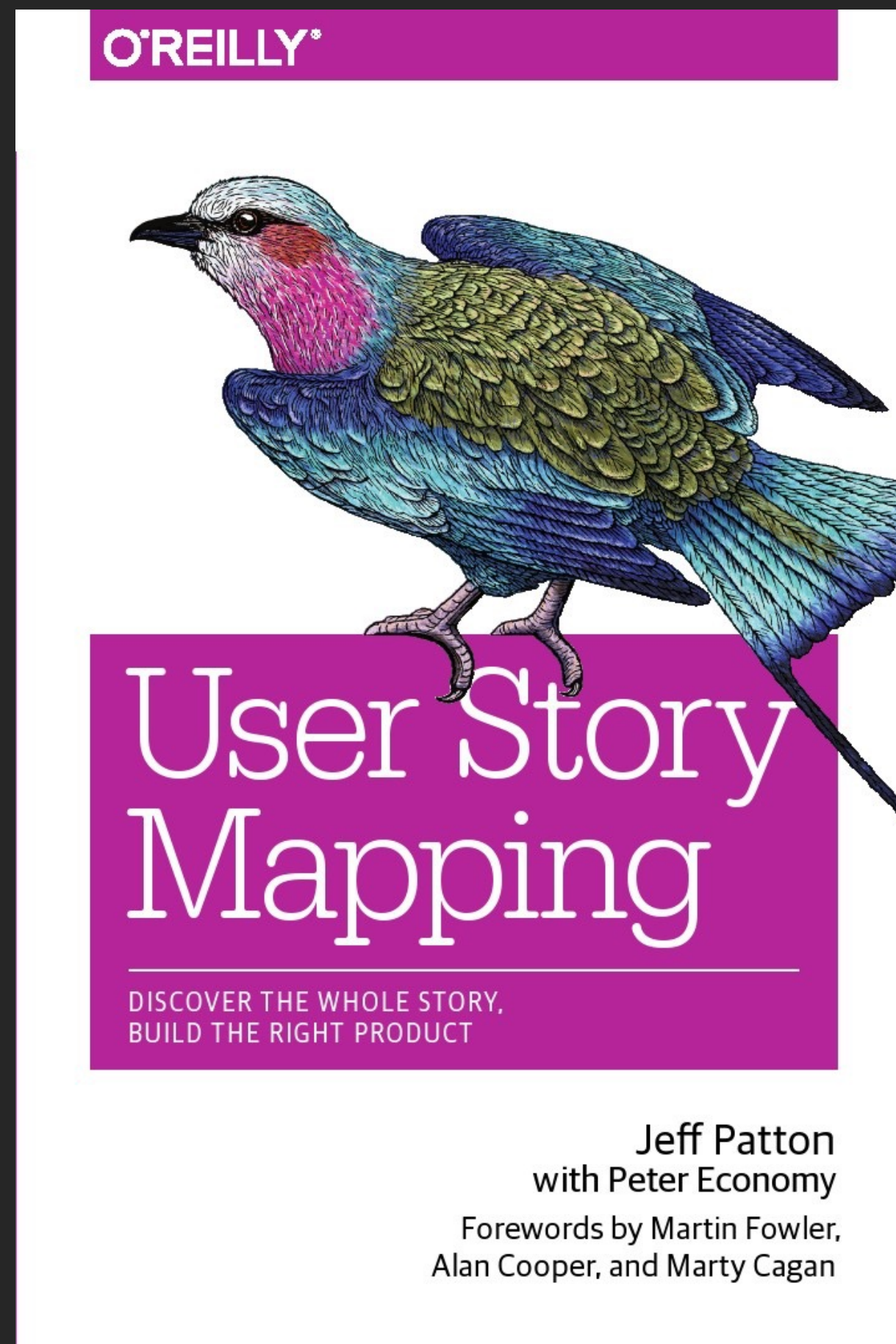
Accelerate

What's the

client's goal?

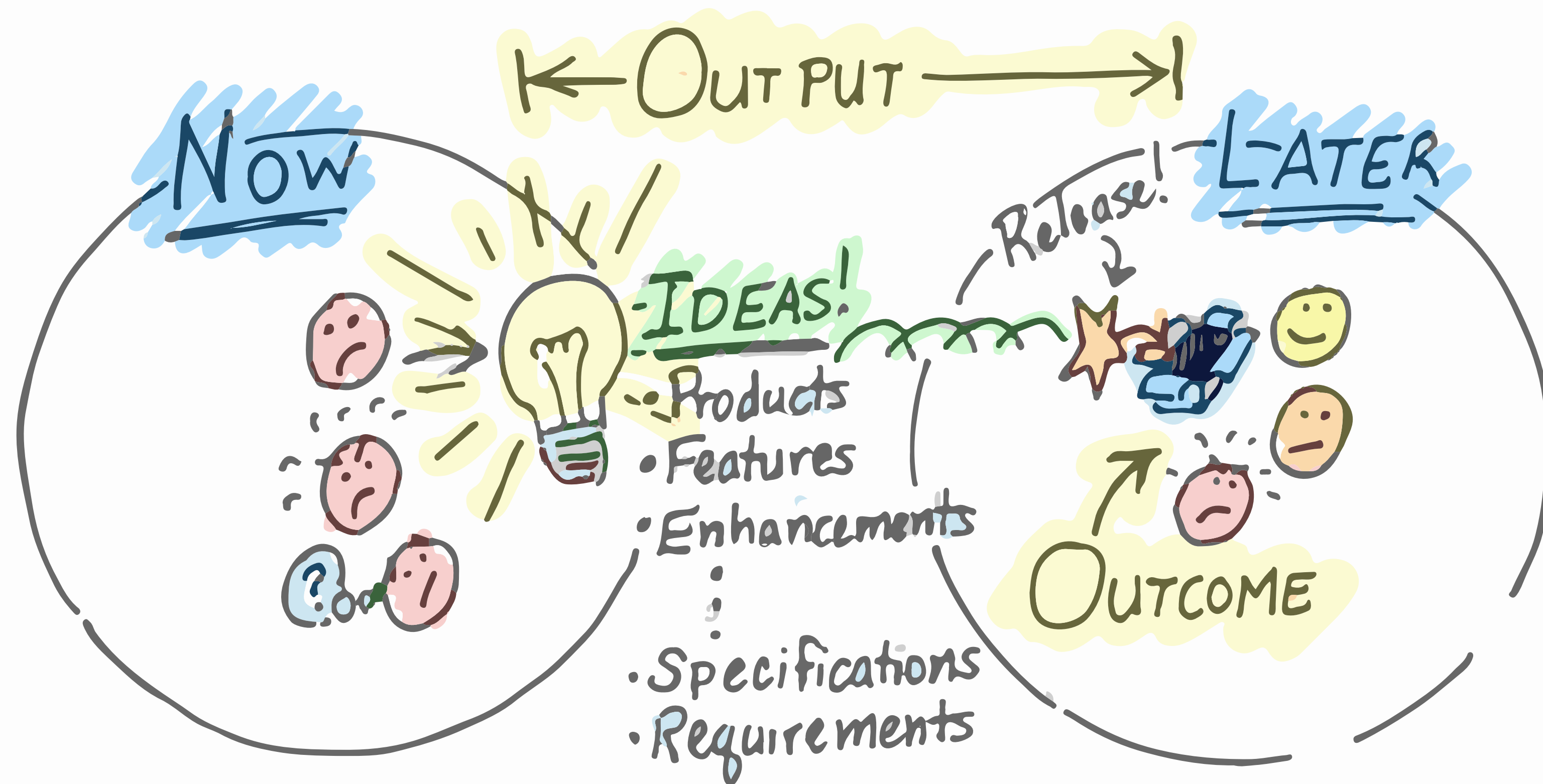
Why

do we build products?



User Story Mapping

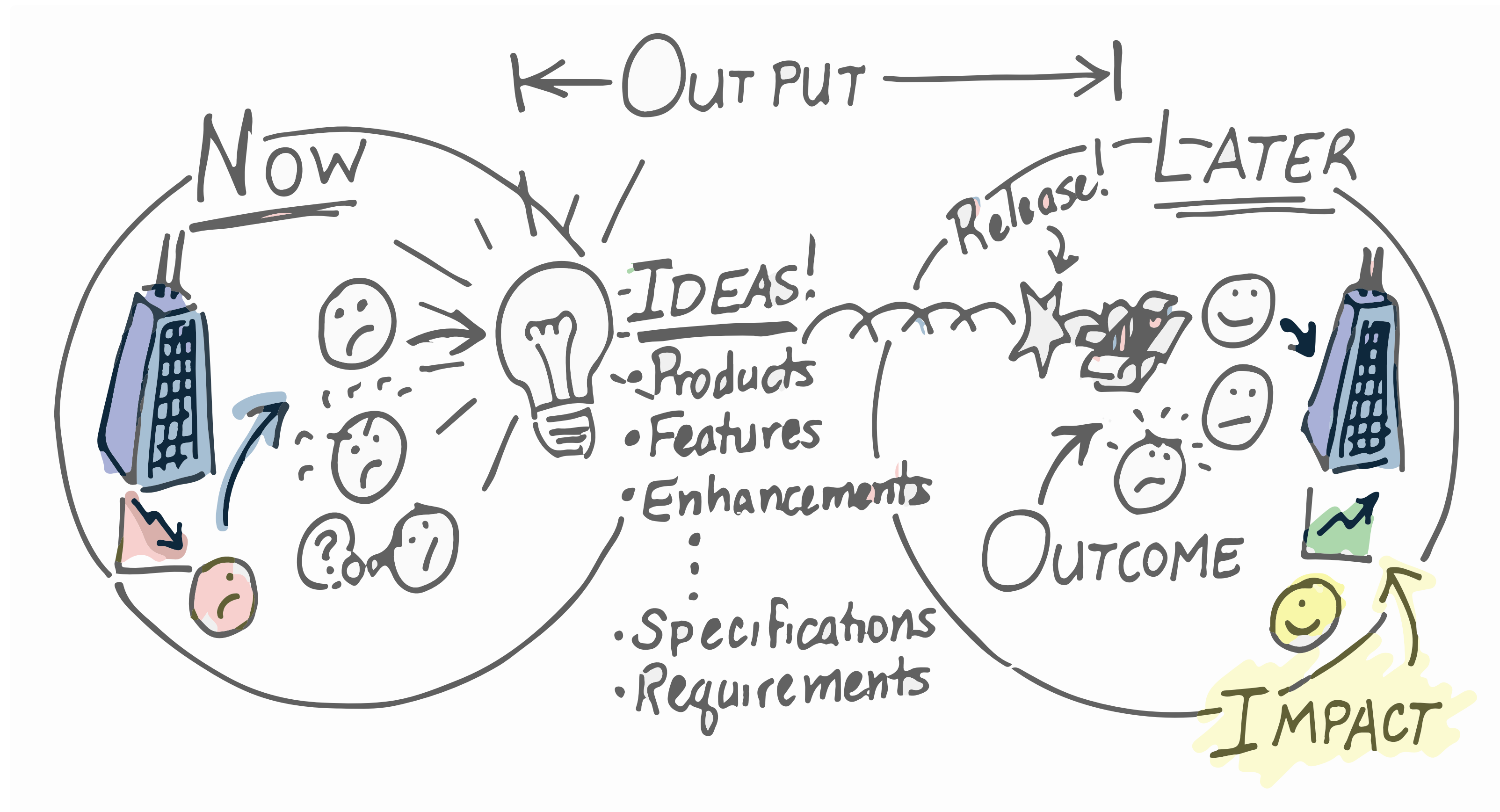
Jeff Patton



“

Your company can't get what it wants unless your customers and users get something they want.

Jeff Patton





Supplier



Client



Customers



Supplier



Client



Customers

Vision Statement

Kurze Beschreibung der Vision/Idee



Zielgruppe

Welches Marktsegment / Ziel- oder Nutzergruppe soll das Produkt ansprechen?

Dieser Punkt kann auch in Form einer Persona beschrieben sein.

Bedürfnisse/Probleme

Welche Bedürfnisse der Zielgruppe werden befriedigt?

In welcher Form schafft das Produkt Wert für den Benutzer oder empfindet er es als wertvoll?

Welche Emotionen weckt das Produkt?

Top Features

Welches sind die drei bis fünf wichtigsten Funktionen?

Gibt es ein Alleinstellungsmerkmal (USP)?

Geschäftsmodell

Wie wird mit dem Produkt Geld verdient?

Wer sind die zahlenden Kunden?

Was ist der Kunde bereit zu bezahlen und wie bezahlt er?

Konkurrenz

Gibt es Organisationen am Markt, die ein ähnliches Produkt anbieten?

Differenznutzen

Was bietet das Produkt mehr, weniger oder anders gegenüber vergleichbaren Produkten?

Understand

what's the real

customer need ...

... and don't just

implement a wish

list of features.

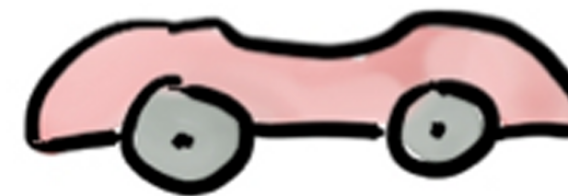
Not like this....



1



2



3



4

Like this!



1



2



3



4



5

Small

Experiments

Batch Sizes

Every Release is a

working product.

Use

**Continuous
Integration!**

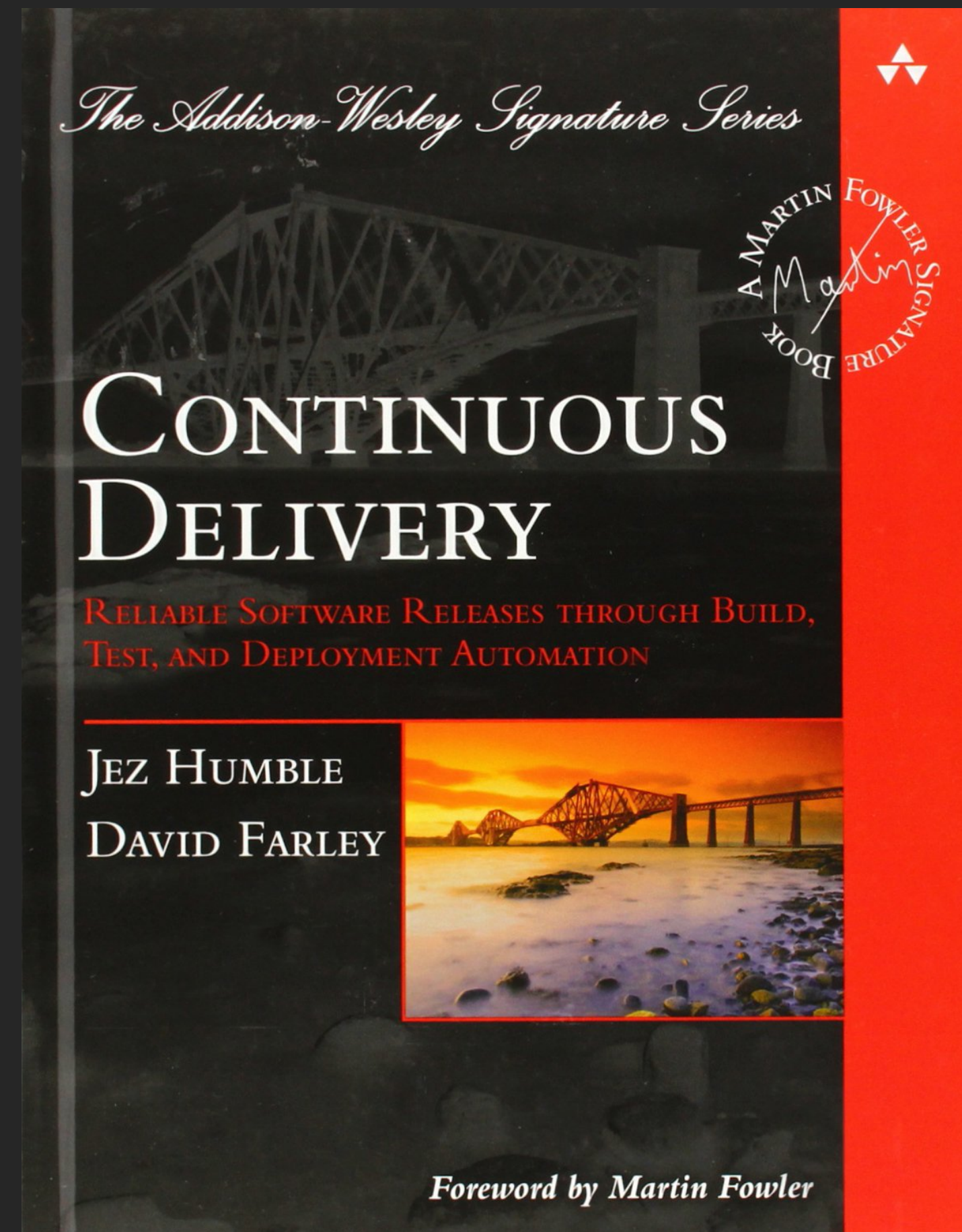
We do!!



Really? 🤔

Trunk Based Development?

#NoFeatureBranches



Continuous Delivery

Jez Humble,
David Farley

Use

**Continuous
Delivery!**

Application is **always** in a
releasable state.



Repo

+



Build

=

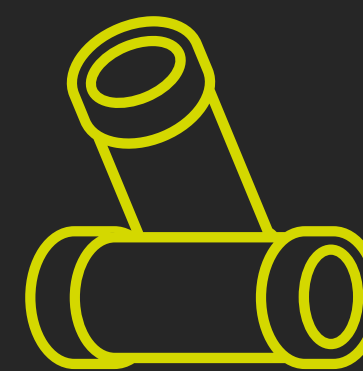


Executable



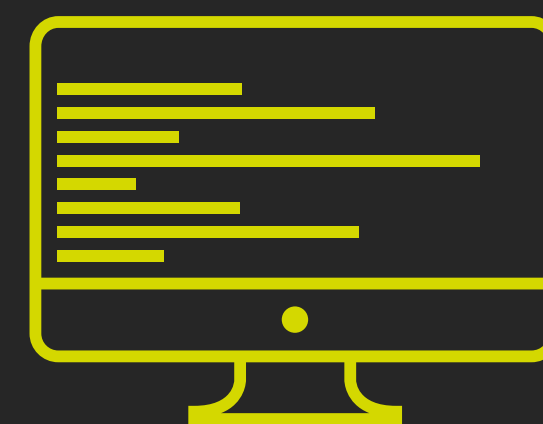
Repo

+



Pipeline

=

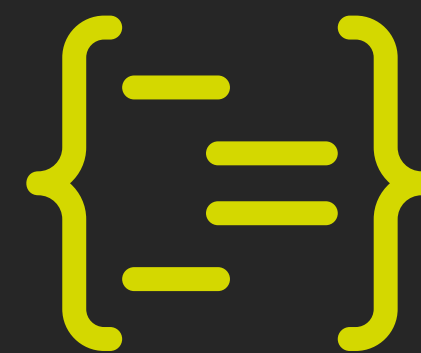


running product

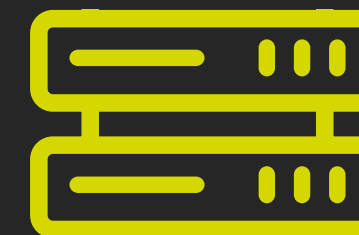
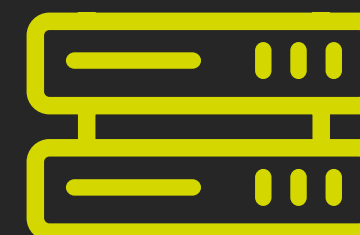
Let production be

**„yet another
environment“.**

Application



Infrastructure



t

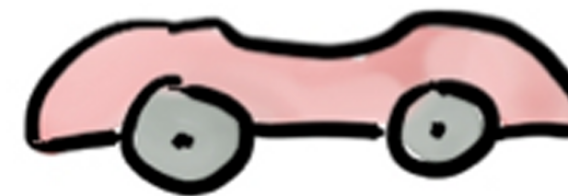
Not like this....



1



2



3



4

Like this!



1



2



3



4



5

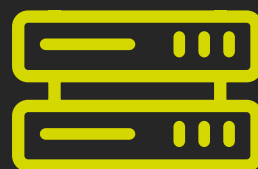
Run your application in

production.

Code



Infrastructure



t

Build

metrics & alerting

in.

**Eat your own
dog food.**

Learn your application footprint by

**early performance
tests.**

We want to ship continuously, but we
have to

wait for the client's ...

designers

IT

sales team

Marketing

Content

Authors

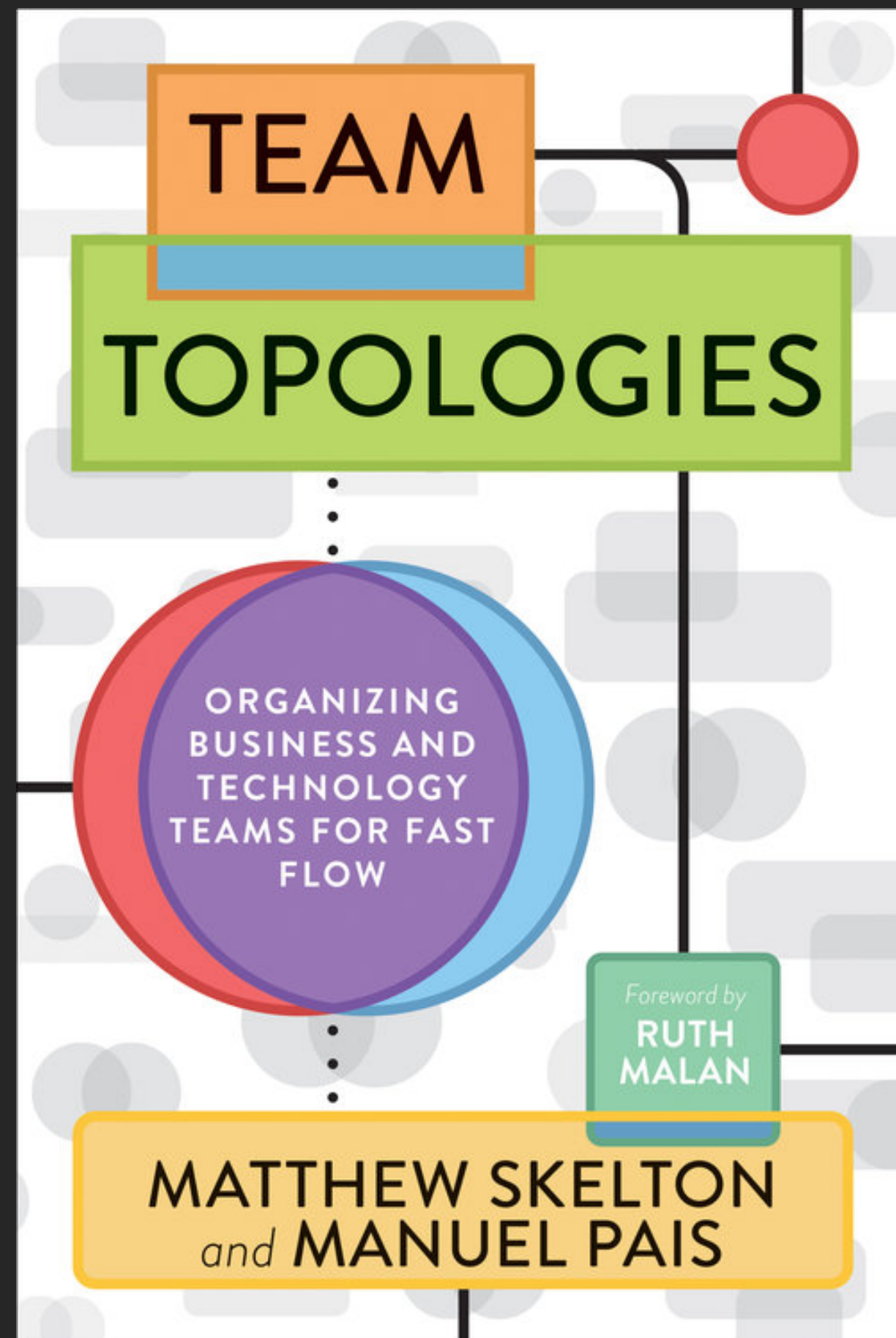
Legal



Use DevOps

collaboration models

to **avoid silos.**



Team Topologies

Matthew Skelton &
Manuel Pais



• Dev • Ops

Embedding

Inspired by devopstopologies.com



• Dev • Ops

Embedding

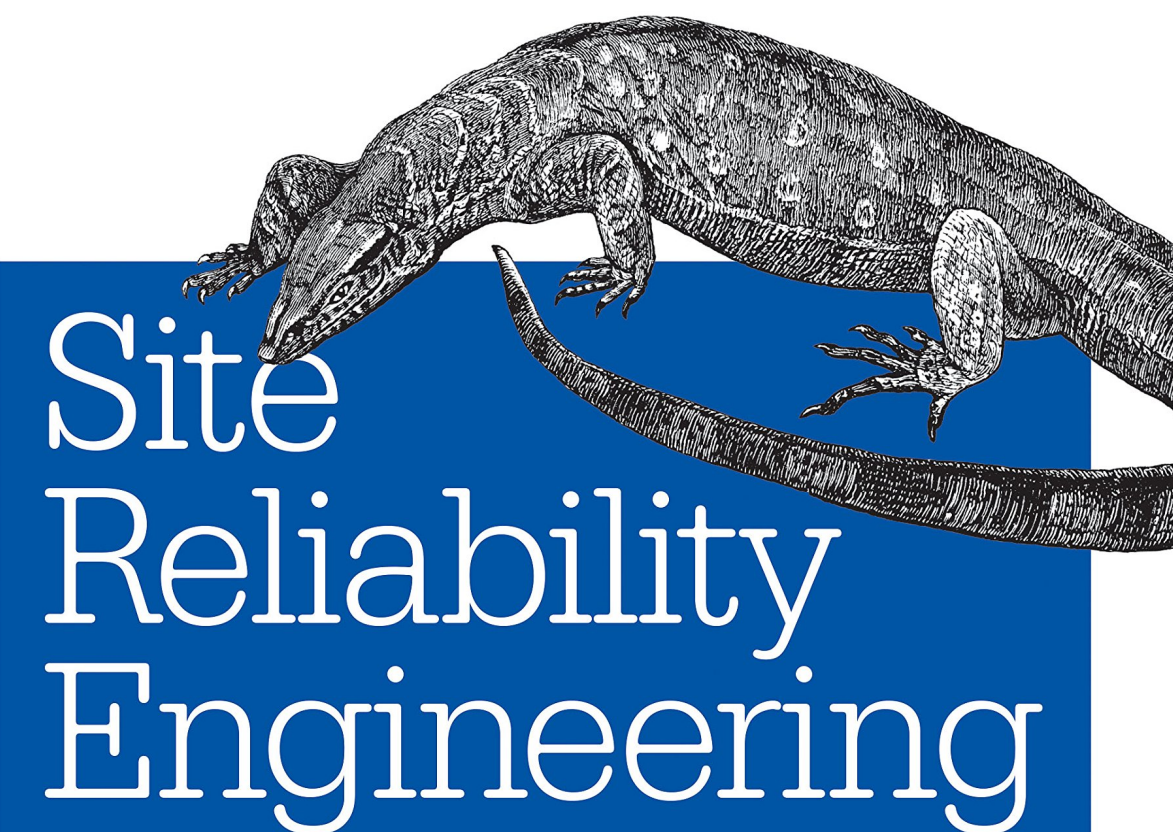


• Dev • DevOps • Ops

x as a Service

Inspired by devopstopologies.com

O'REILLY®



HOW GOOGLE RUNS PRODUCTION SYSTEMS

Edited by Betsy Beyer, Chris Jones,
Jennifer Petoff & Niall Richard Murphy

Site Reliability Engineering



• Dev • Ops

Embedding



• Dev • DevOps • Ops

x as a Service



• Dev • DevOps • Ops

Collaboration

Inspired by devopstopologies.com

Use DevOps collaboration models for

**handover &
onboarding.**



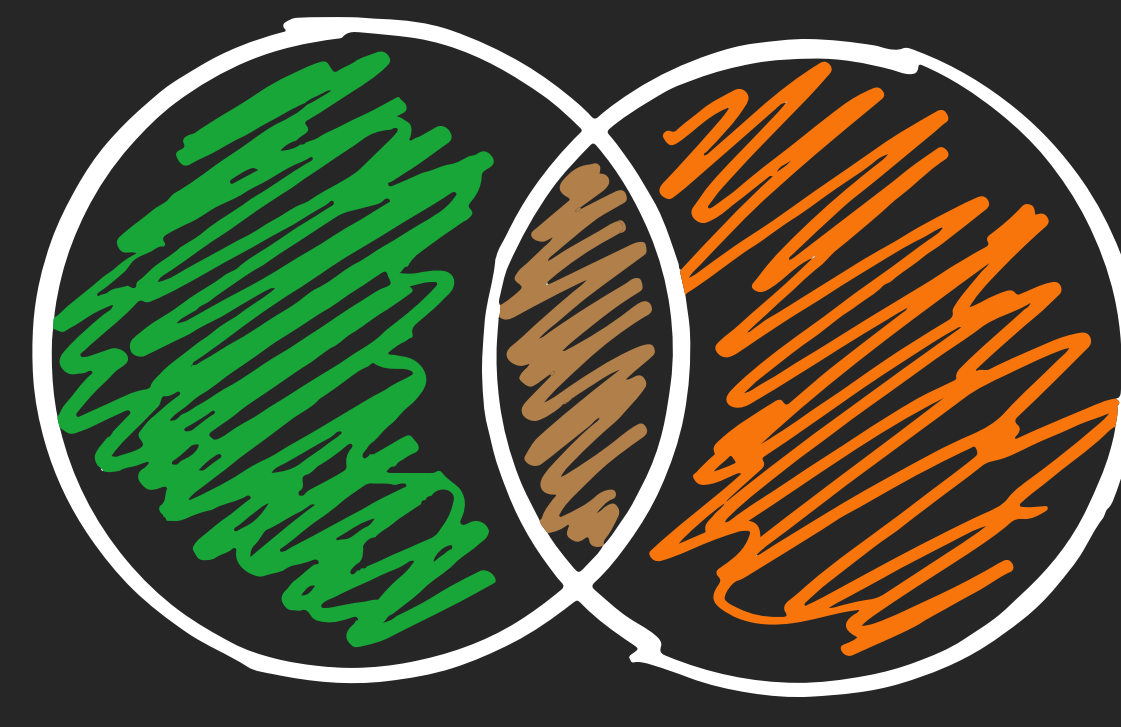
• Dev • Ops

Embedding



• Dev • DevOps • Ops

x as a Service



• Dev • DevOps • Ops

Collaboration

Inspired by devopstopologies.com

Use

blameless postmortems

to **share knowledge**

about the application with the client.

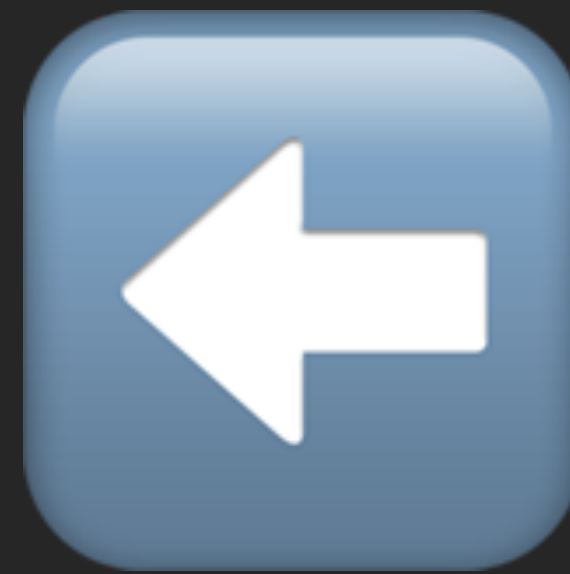
Use **basic**

DevOps practices ...

Make **work visible**

... also to the client.

Shift left.

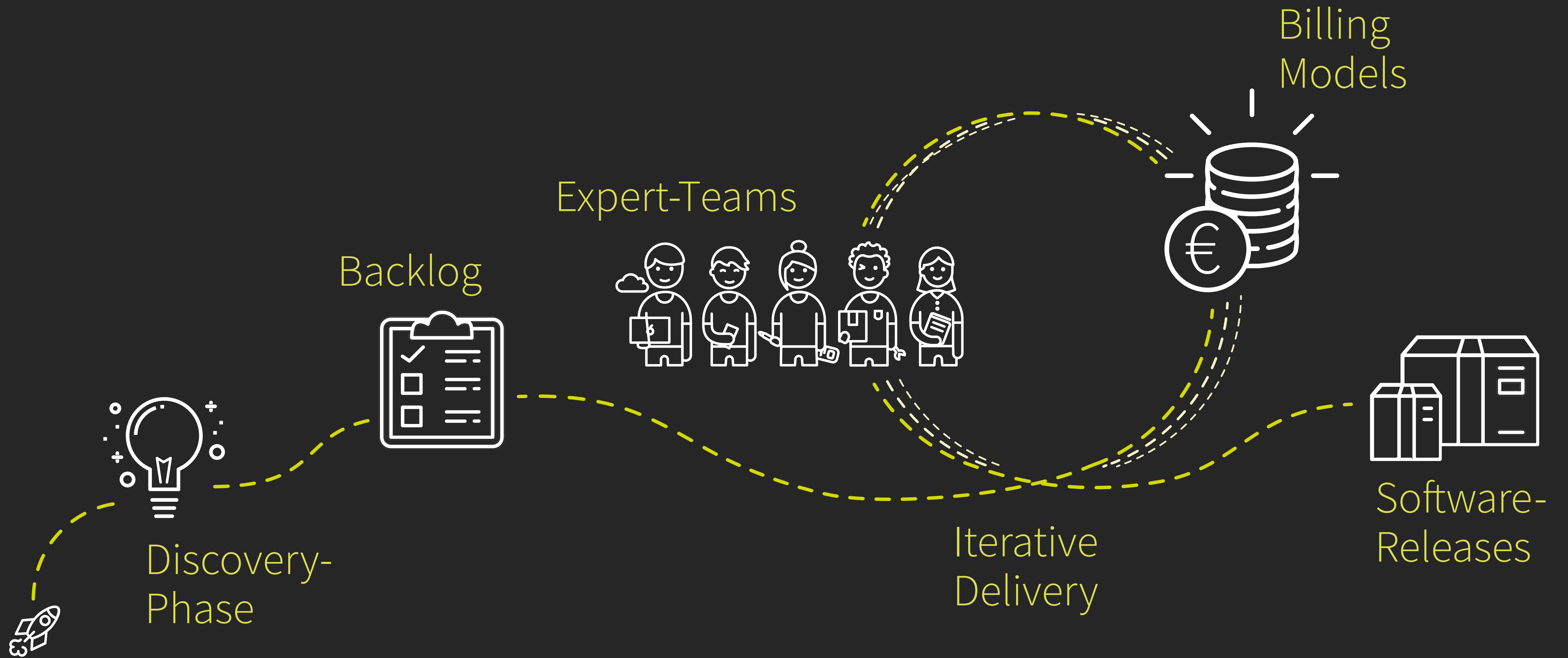


Everybody is

responsible.

Continuous

Improvement 🦾



Interested?

cosee.biz

Konstantin Diener | cosee GmbH

konstantin.diener@cosee.biz |  @onkelkodi



Picture credits:

Boat: <https://www.shutterstock.com/de/g/solovushka>

Silos: <https://www.shutterstock.com/de/g/davotibarna>

Gated Bridge: <https://www.istockphoto.com/de/portfolio/vesilvio>

