

Achieving unicorn scale

Using Team Topologies at cinch

Toli Apostolidis - Engineering Practice Lead

@apostolis09

Andy Norton - Head of Engineering Practice

@andyjnorton



Who are we?

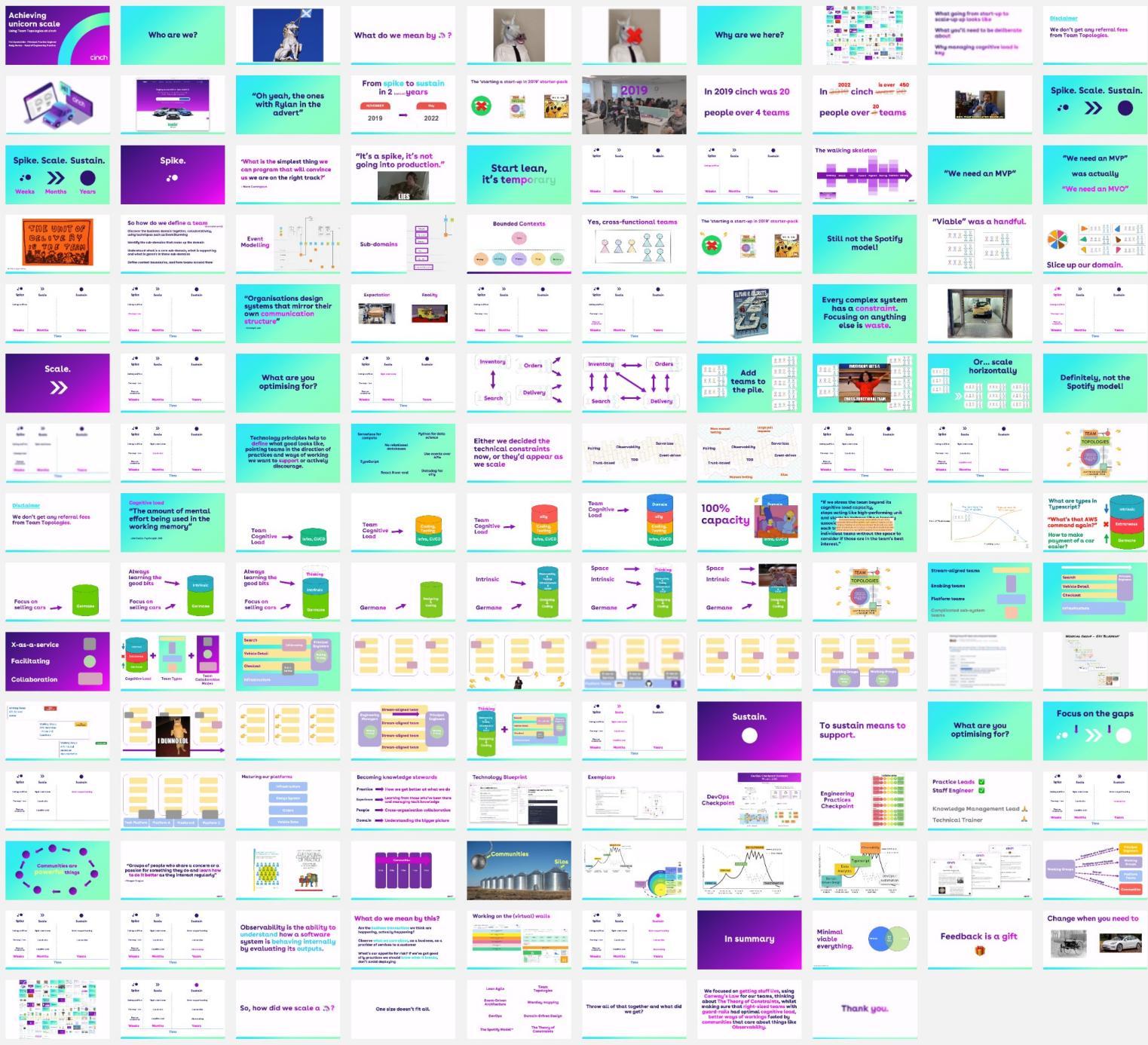


What do we mean by  ?





Why are we here?



What going from start-up to scale-up up looks like

What you'll need to be deliberate about

Why managing cognitive load is key

Disclaimer

**We don't get any referral fees
from Team Topologies.**



cinch.co.uk

Find a car ▾ Car finance Part exchange How cinch works ▾ News & reviews ▾ My profile 

Buying a car online - just cinch it.

Buy, finance and part exchange. Free home delivery or collection. 14-day money-back guarantee.

Select make ▾ | Select model ▾ | Search all 11,281 cars

Not sure where to start? Try our [Help Me Choose tool](#)



Chat

Check out our 12,909 reviews


Trustpilot

“Which one are we?”

**“Oh yeah, the ones
with Rylan in the
advert”**

From spike to sustain in 2 (and a bit) years

NOVEMBER

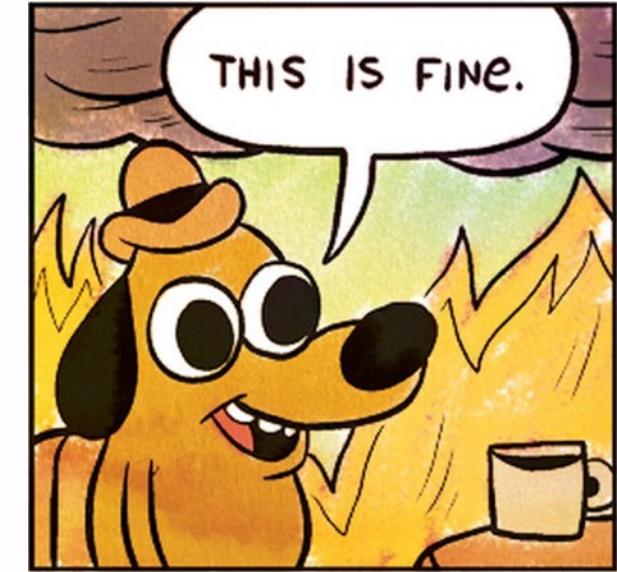
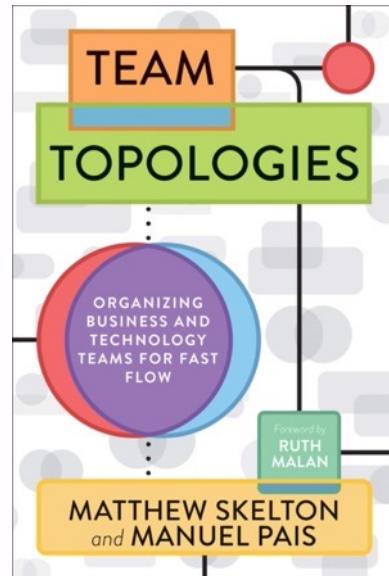
2019

May

2022



The 'starting a start-up in 2019' starter-pack



2019



In 2019 cinch was 20
people over 4 teams

2022

is over 450

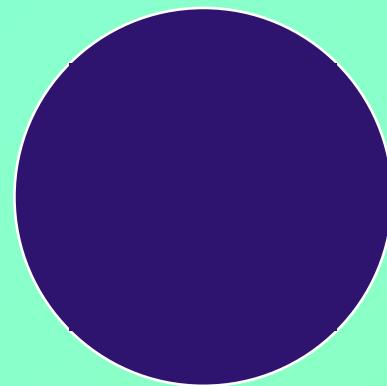
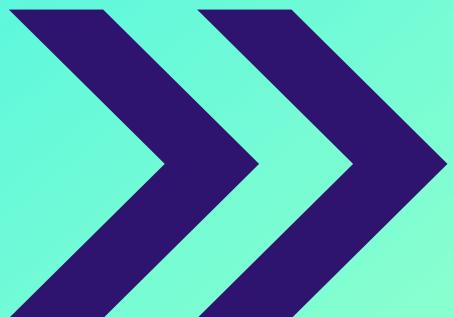
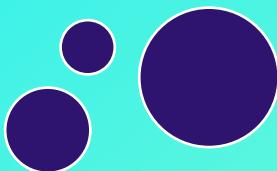
In ~~2019~~ cinch ~~was~~ 20

20

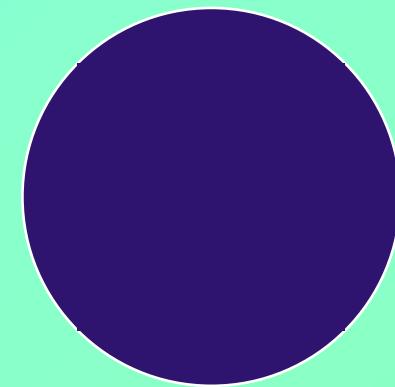
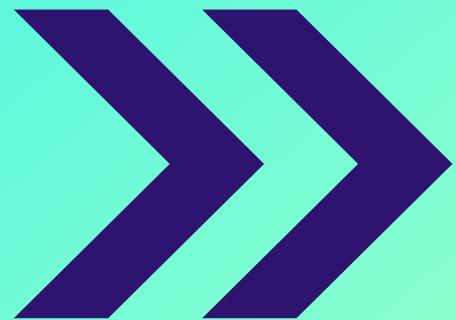
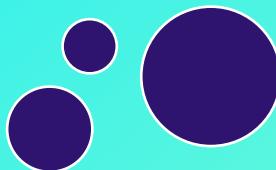
people over ~~4~~ teams



Spike. Scale. Sustain.



Spike. Scale. Sustain.



Weeks

Months

Years

Spike.



**'What is the simplest thing we
can program that will convince
us we are on the right track?'**

- Ward Cunningham

**“It's a spike, it's not
going into production.”**



**Start lean,
it's temporary**



Spike



Scale



Sustain



Weeks

Months

Years

Time



Spike



Scale



Sustain

Getting stuff live

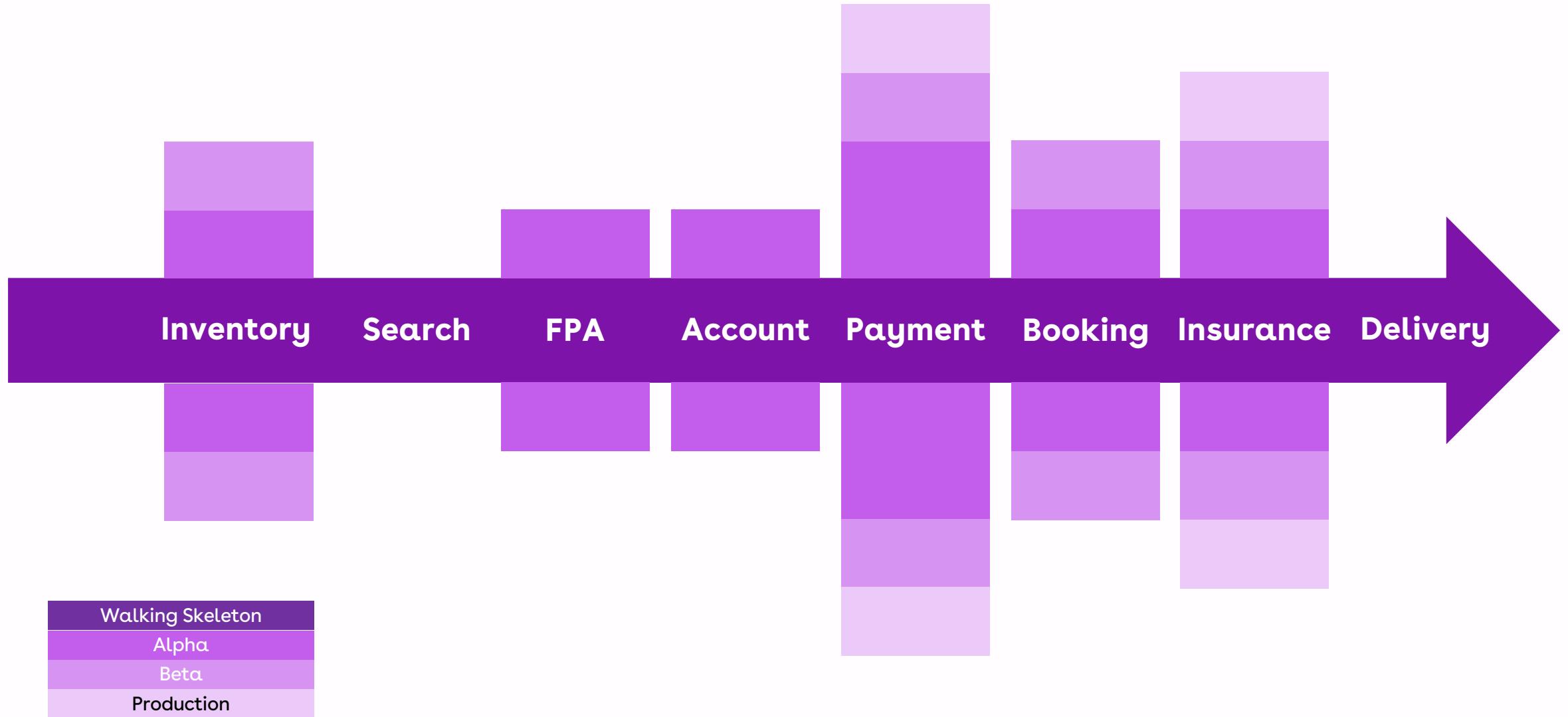
Weeks

Months

Years

Time

The walking skeleton



"We need an MVP"

"We need an MVP"

was actually

"We need an MVO"

THE UNIT OF
DELIVERY
IS THE TEAM



So how do we define a team

(in an ideal world)

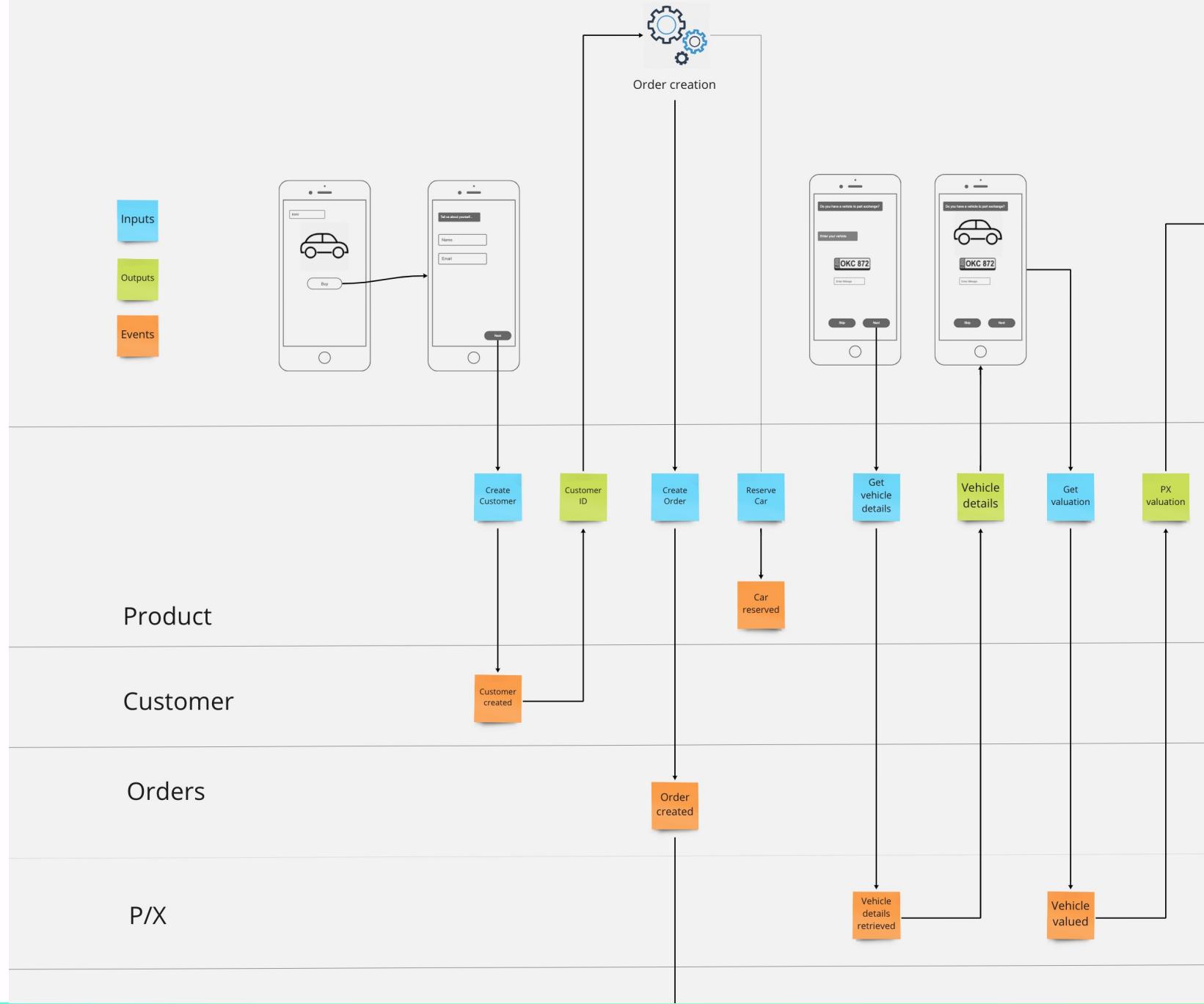
Discover the business domain together, collaboratively, using techniques such as EventStorming and Event Modeling

Identify the sub-domains that make up the domain

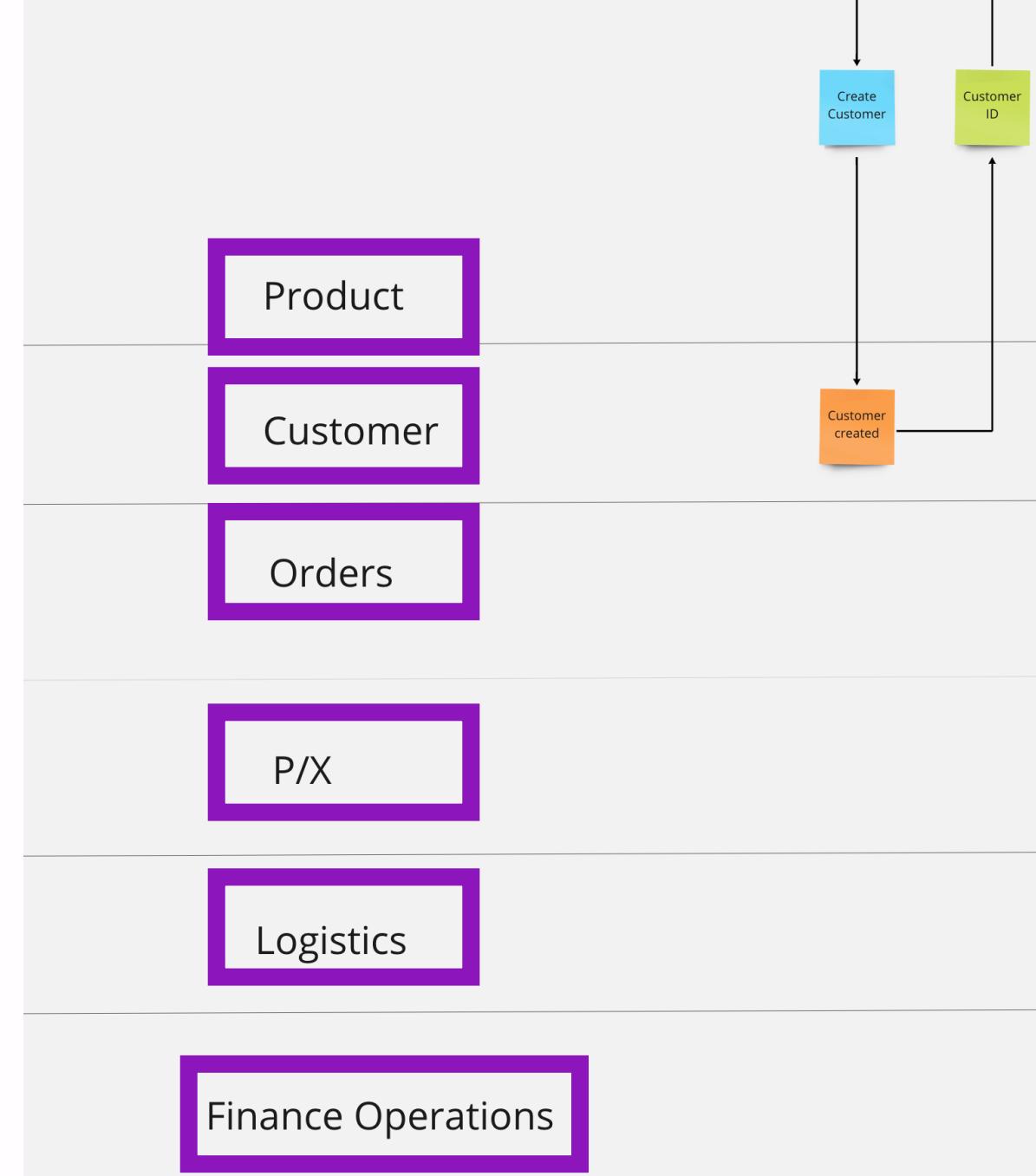
Understand what is a core sub-domain, what is supporting and what is generic in these sub-domains

Define context boundaries, and form teams around them

Event Modeling



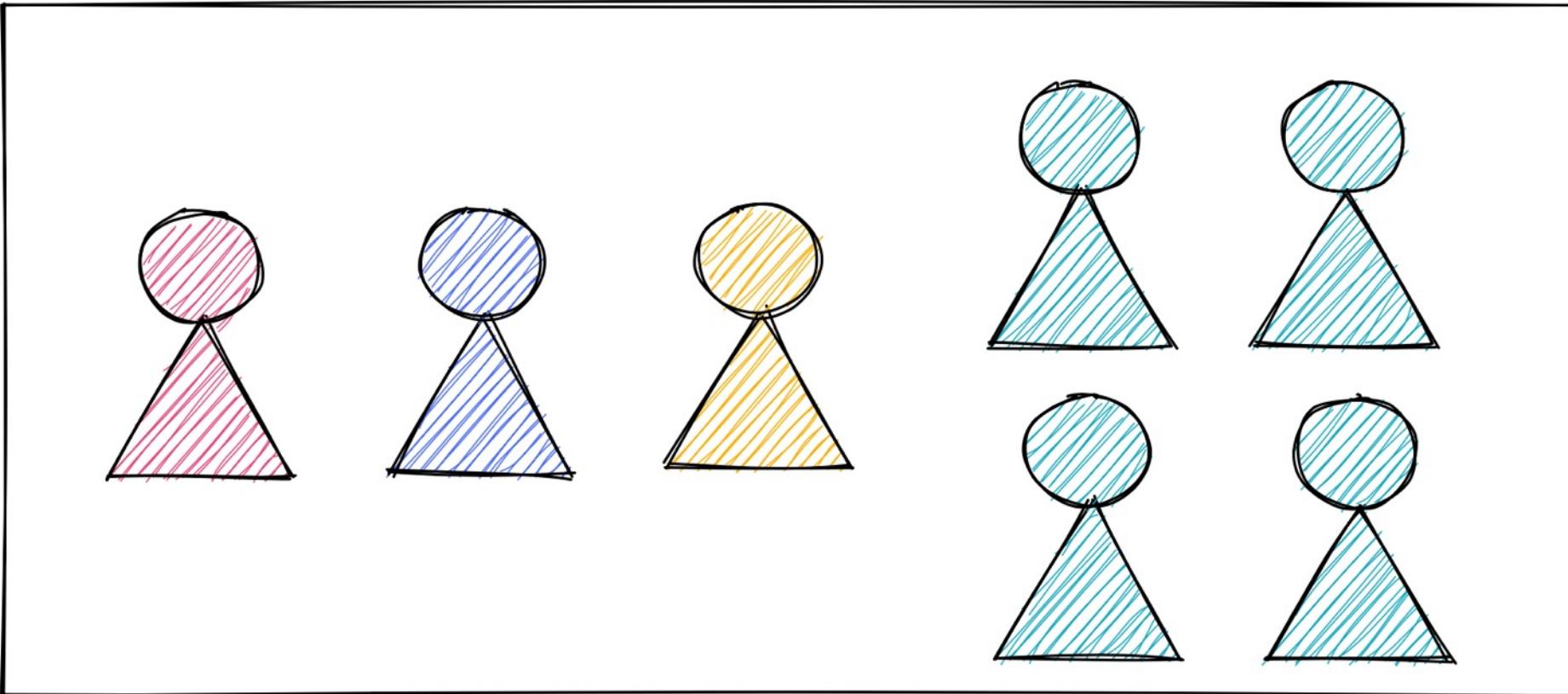
Sub-domains



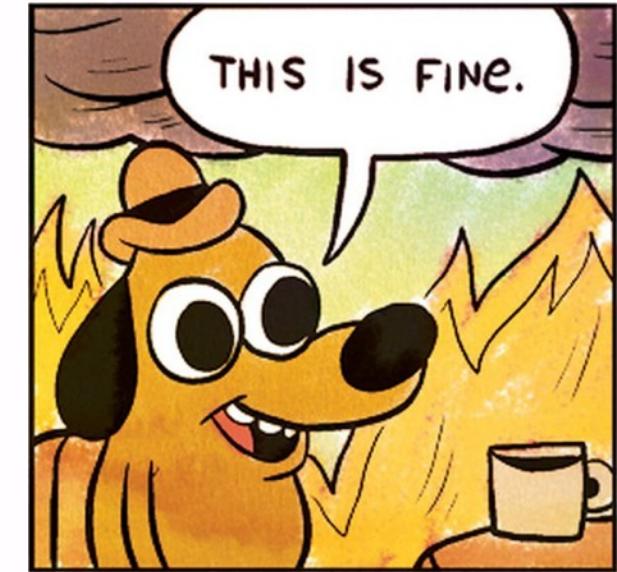
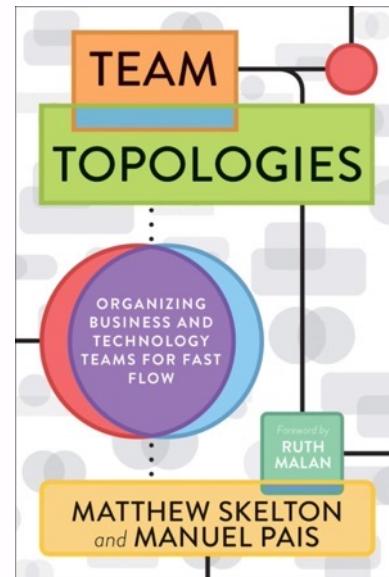
Bounded Contexts



Yes, cross-functional teams

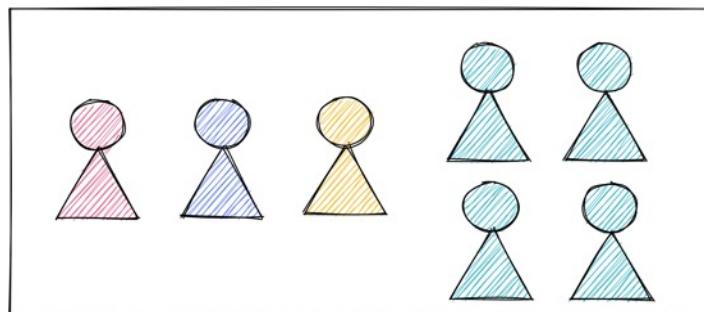
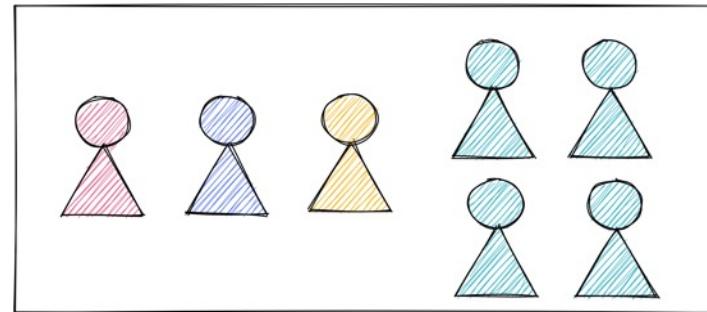
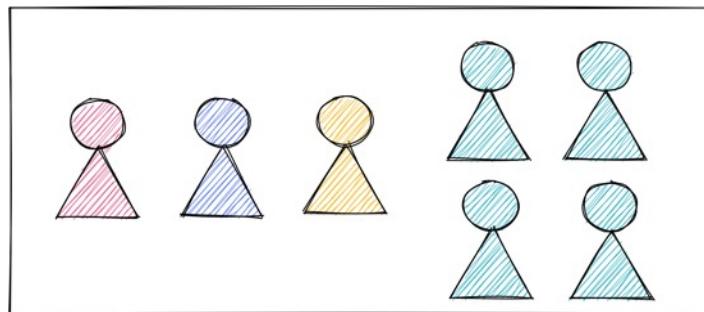
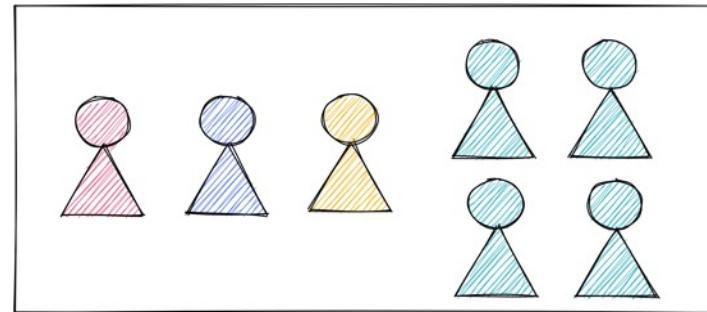
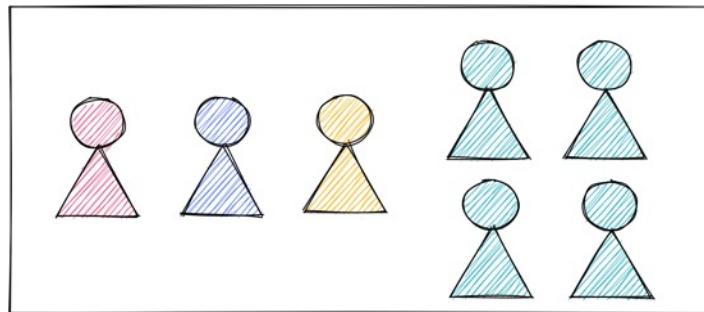


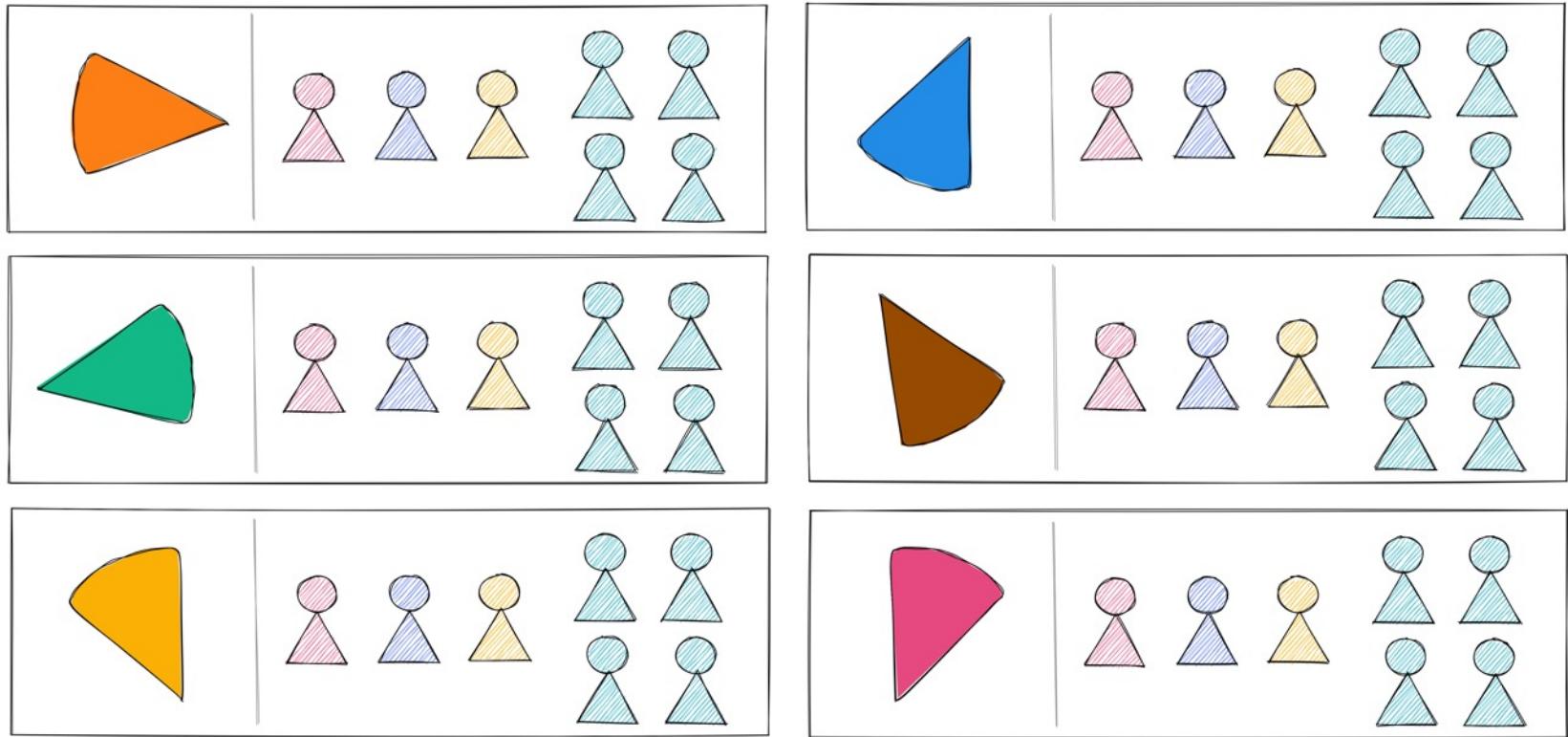
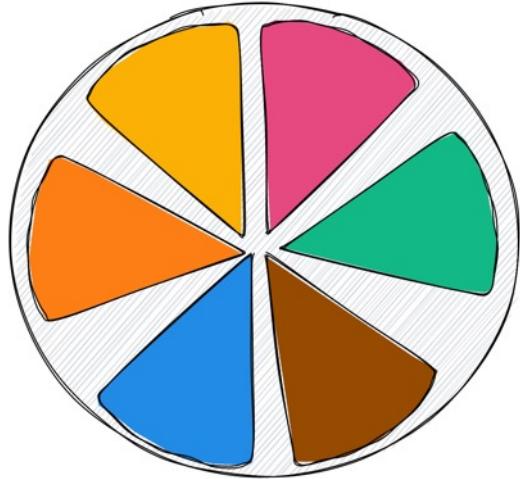
The 'starting a start-up in 2019' starter-pack



**Still not the Spotify
model!**

“Viable” was a handful.





Slice up our domain.



Spike



Scale



Sustain

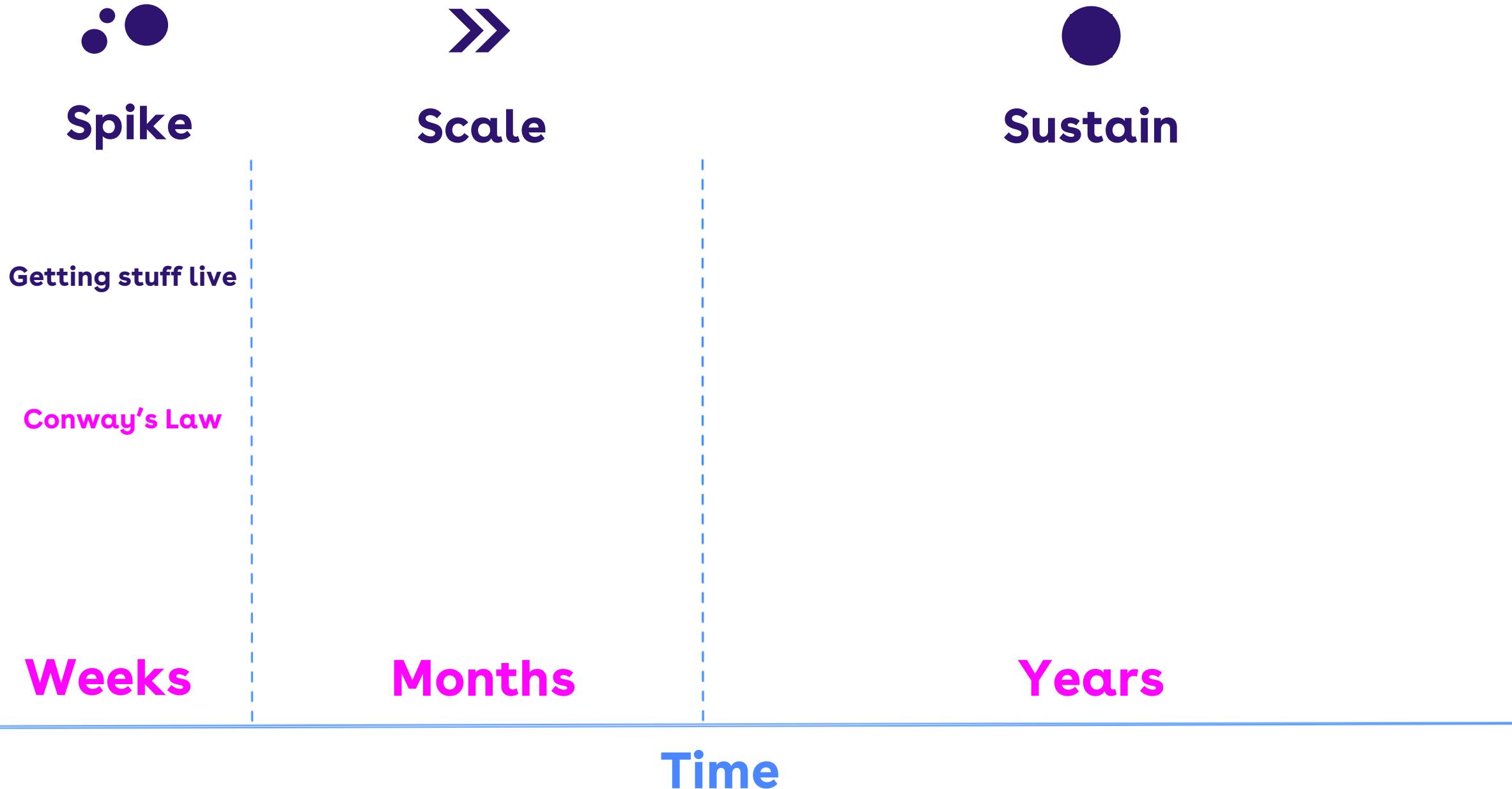
Getting stuff live

Weeks

Months

Years

Time



**“Organisations design
systems that mirror their
own communication
structure”**

- Conway's Law

Expectation



Reality





Spike

Scale

Sustain

Getting stuff live

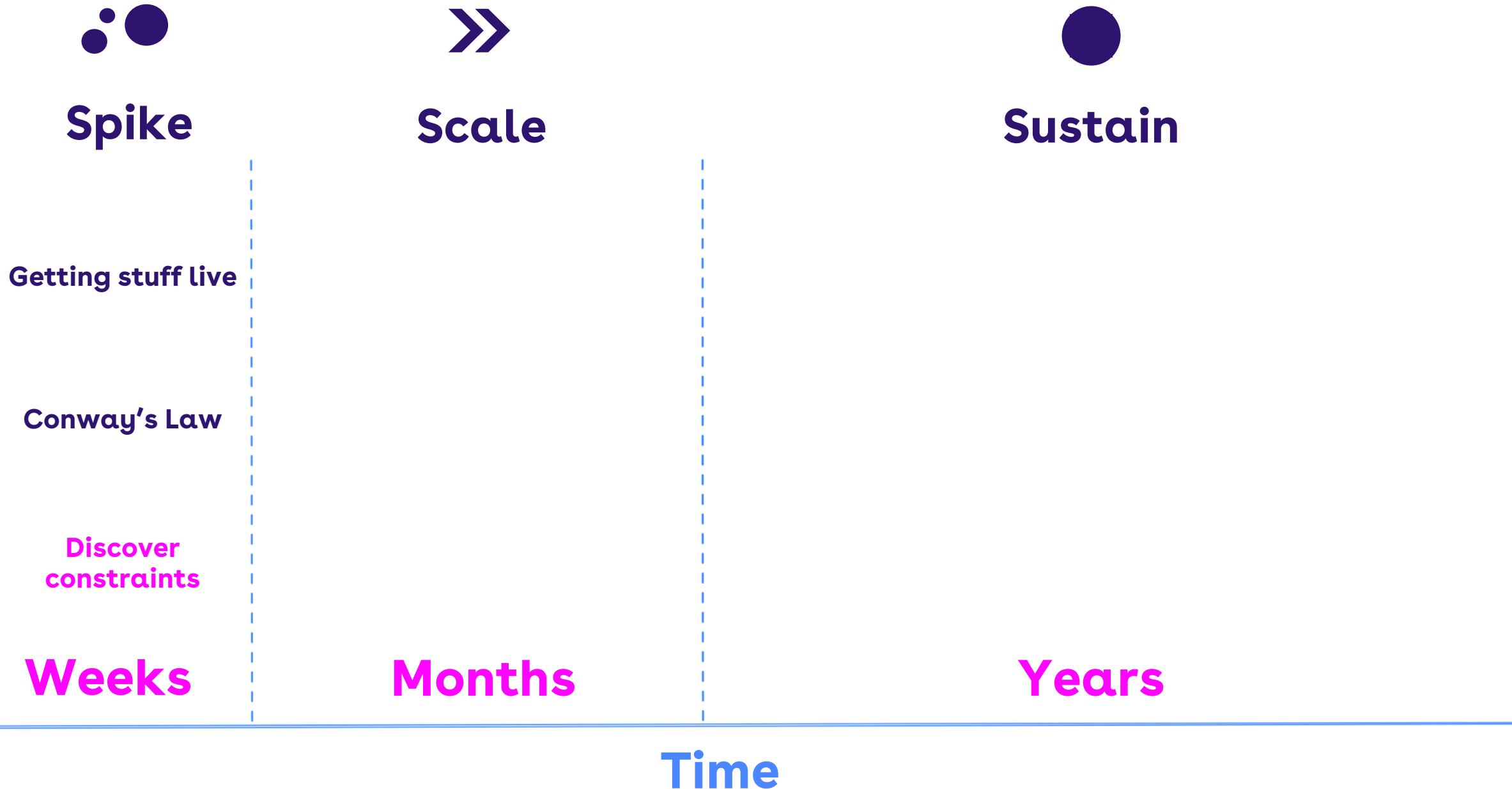
Conway's Law

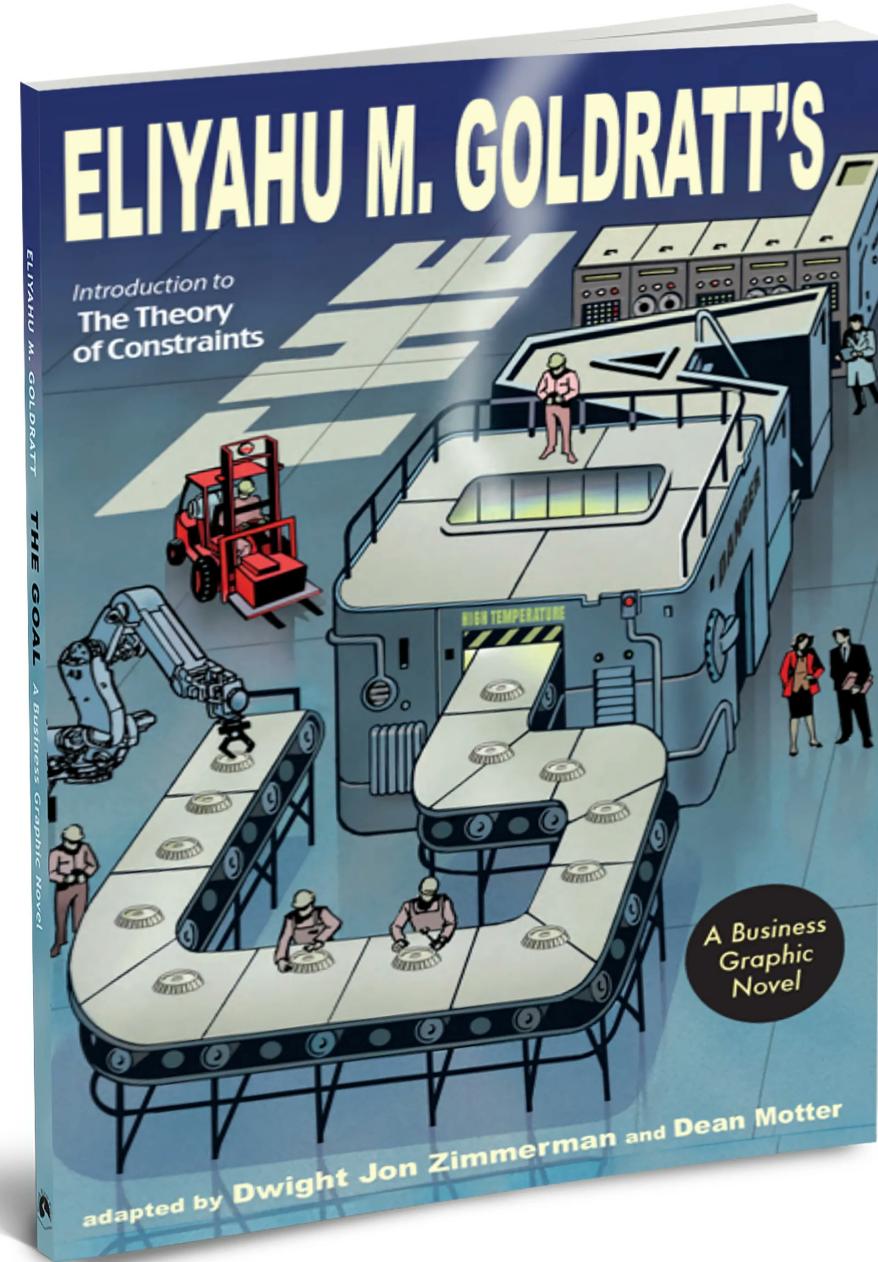
Weeks

Months

Years

Time





**Every complex system
has a constraint.**

**Focusing on anything
else is waste.**





Spike

Scale

Sustain

Getting stuff live

Conway's Law

Discover
constraints

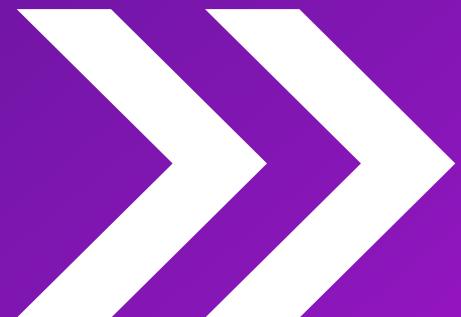
Weeks

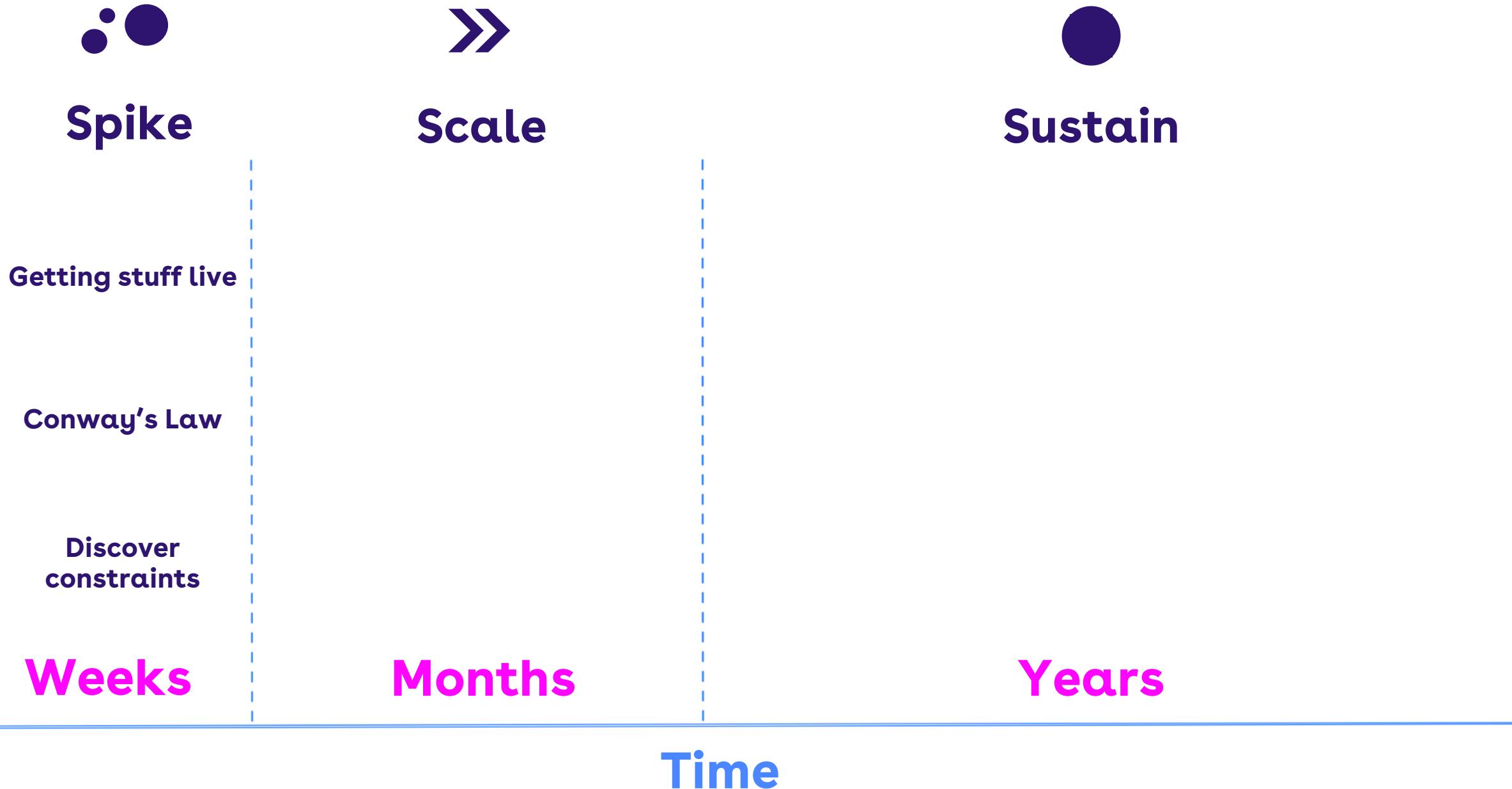
Months

Years

Time

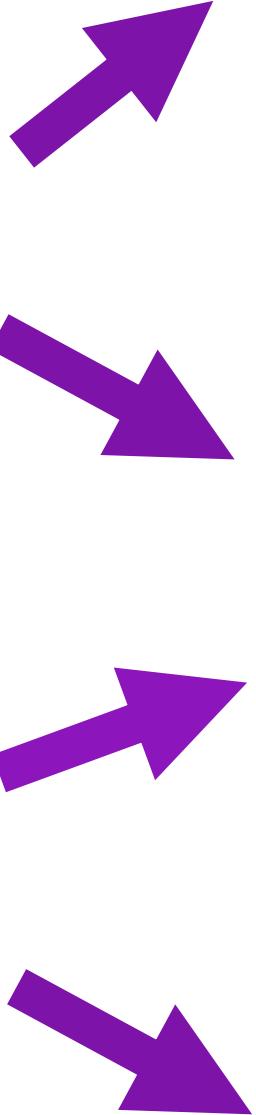
Scale.



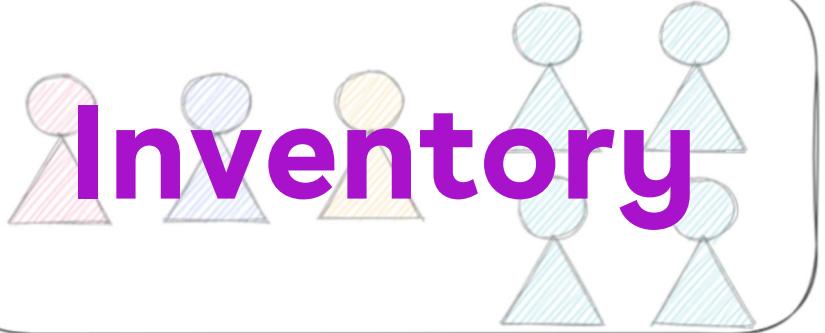


**What are you
optimising for?**





Inventory



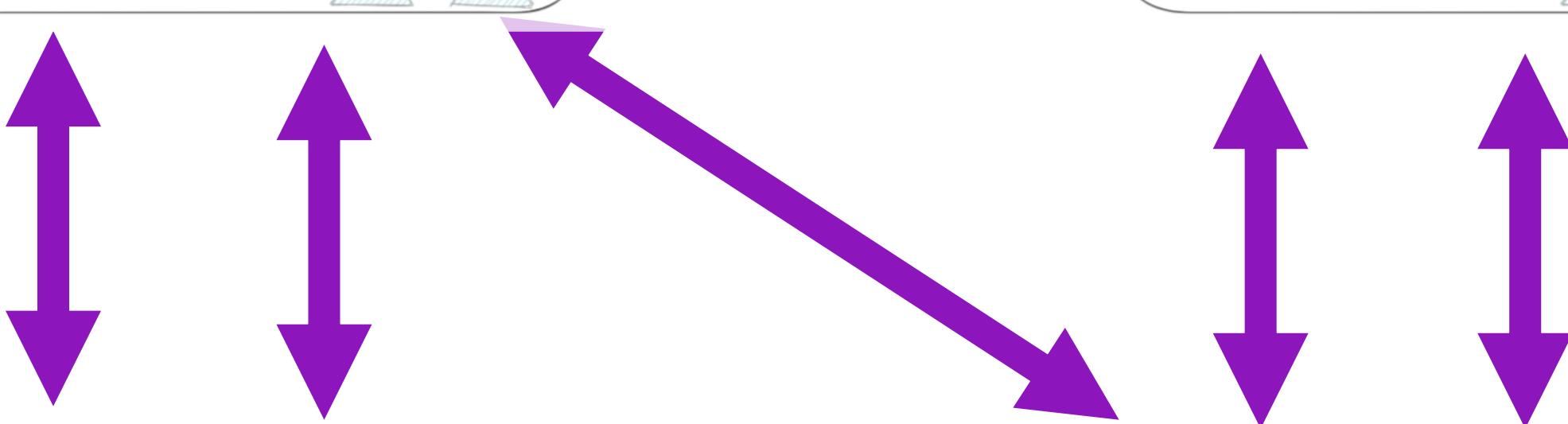
Orders



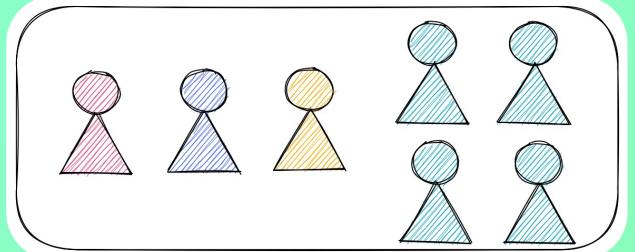
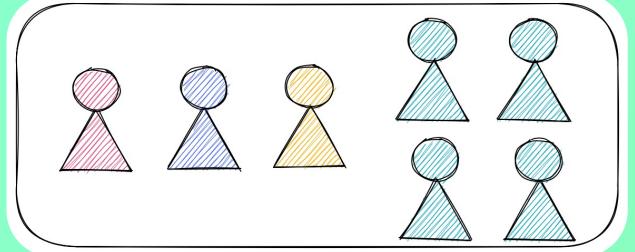
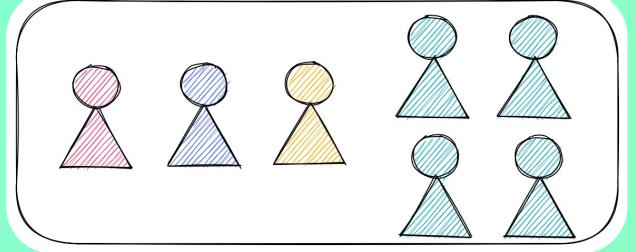
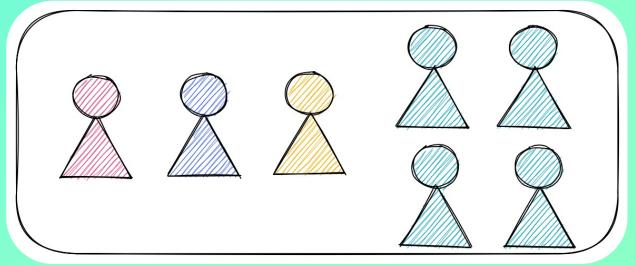
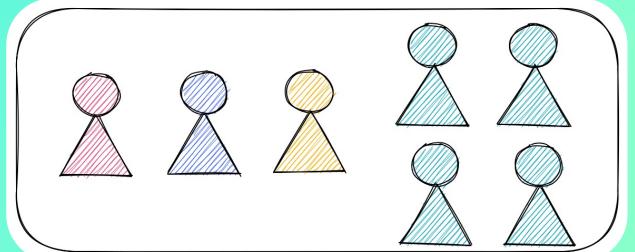
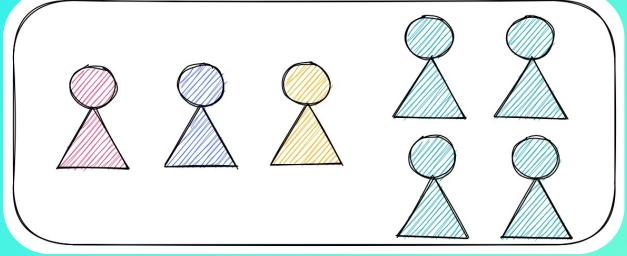
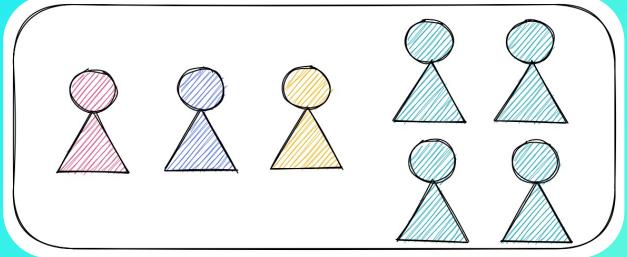
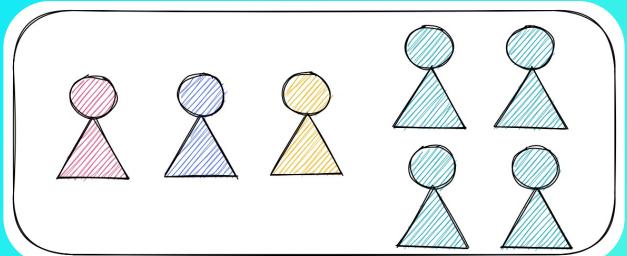
Search



Delivery



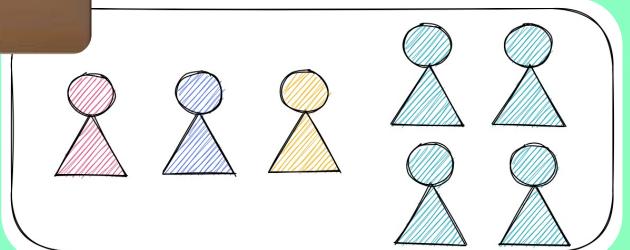
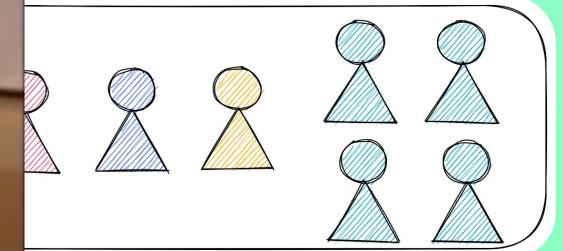
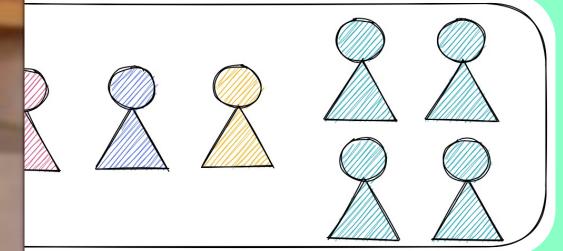
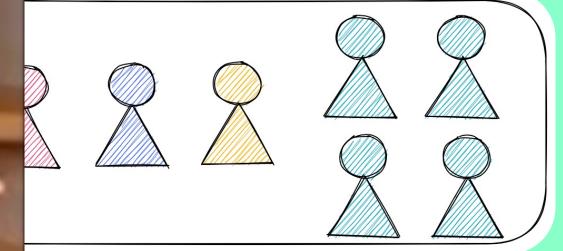
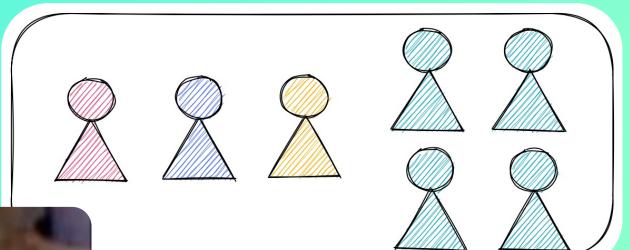
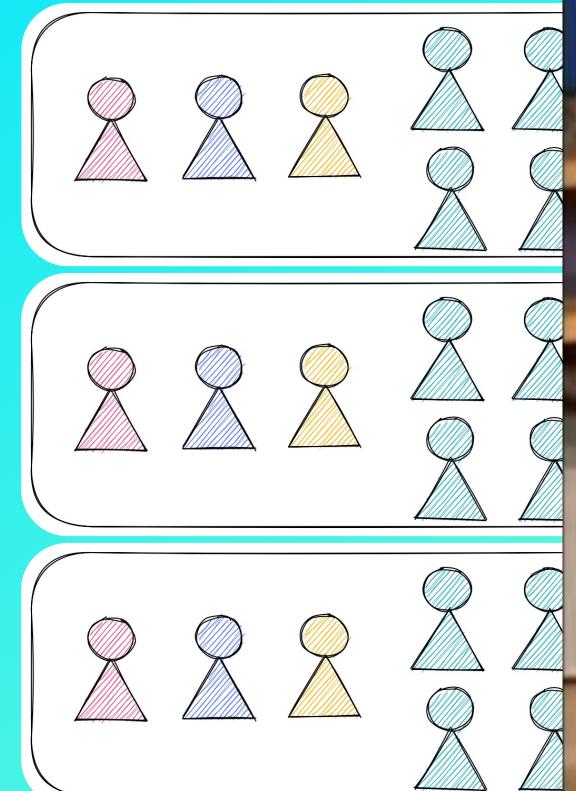
Add
teams to
the pile.



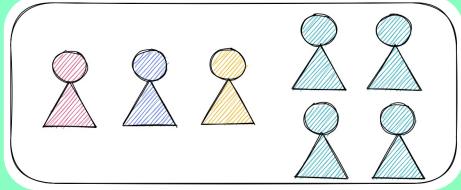
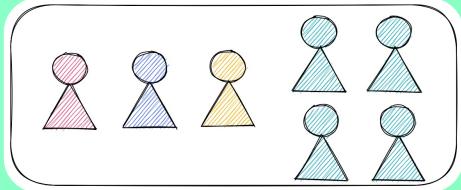
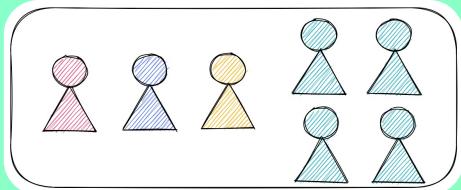
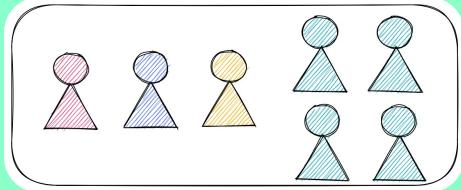
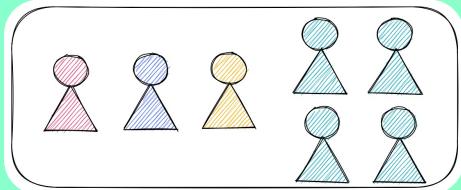
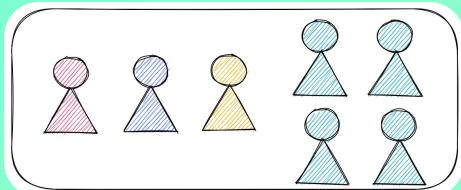
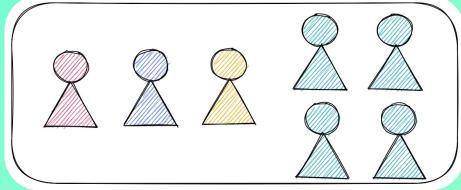
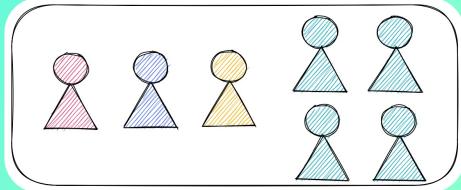
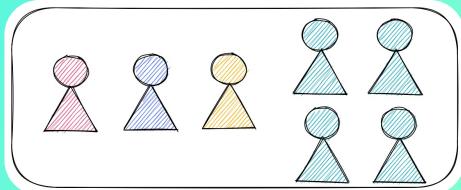
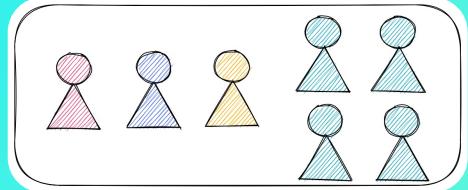
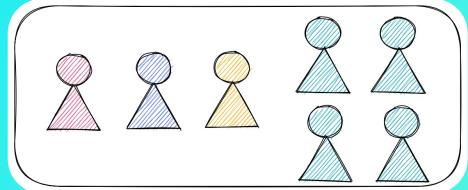
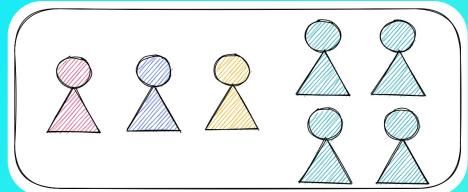
EVERYBODY GETS A

CROSS-FUNCTIONAL TEAM

imgflip.com



Or... scale horizontally



**Definitely, not the
Spotify model!**





Spike

Getting stuff live

Conway's Law

Discover constraints

Weeks



Scale

Right-sized teams

Guard-rails

Months



Sustain

Years

Time

Technology principles help to define what good looks like, pointing teams in the direction of practices and ways of working we want to support or actively discourage.

**Serverless for
compute**

**Python for data
science**

**No relational
databases**

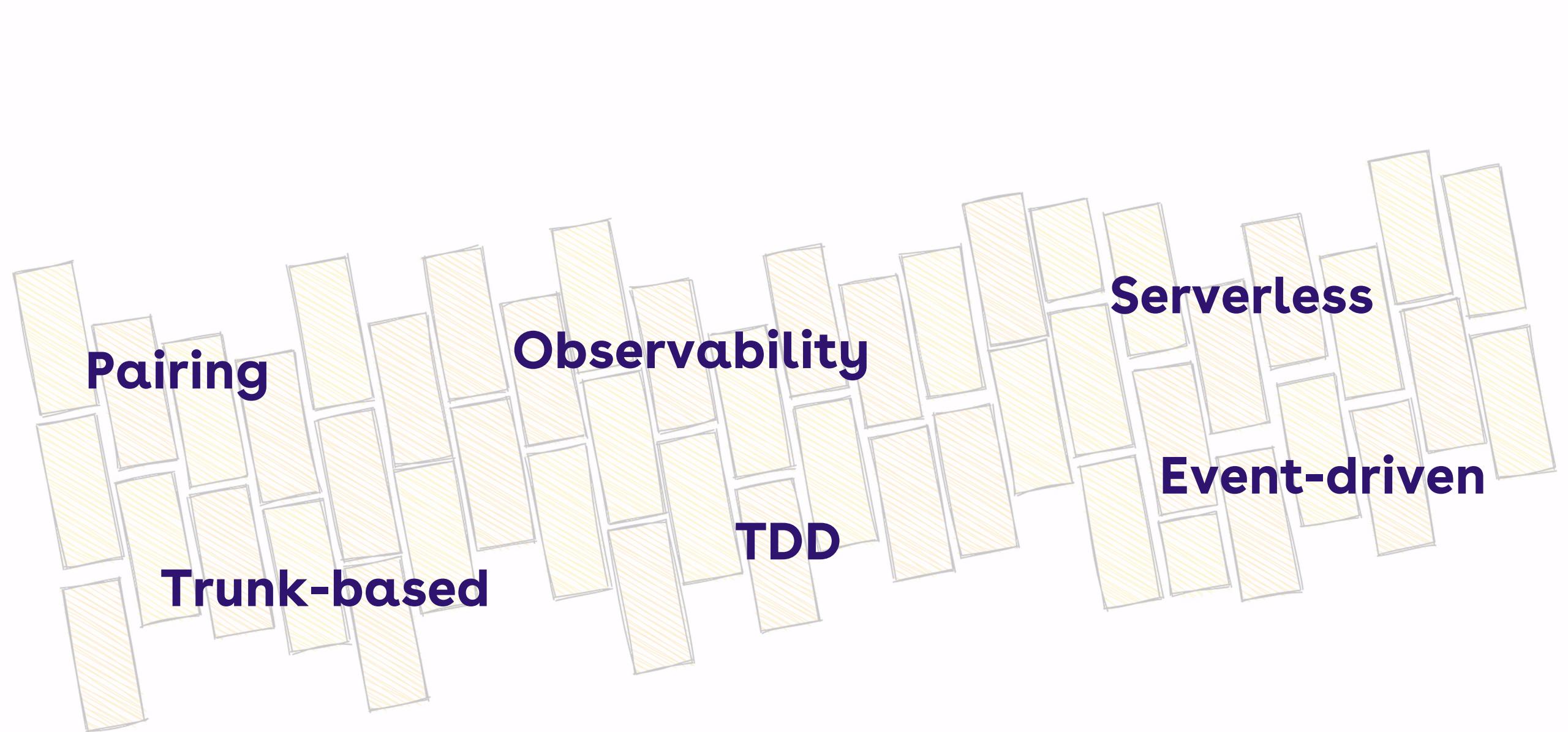
**Use events over
APIs**

TypeScript

React front-end

**Datadog for
olly**

**Either we decided the
technical constraints
now, or they'd appear as
we scale**



Pairing

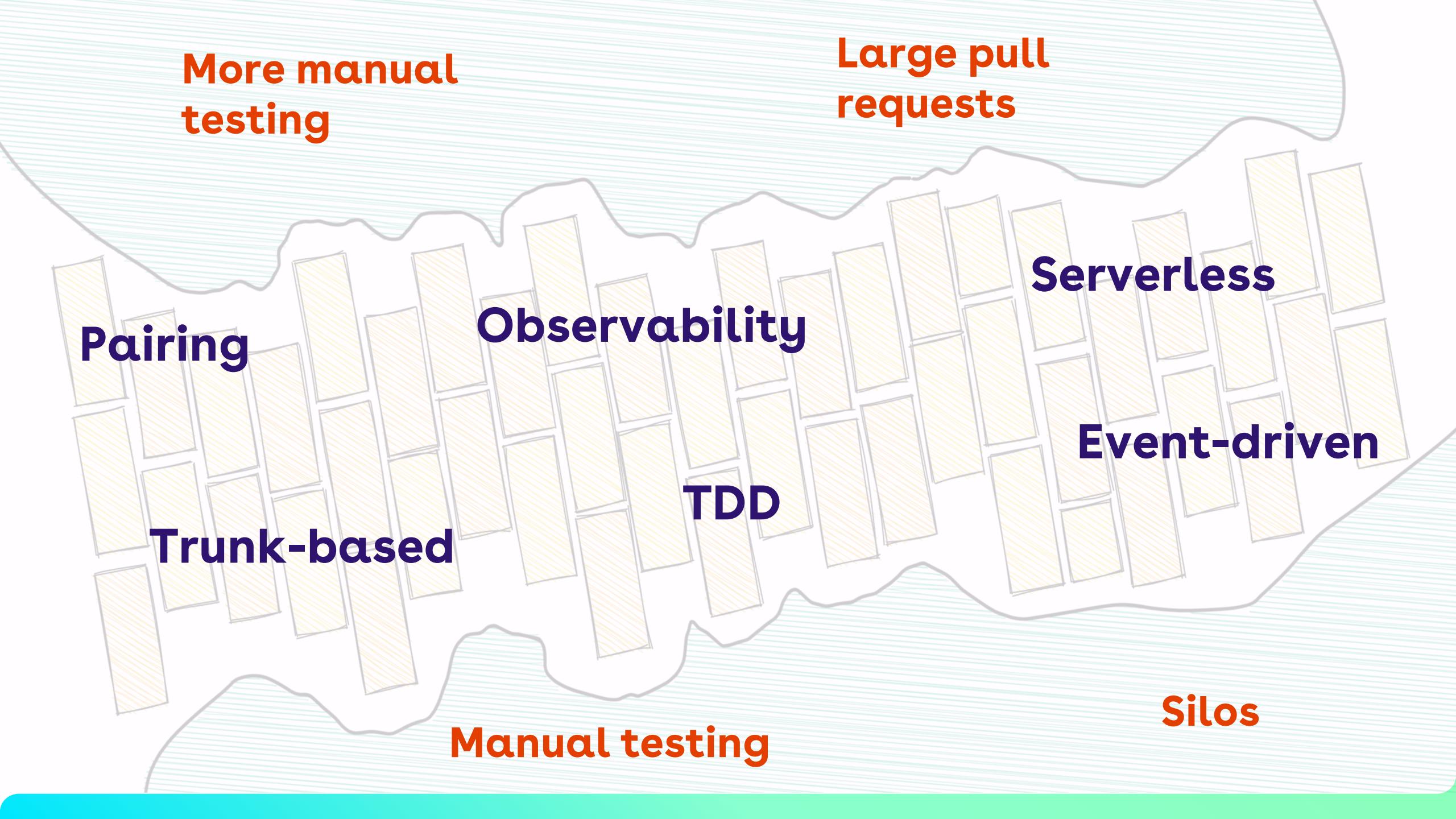
Trunk-based

Observability

TDD

Serverless

Event-driven



**More manual
testing**

**Large pull
requests**

Pairing

Observability

Trunk-based

TDD

Serverless

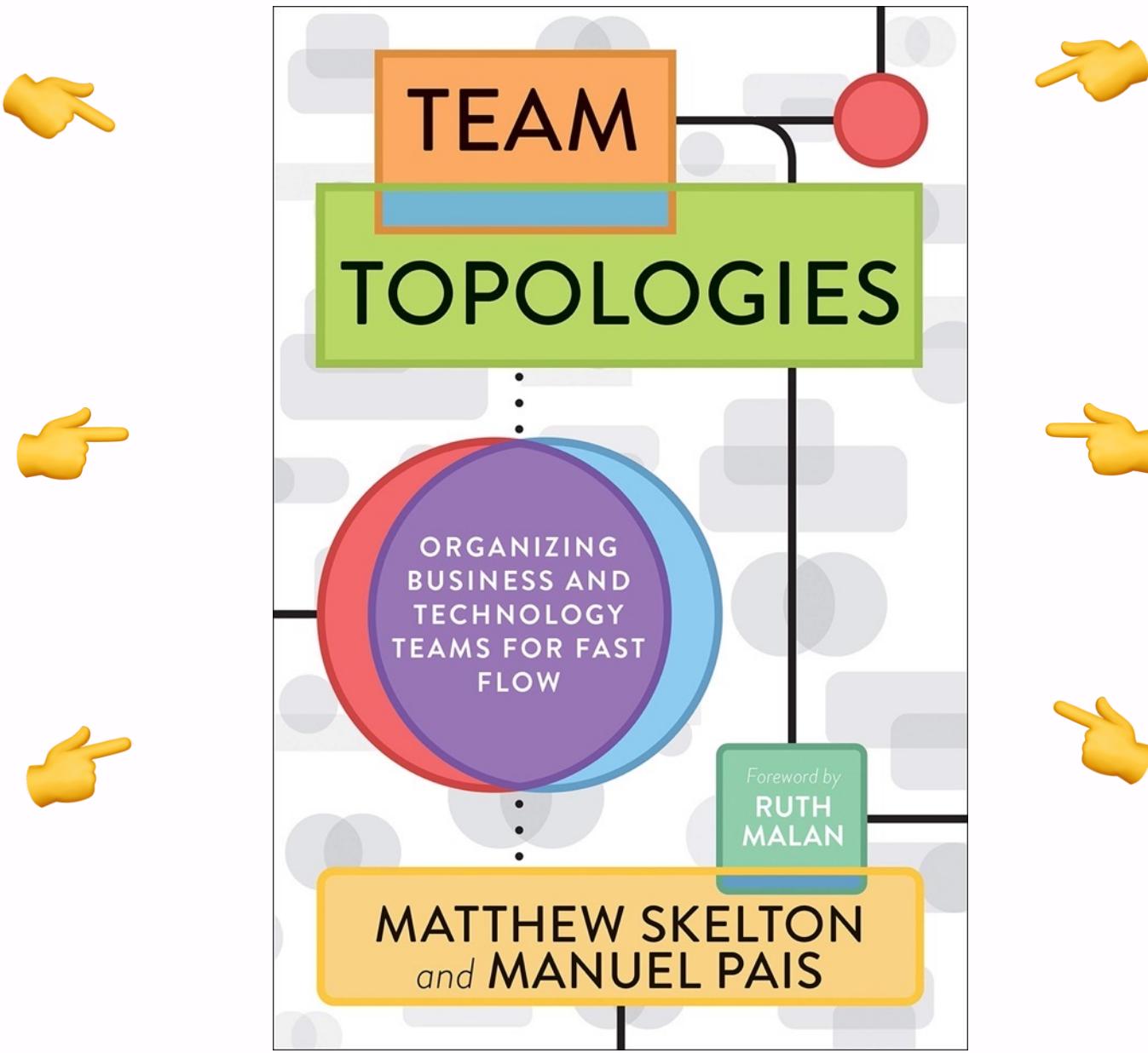
Event-driven

Silos

Manual testing







Disclaimer

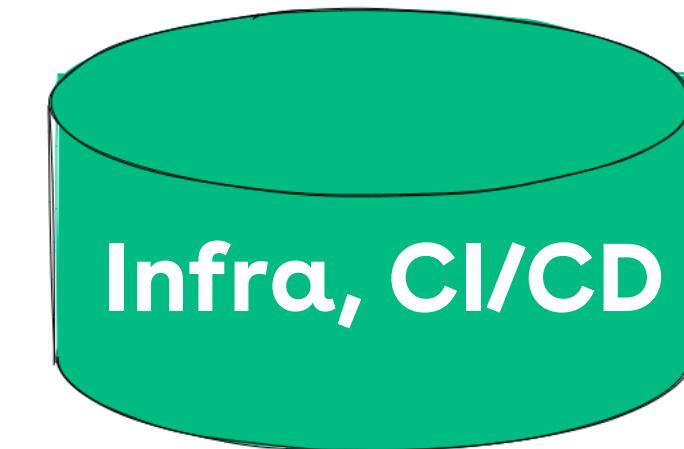
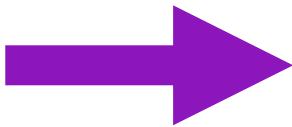
**We don't get any referral fees
from Team Topologies.**

Cognitive load

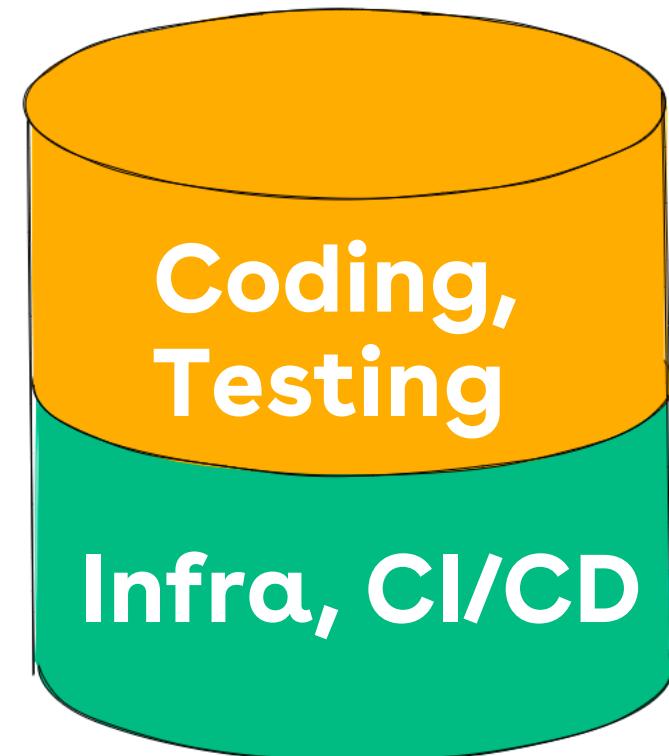
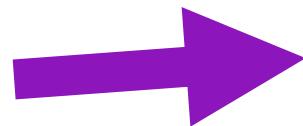
“The amount of mental effort being used in the working memory”

- John Sweller, Psychologist, 1988

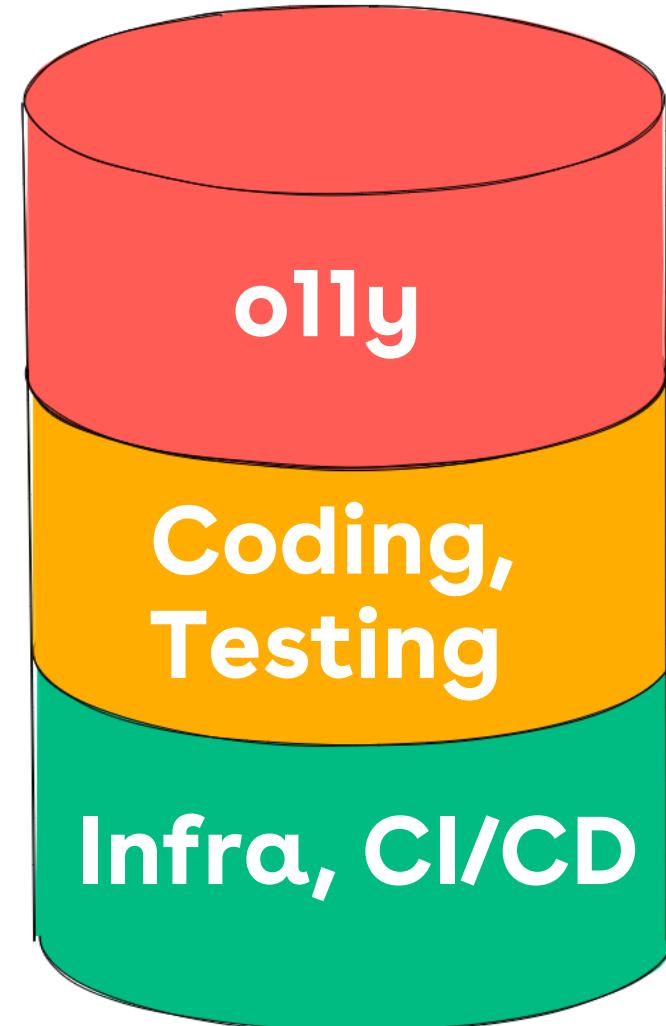
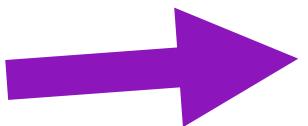
**Team
Cognitive
Load**



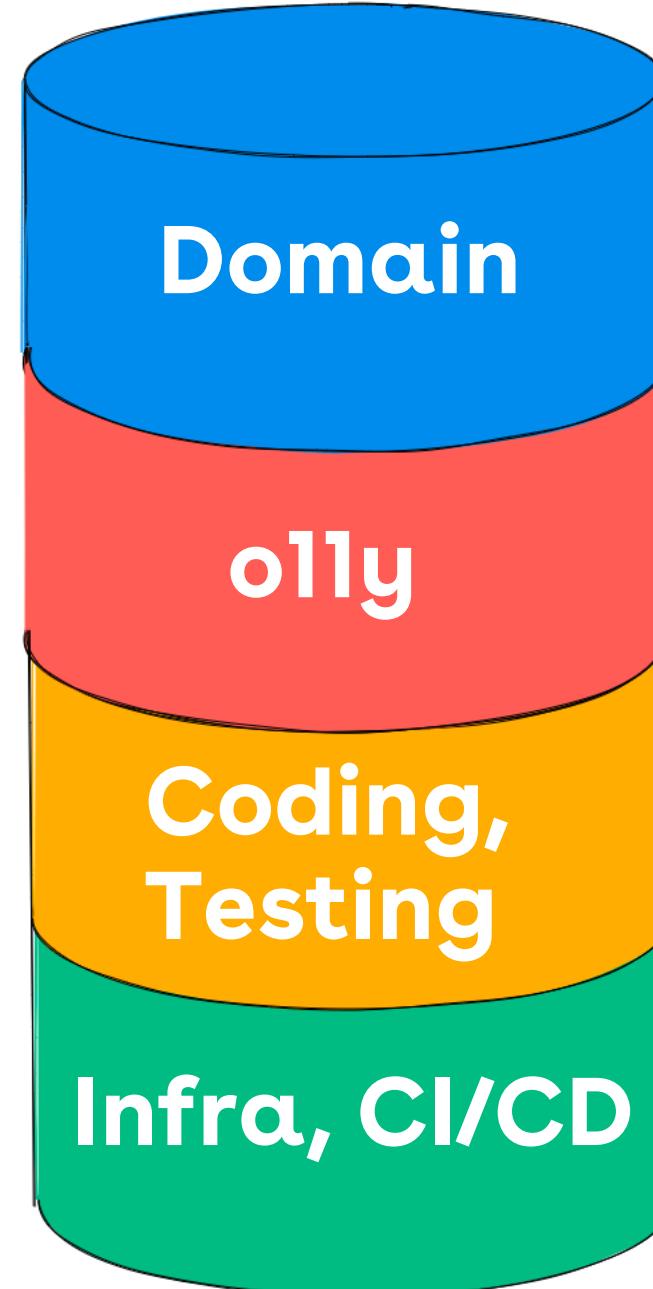
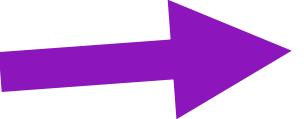
**Team
Cognitive
Load**



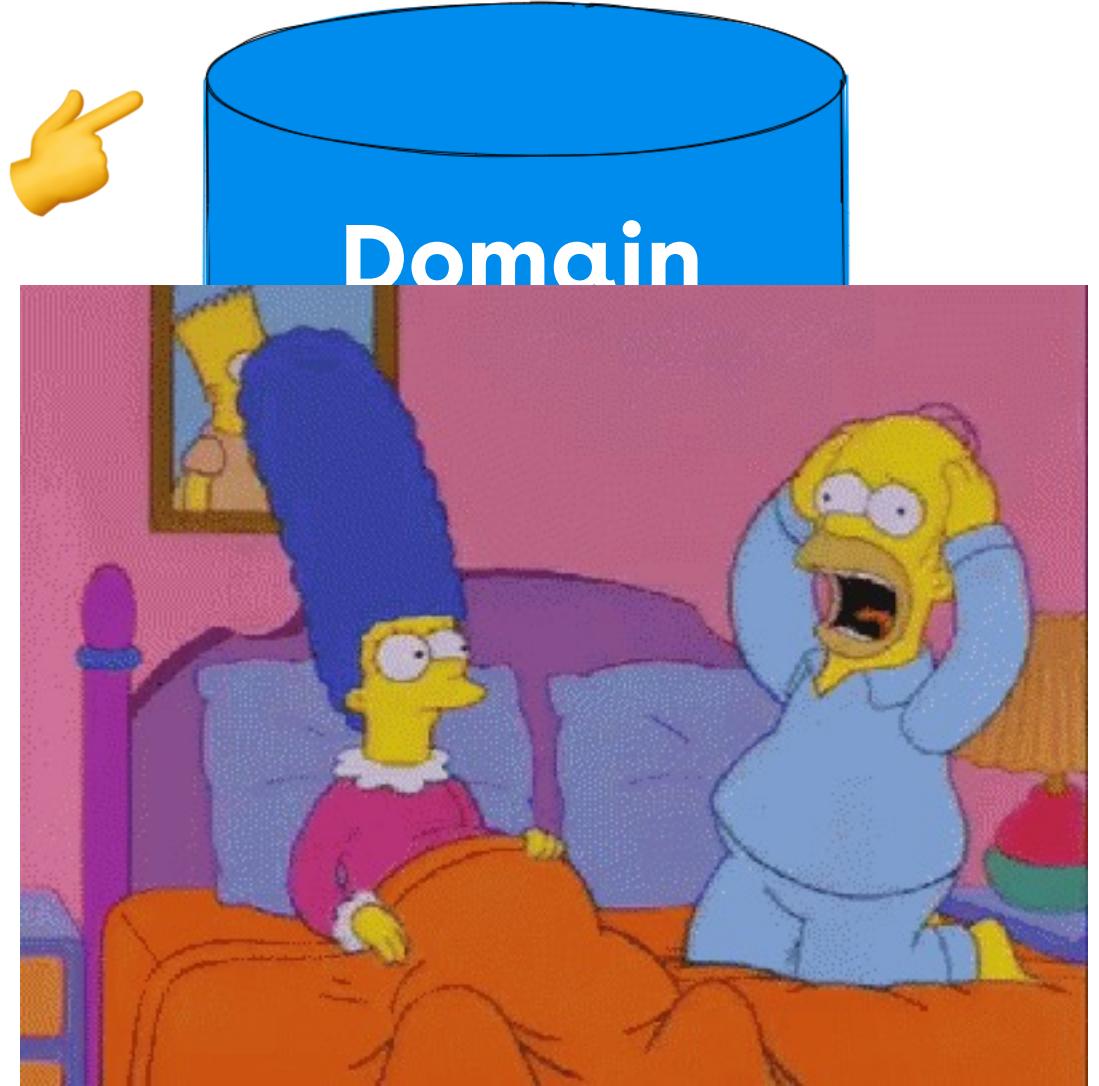
**Team
Cognitive
Load**

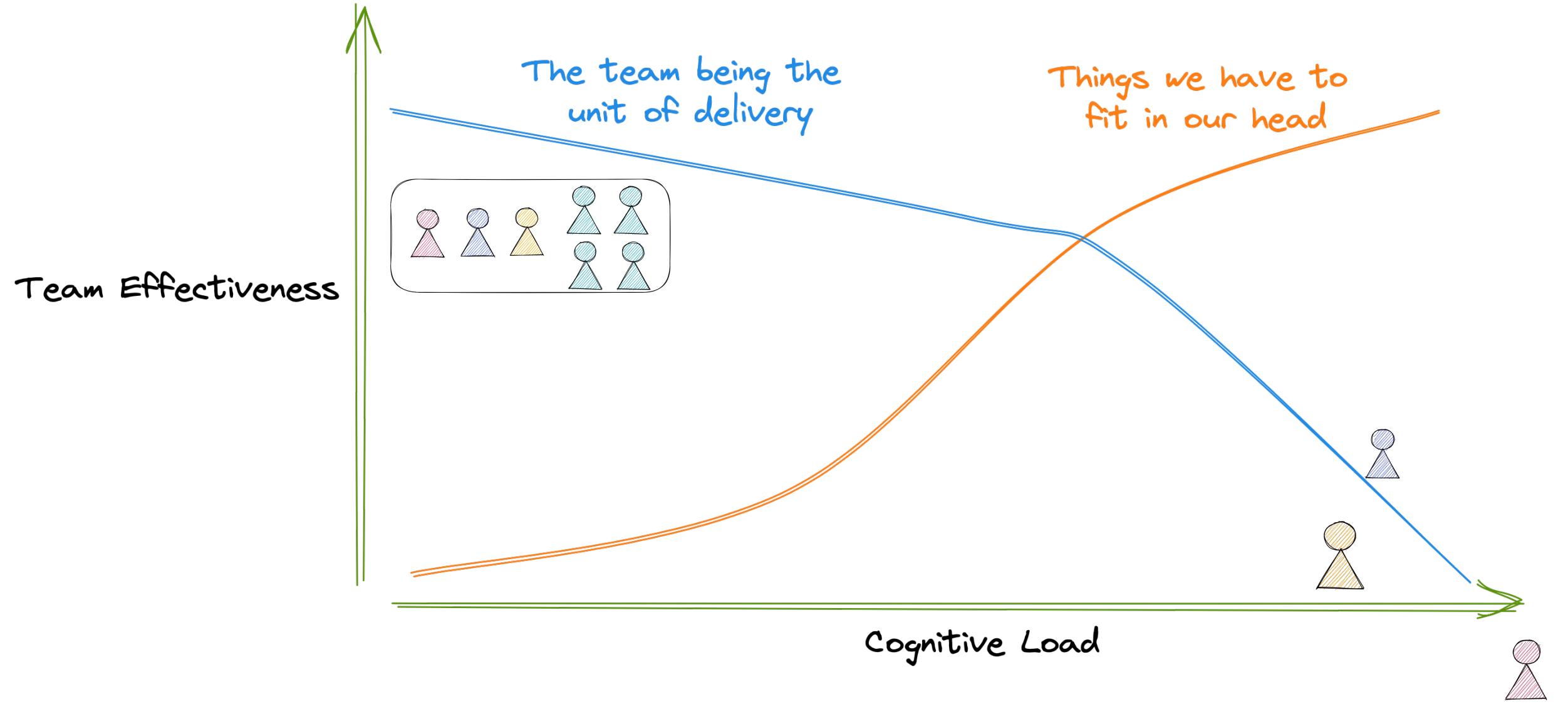


**Team
Cognitive
Load**



100% capacity

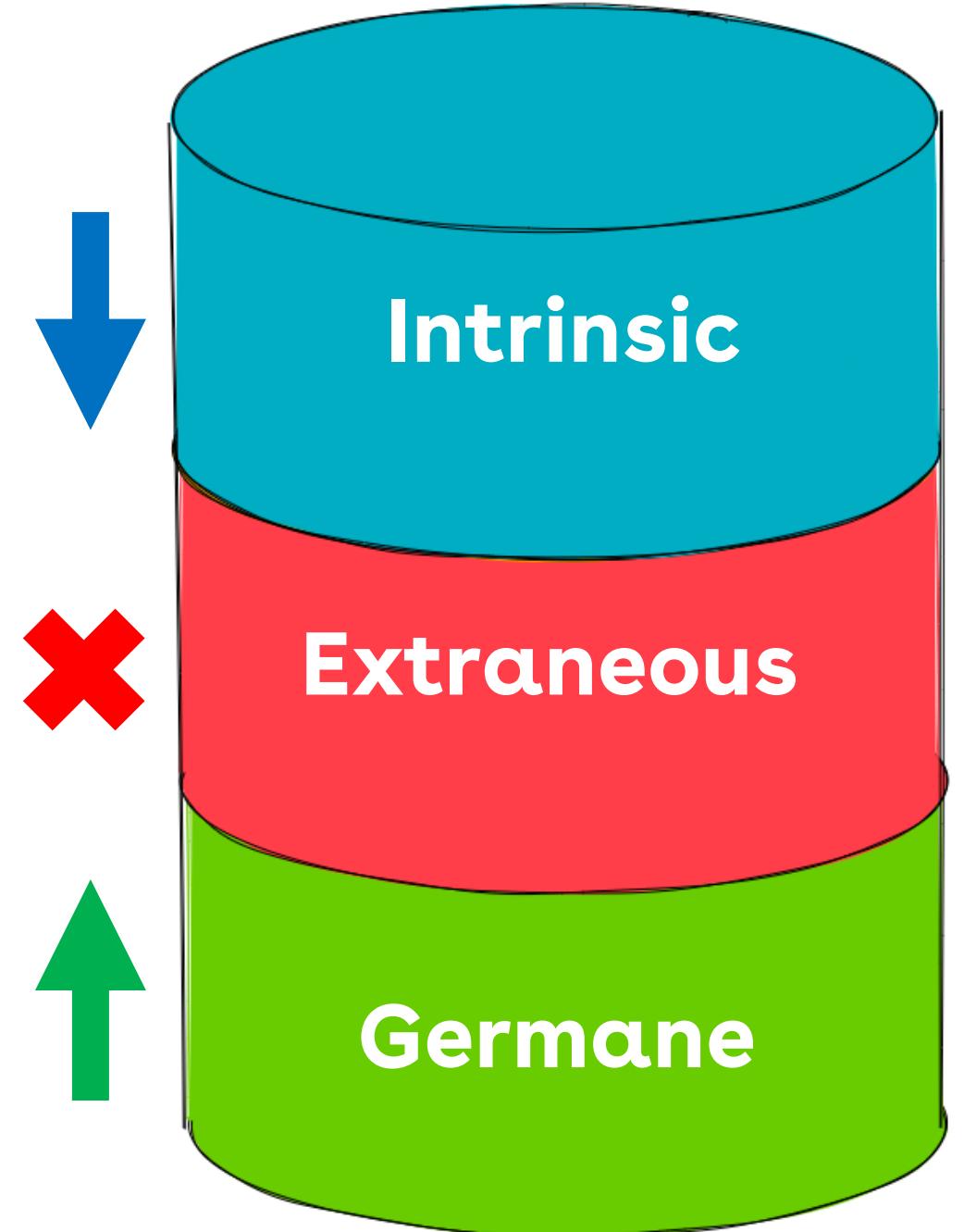




What are types in TypeScript?

“What’s that AWS
command again?”

How to make
payment of a car
easier?



**Focus on
selling cars**

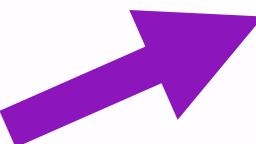


Germane

**Always
learning the
good bits**



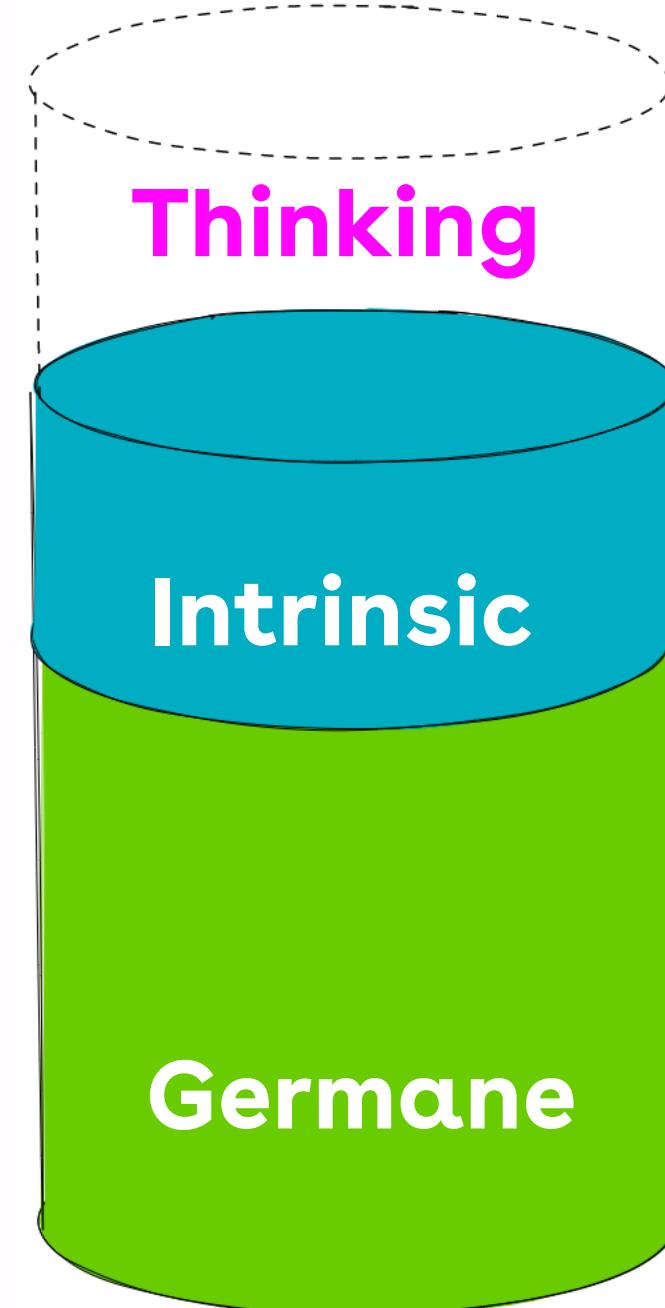
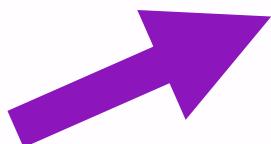
**Focus on
selling cars**



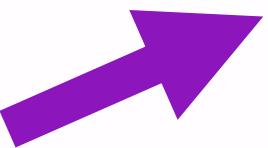
**Always
learning the
good bits**



**Focus on
selling cars**



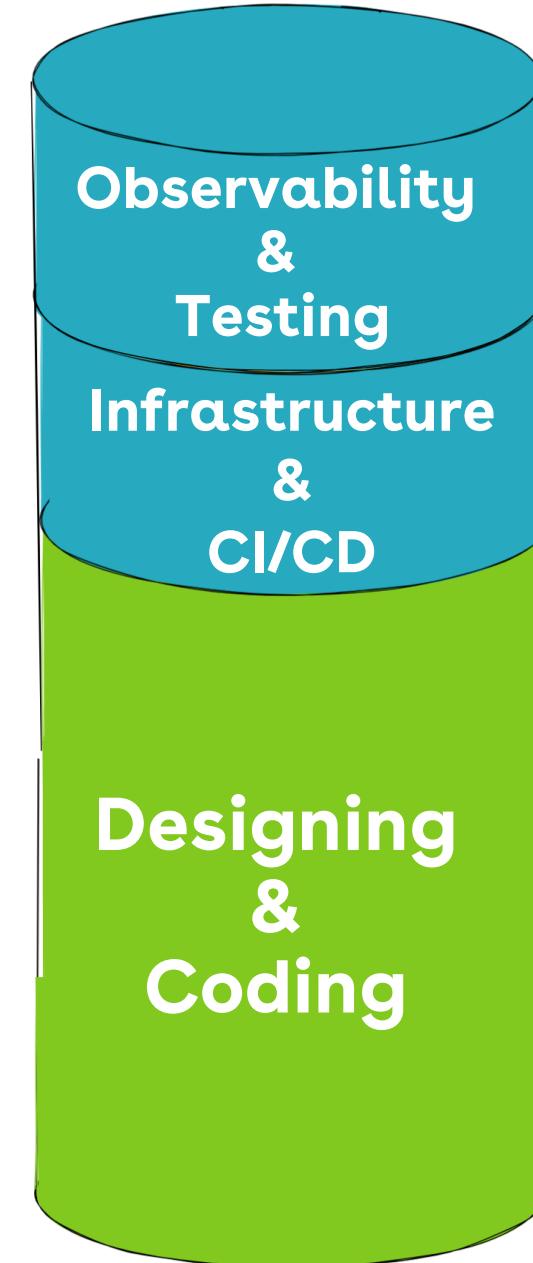
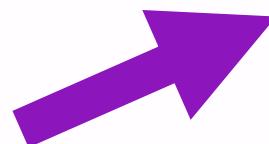
Germane



Designing
&
Coding

Intrinsic

Germane



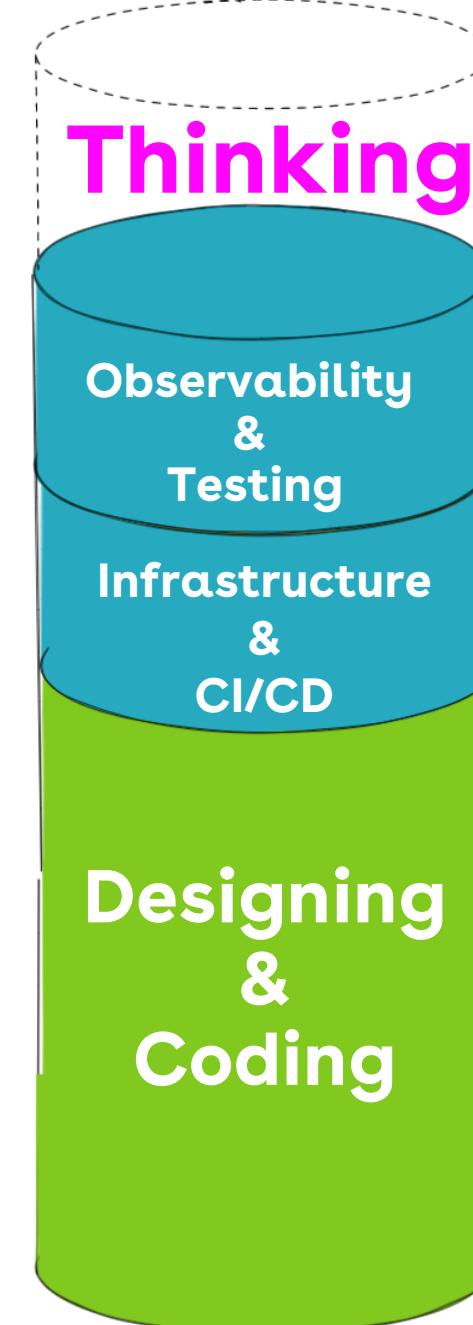
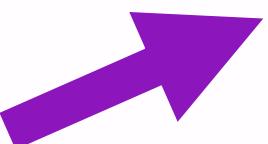
Space



Intrinsic



Germane



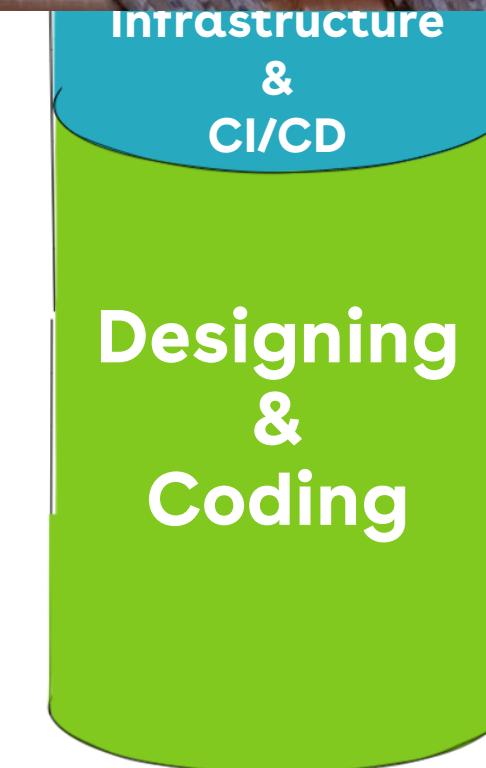
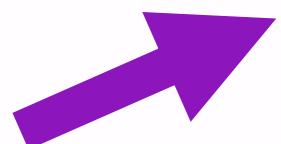
Space

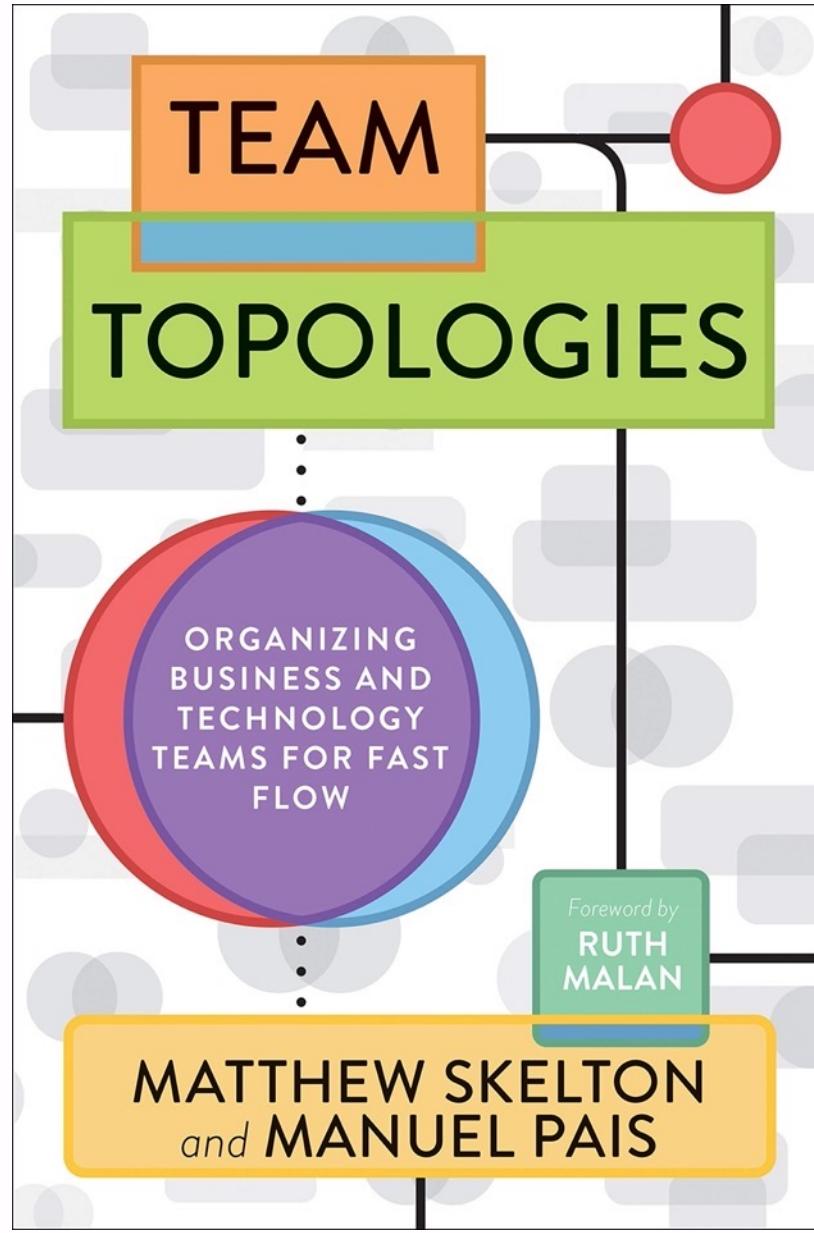


Intrinsic



Germane



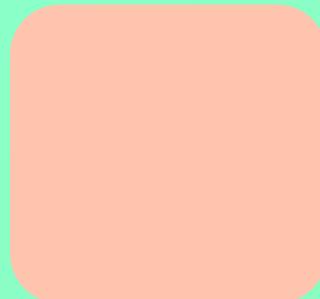
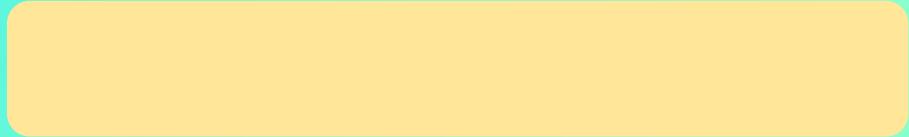


Stream-aligned teams

Enabling teams

Platform teams

Complicated sub-system teams



```
graph LR; A[Search] --> B[Vehicle Detail]; B --> C[Checkout]; C --> D[Infrastructure]; D --- E[Principal Engineers]
```

Principal
Engineers

Search

Vehicle Detail

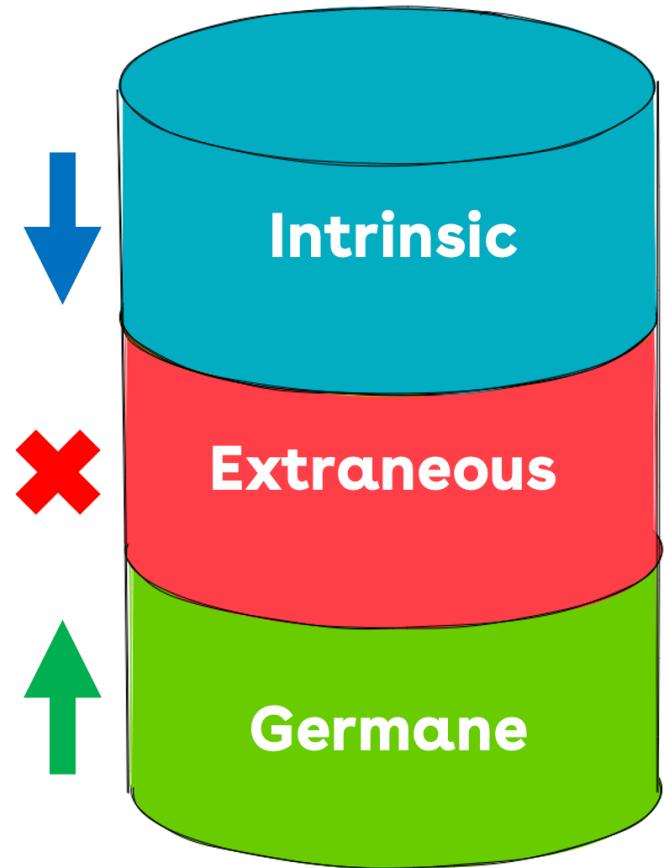
Checkout

Infrastructure

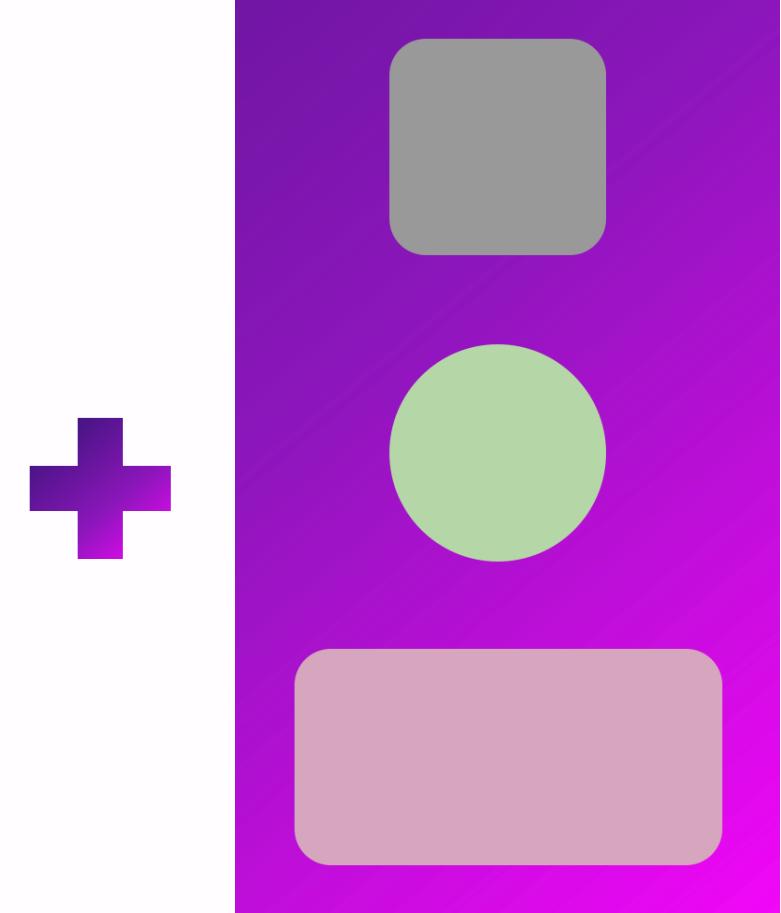
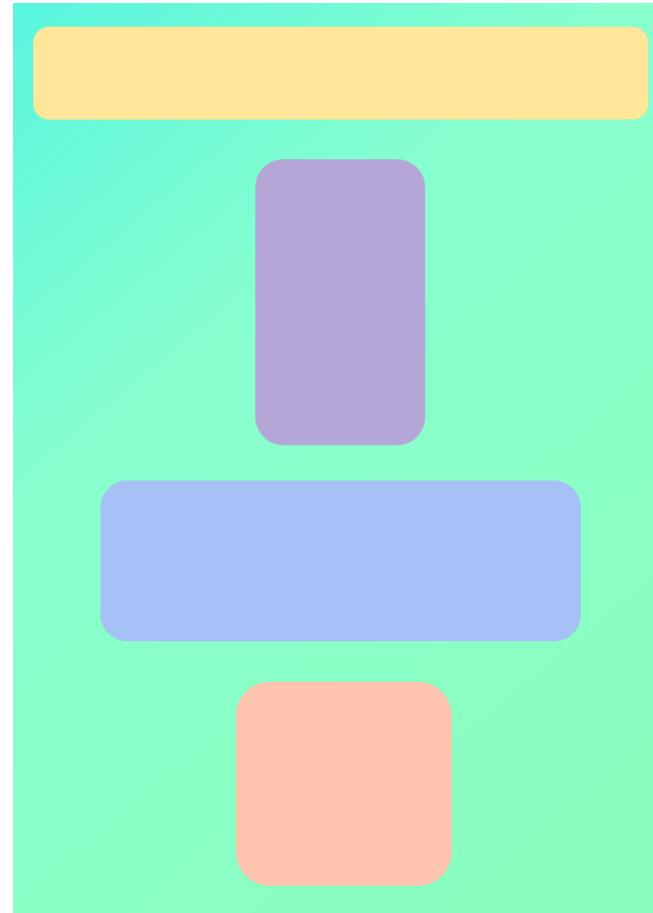
X-as-a-service

Facilitating

Collaboration



Cognitive Load



Search

Vehicle Detail

Checkout

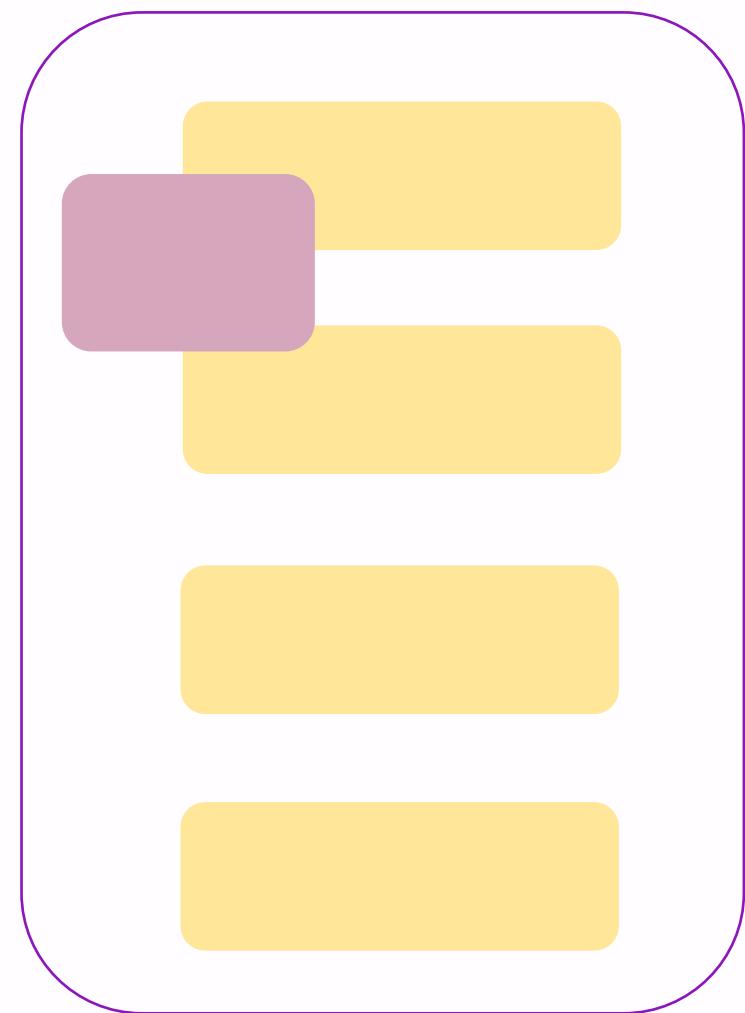
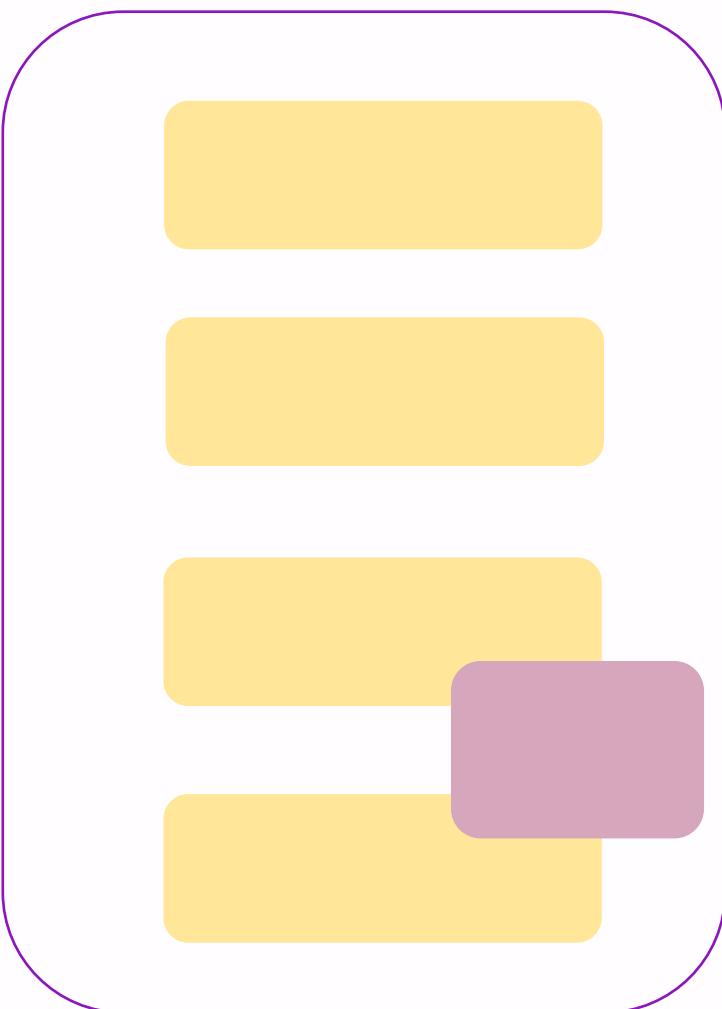
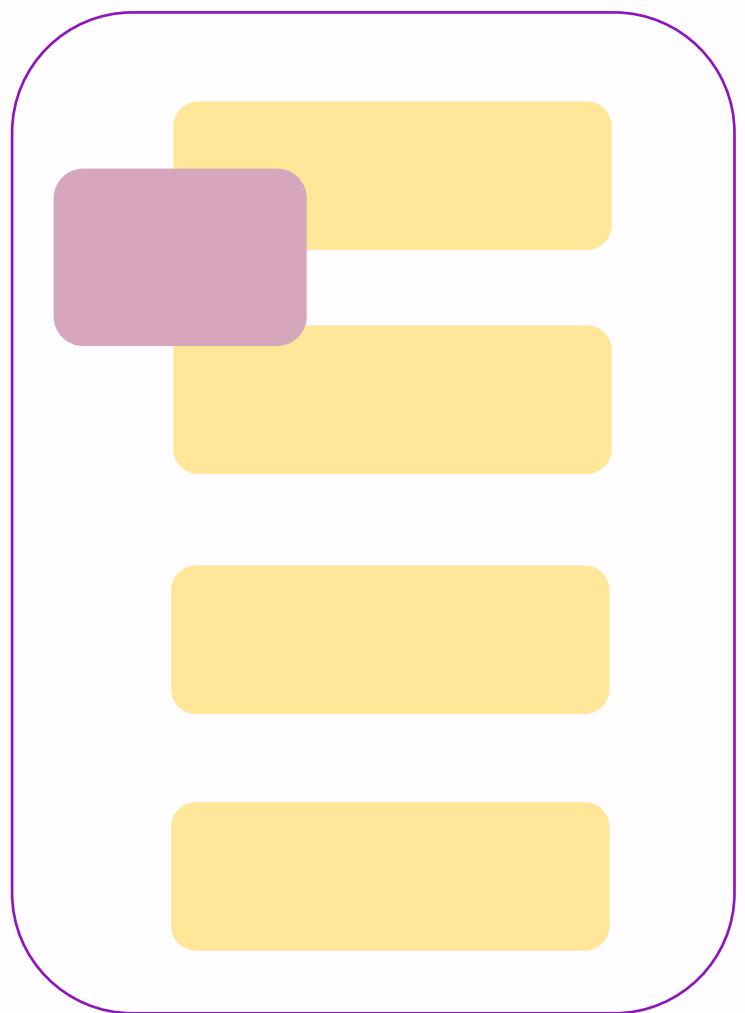
Infrastructure

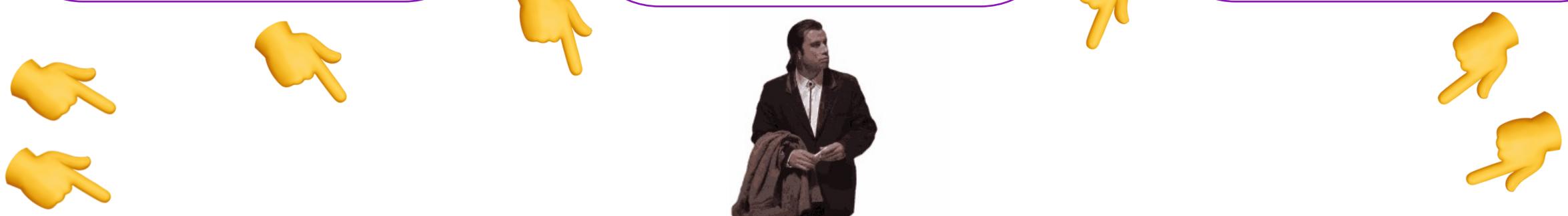
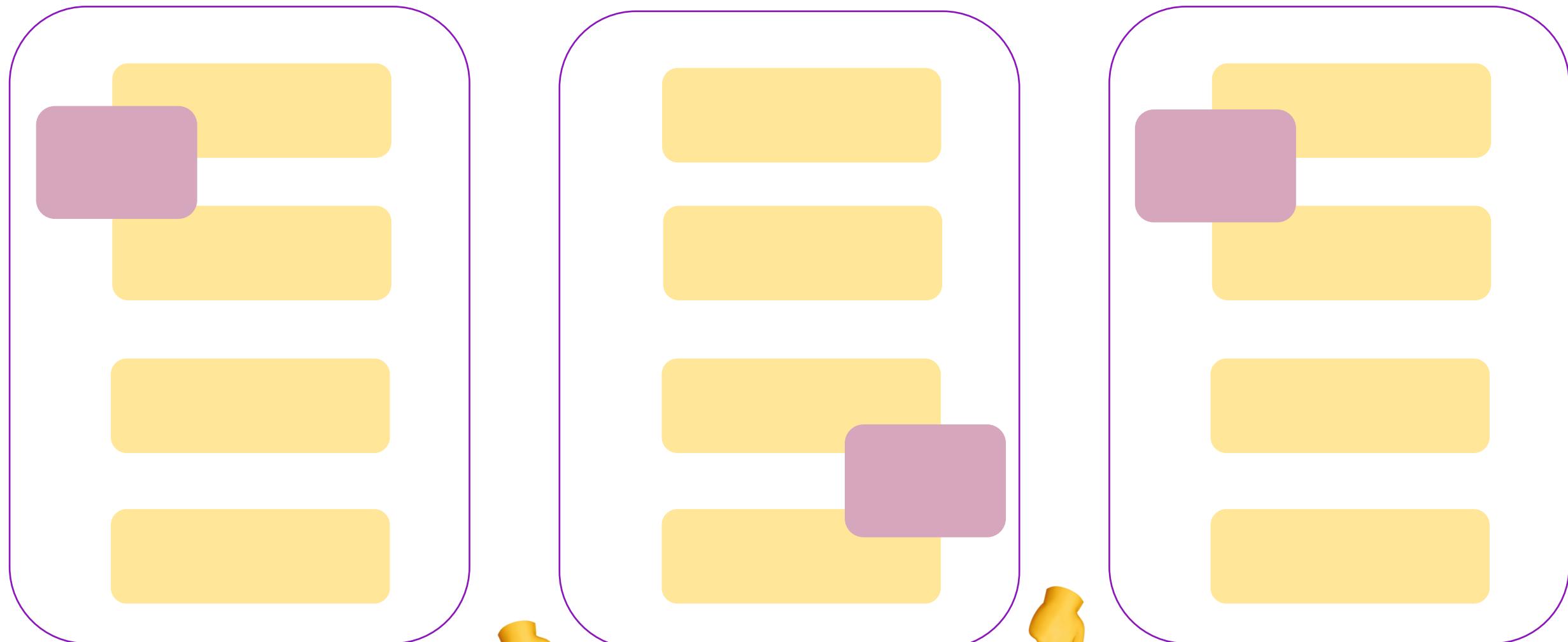
Collaborating

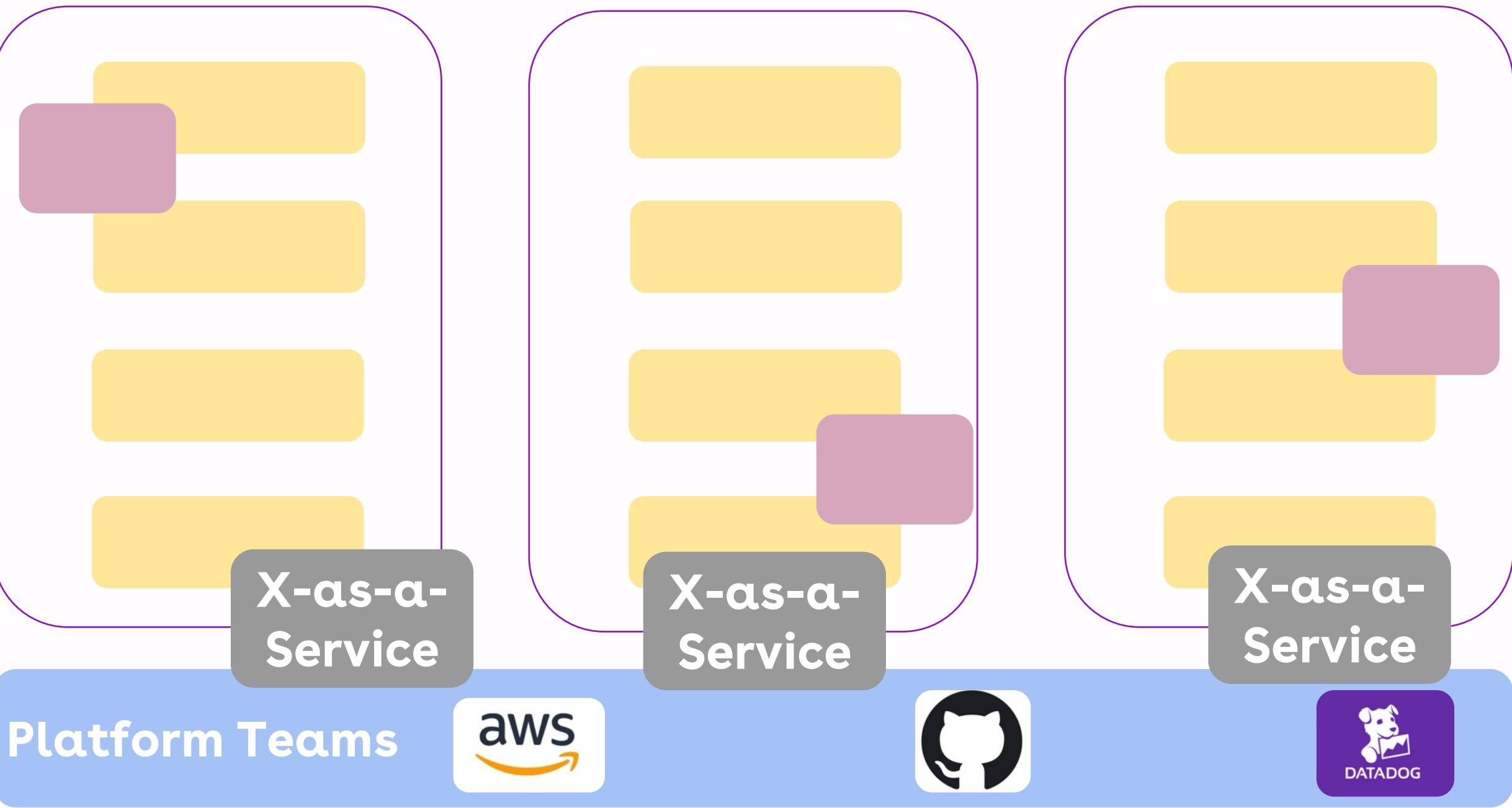
**X-as-a-
Service**

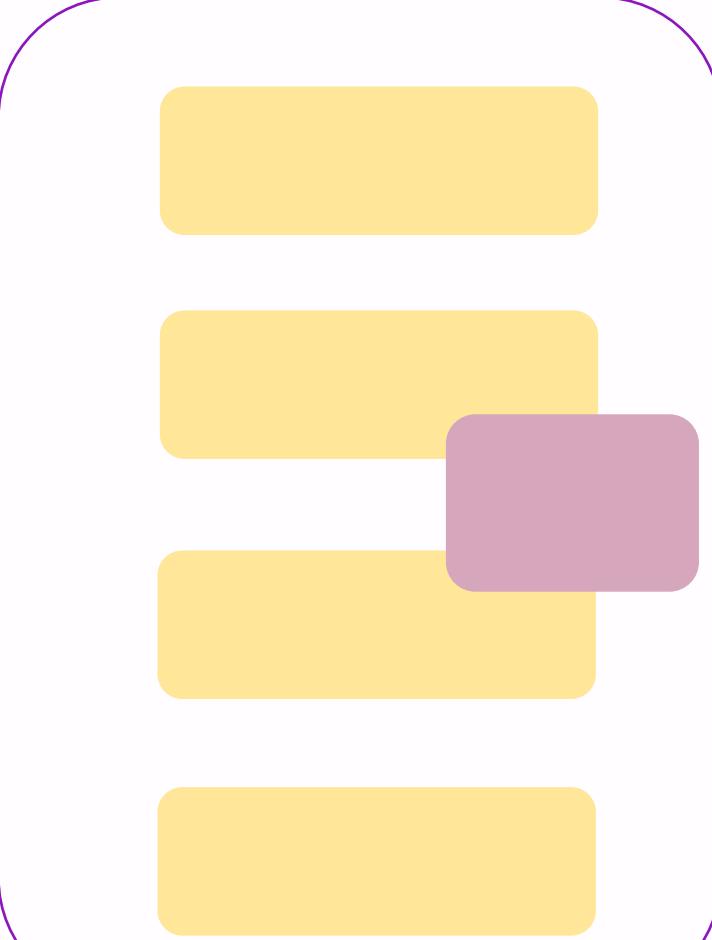
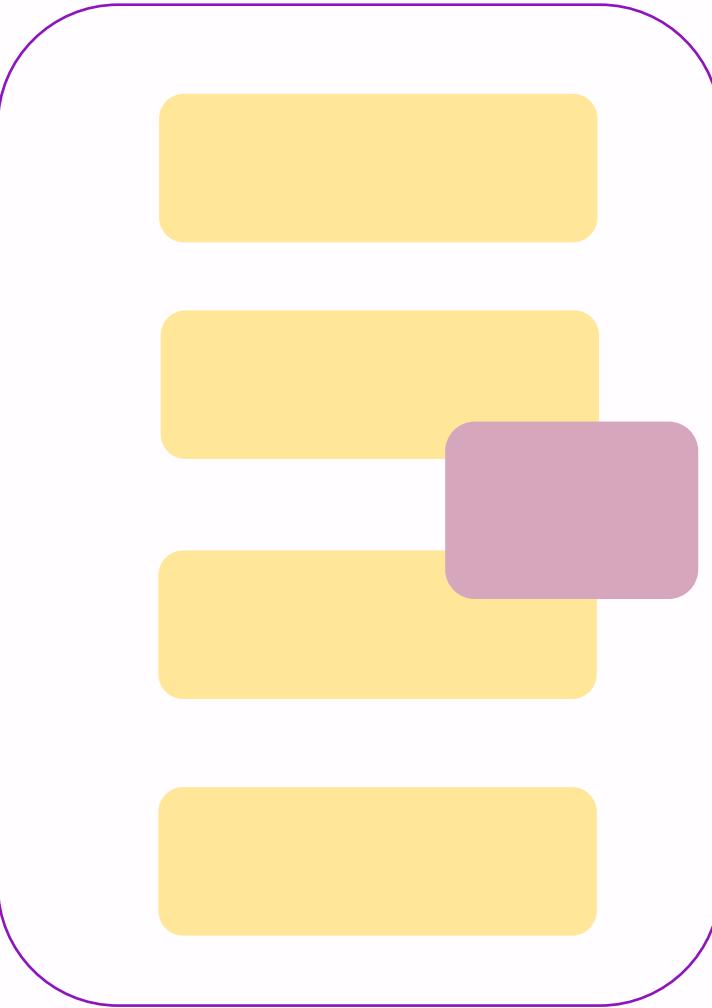
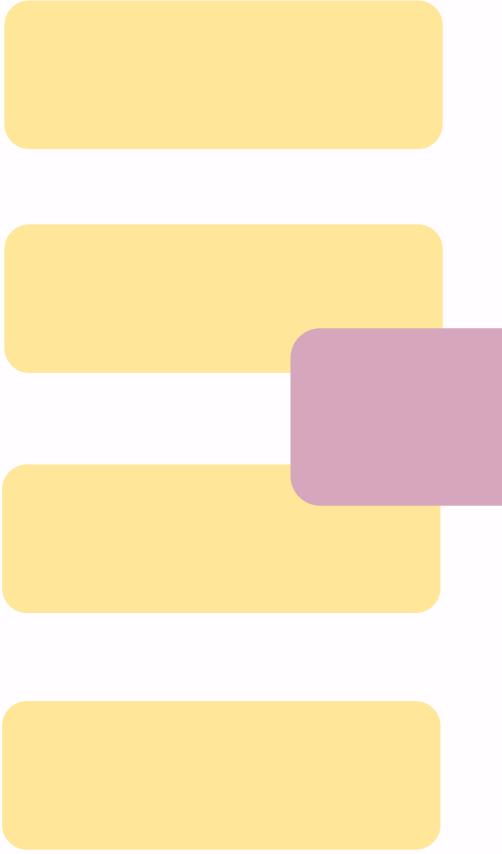
**Principal
Engineers**

**Making
it easy**









Working Groups

Make it
easy

Working Groups

Make it
easy

Working Group API: Back-end component template



+2

Created by Dan Haughey (Unlicensed)

Last updated: Jan 05, 2022 by Apostolis Apostolidis • 3 min read • 57 people viewed

[i Overview](#) | [🤔 What is the problem?](#) | [⚽ Goals](#) | [📝 Kill Criteria](#) | [✂️ What is out of scope?](#) | [💬 How are we collaborating?](#) | [📣 Team news](#) | [📝 Relevant Documents](#) | [✖️ Limitations & Concerns](#) | [🔮 Future Work](#) | [🕒 Recently updated](#)

?? Name	Back-end component template
⏳ Status	COMPLETE
👤 Owner	@Dan Haughey (Unlicensed)
💰 Sponsor(s)	@Jonnie Barnes
👥 Team	@Apostolis Apostolidis @Paul Richards @Dan Haughey (Unlicensed) @Greg Farrow @Ben Wainwright @Nick.Keers (Unlicensed) @dharmesh.lad (Unlicensed) @john.bradshaw (Unlicensed) @Simon Scarfe
📍 Miro Board	Working Group: Code Template
🏷️ Labels	backend-component-template
⌚ Period Active	14 Jul 2021 - 30 Nov 2021
🏆 Outcome	Single backend component template was created using CDK. Template ownership is handed over to the backend guild

WORKING GROUP - OILY BLUEPRINT

0. 📋 Agenda

Agenda Working Gr...



1. 🖼 Set the scene

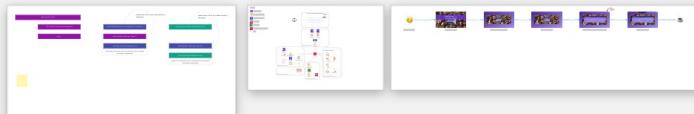
Meeting Det... Define Scope Ways of Wo...



[Work in Progress \(GitHub Project board\)](#)

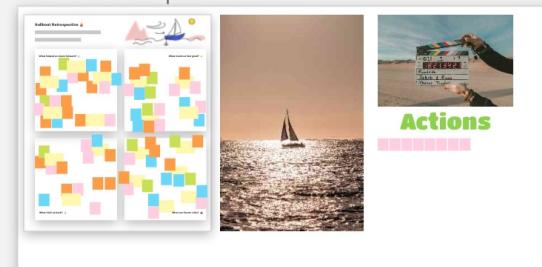
2. 🔨 Architect

Linking Spans Architecture... Customer Journey



3. 🧽 Wash up / Retro

Washup and Retro...
Sandboxed Retrospe...



Working Group API: Service status

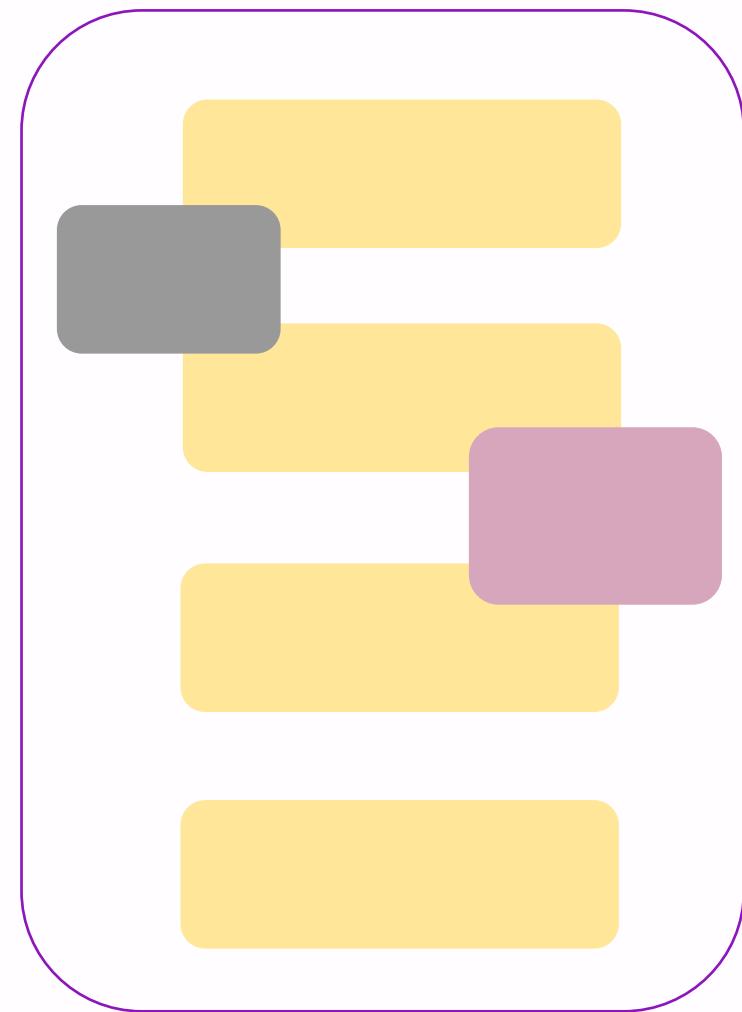
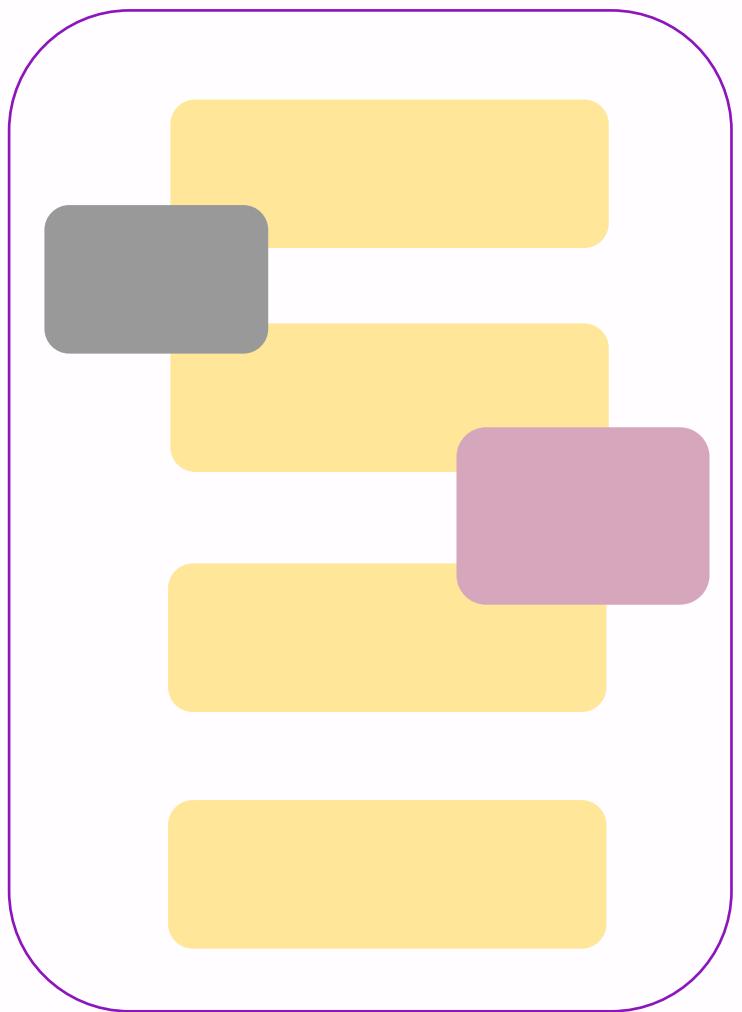
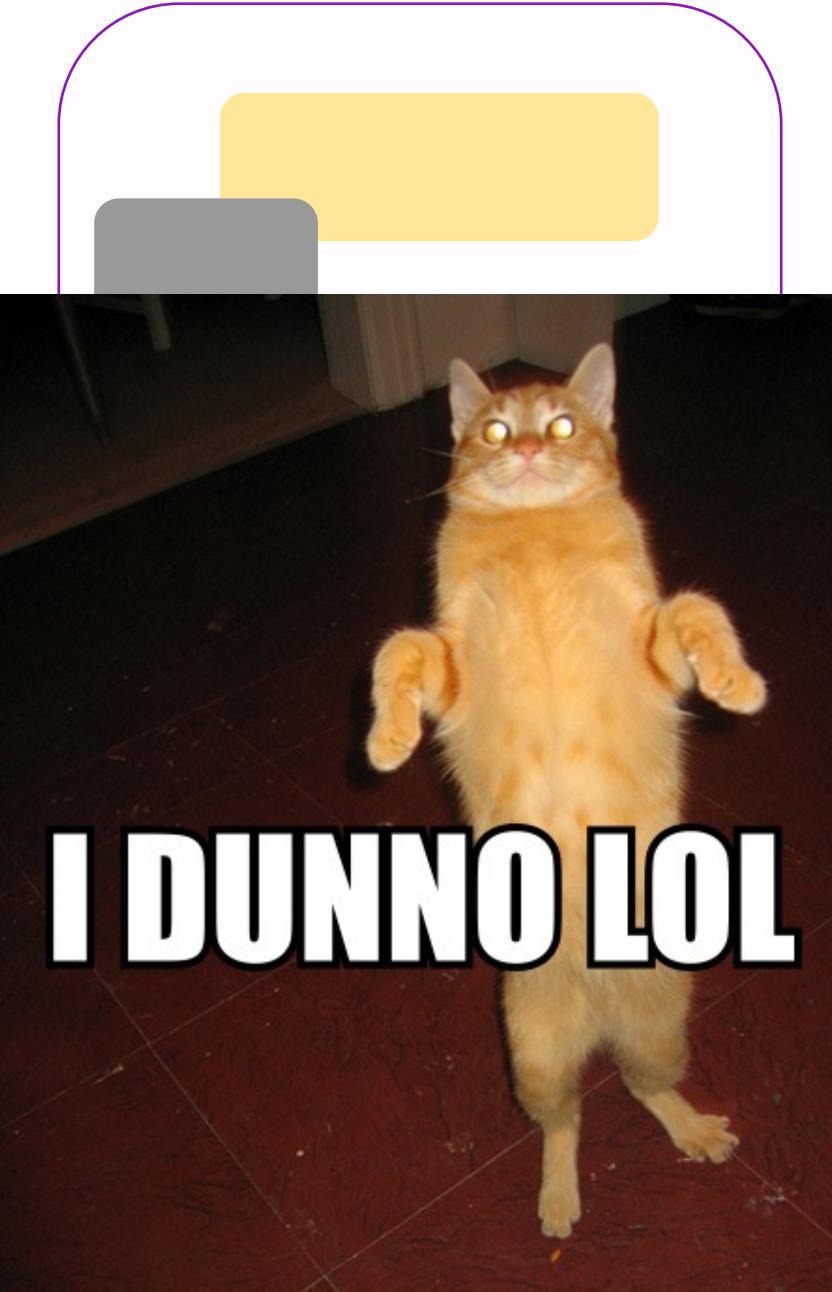
NOT
STARTED

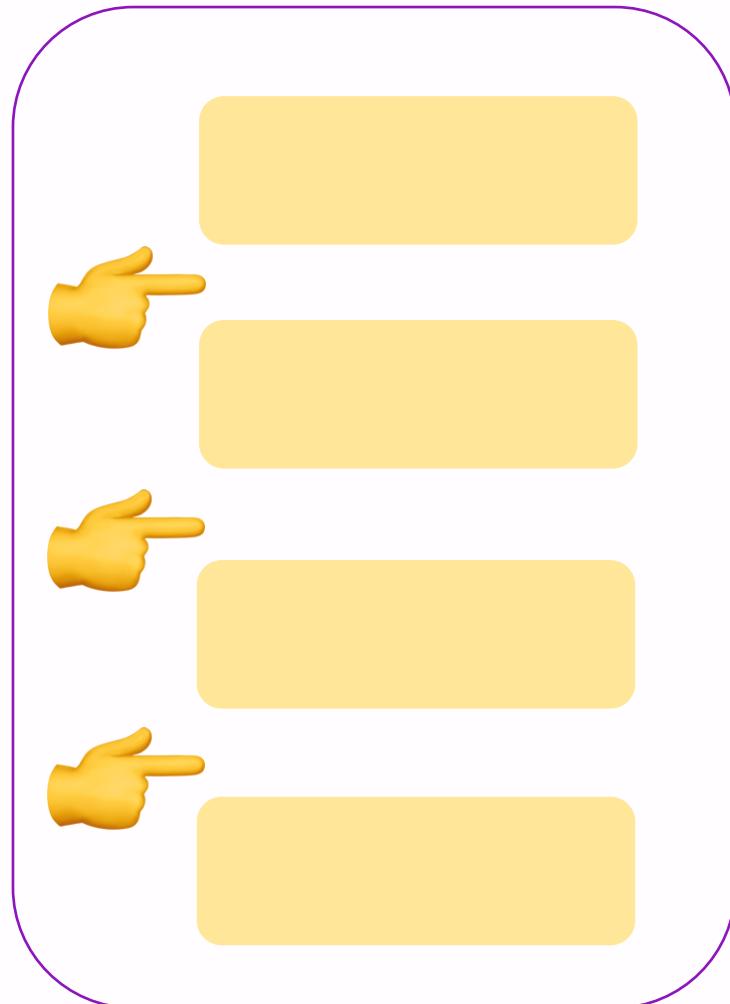
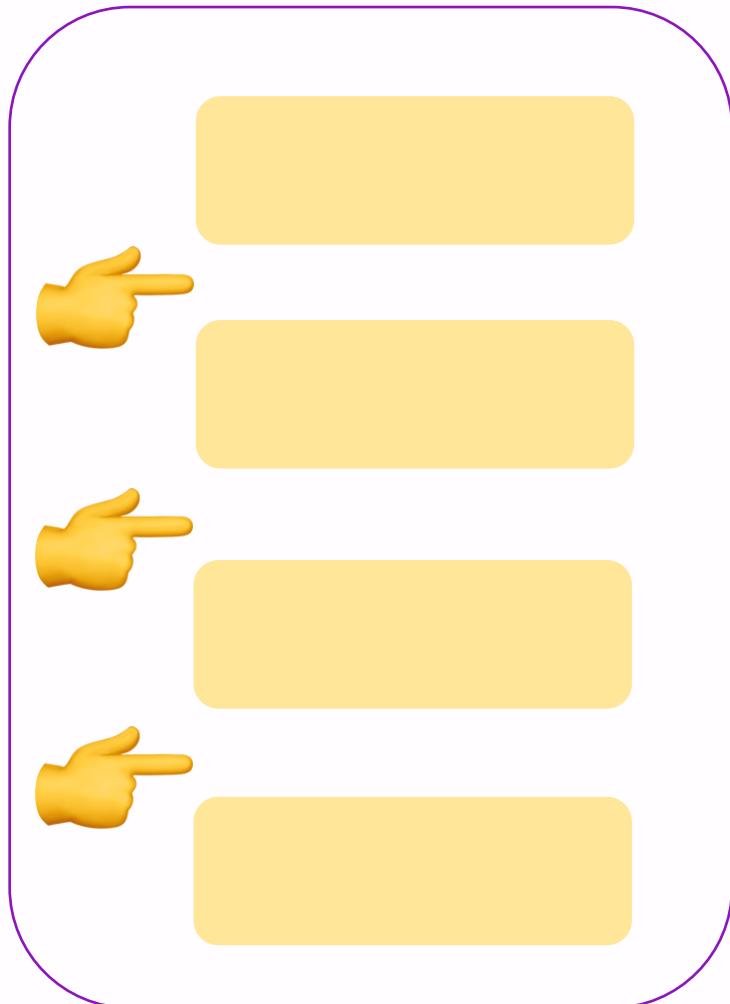
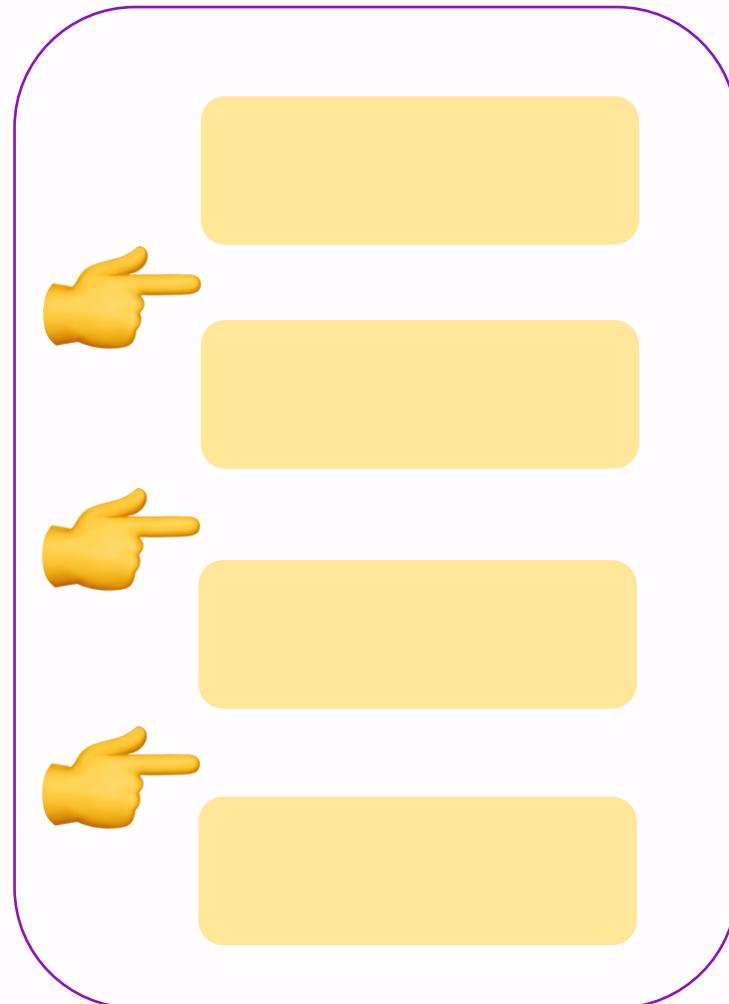
Working Group
API: Monorepo
- Vision and
Guardrails

IN
PROGRESS

Working Group
API: internal
developer
documentation

COMPLETE





Engineering
Managers

Making
it easy

Stream-aligned team



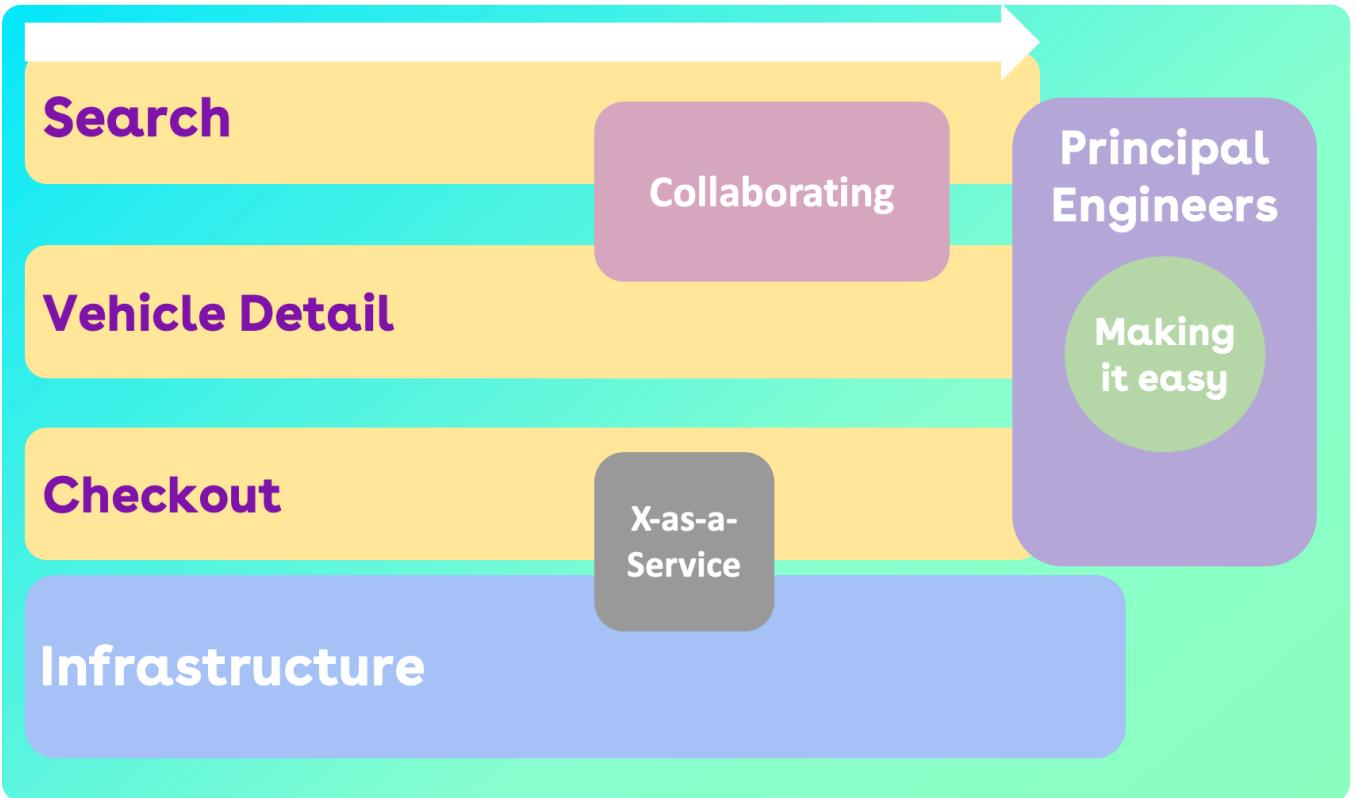
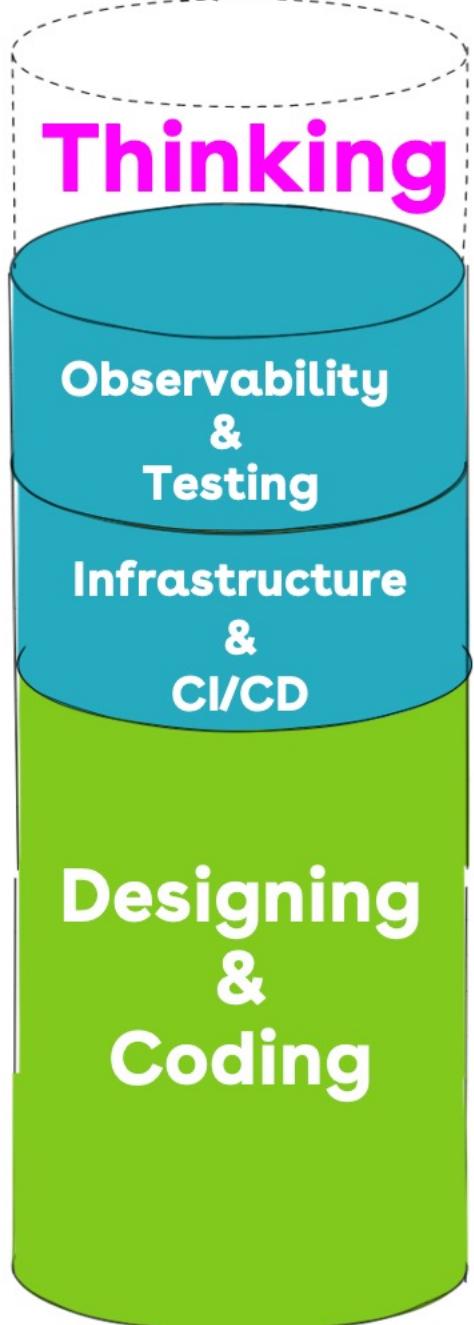
Principal
Engineers

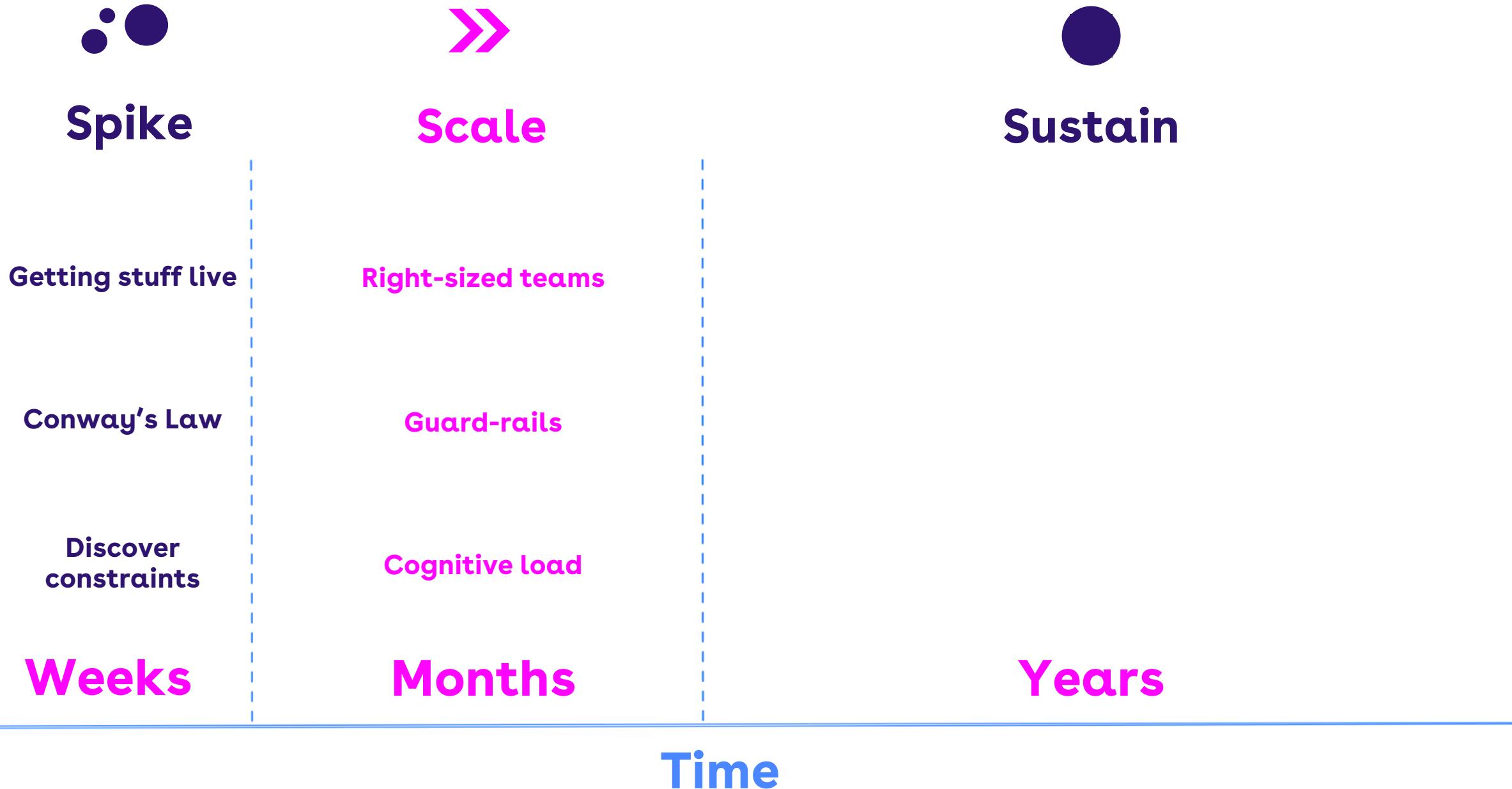
Making
it easy

Stream-aligned team

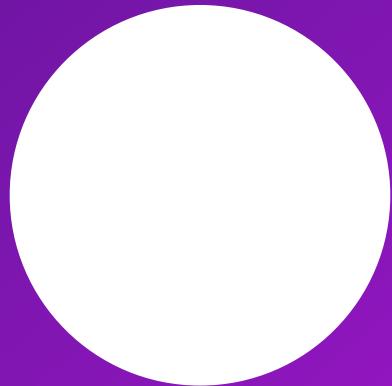
Stream-aligned team

Stream-aligned team





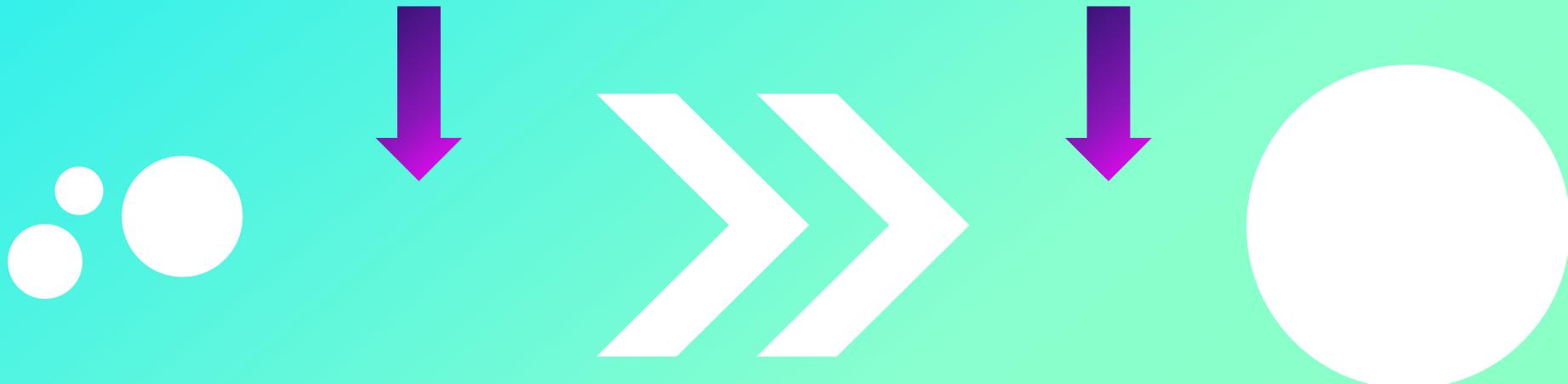
Sustain.



To sustain means to
support.

**What are you
optimising for?**

Focus on the gaps





Spike

Getting stuff live

Conway's Law

Discover constraints

Scale

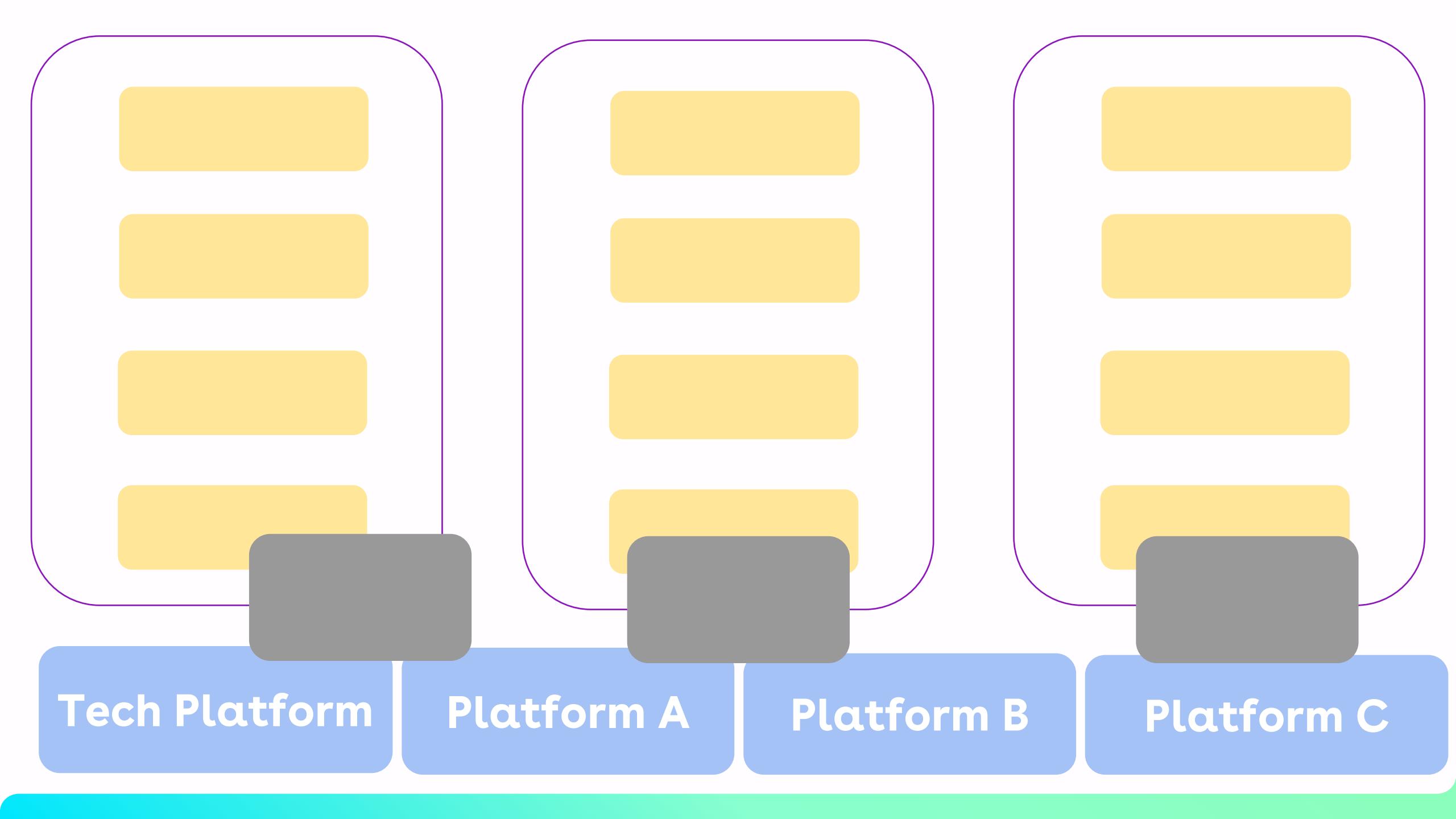
Right-sized teams

Guard-rails

Cognitive load

Sustain

Better ways of working



Maturing our platforms

Infrastructure

Design System

Orders

Vehicle Data

Becoming knowledge stewards

Practice → **How we get better at what we do**

Experience → **Learning from those who've been there
and managing tacit knowledge**

People → **Cross-organisation collaboration**

Domain → **Understanding the bigger picture**

Technology Blueprint

The image displays two side-by-side screenshots of the Cinch Technology Blueprint website, showing different sections of the platform.

Left Screenshot (Observability principles page):

- Header:** cinch Technology Blueprint, Guidance, Patterns, Examples, Technology radar.
- Sidebar:** Guidance, Principles, Technology, Organisational, **Observability** (selected), CI/CD, Architecture, Decision records, Cloud, Event Driven Architecture, Patterns, Examples, Technology radar.
- Section:**

Observability principles

The core goal of cinch is to operate a car buying platform that takes the faff out of finding, buying, and owning a used car. Below are some guiding principles and practices for good observability.

Observability at cinch

Change in software is a constant. A software system that doesn't change will inevitably suffer. At cinch, we embrace change but we also strive to build software with observability at its core.

Why do we need observability?

At cinch, we build event-driven distributed software systems. This does provide us with resilient software. It also introduces complexity. Understanding the what, when and why something happens in the system is hard. We provide value to customers as they use our software systems. So we would like to answer the question:

 - How are our software doing when exercised by real users?

We can extrapolate a mental model of what *may* happen by reading the codebase. This is rarely efficient or even enough. This is where observable software systems come in handy. With observability, we can understand how our software is working. And we do this by evaluating its outputs.

What do observable systems provide?

Insights into the workings of our software are in the form of telemetry data. We can build a mental model that maps what is happening within the software. With this we can understand both the *what* is going on (and *when*) but crucially, the *why*. With observable systems we are able to answer the following questions by querying the telemetry outputs of our systems:

 - What is going wrong?
 - Why is it going wrong?
 - How well is the system performing against our business objectives?

How do we achieve observability at cinch?

At cinch, we use [Datadog](#) as our observability platform. The vehicle to more observable systems is to have good telemetry data. Coupled with good squad engineering practices and ceremonies.

For more practical observability guidance please refer to [The Observability Journey](#) on Confluence

Right Screenshot (Consumer Contract Tests for Http Requests page):

- Header:** cinch Technology Blueprint, Guidance, Patterns, Examples, Technology radar.
- Sidebar:** Guidance, Principles, Architecture, Cloud, Event Driven Architecture, Patterns, **Examples** (selected), **Blueprints**, Customer API authentication, Optimise for awareness, Evolve meaningful monitors and dashboards, Glossary, Contract testing, Consumer HTTP test (selected), Provider HTTP test, Consumer event test, Provider event test, Templates, Backend component, Technology radar.
- Section:**

Consumer Contract Tests for Http Requests

Scenario

A backend-for-frontend component, serving a mobile app, makes a POST request to the cinch-brew-order-component to create an Order.

Acceptance Criteria

GIVEN the Customer has selected what they want to drink
WHEN a valid request is made
THEN the Order is created

Example Code

```
// imports and mocks have been omitted
const orderClient = async (url: string) => {
  return axios.get(`${url}/orders`, { headers: { Accept: 'application/json' }, validateStatus: false });
};

const provider = new Pact({
  consumer: 'cinch-brew-app-bff-component', // the consuming participant
  provider: 'cinch-brew-order-component', // the providing participant
  port: 8091,
  pactfileWriteMode: 'merge',
  log: path.resolve(process.cwd(), 'logs', 'pact.log'),
  dir: path.resolve(process.cwd(), 'pacts'), // where to write the pact file to
  logLevel: 'debug',
});

beforeAll(async () => await provider.setup());

afterAll(() => provider.finalize());

 afterEach(() => provider.verify());

 describe('Orders HTTP API Consumer', () => {
  it('can create an Order', async () => {
    const interaction = {
      state: undefined, // corresponds to a provider's stateHandler; defining null here, uponReceiving: 'a request to create an Order', // a description of the test
    };
  });
});
```

Exemplars

github.com

README.md

cinch-brew

Cinch brew is a fictional software system demonstrating good observability. The system is designed with common cinch architectural patterns in mind - including backend and frontend.

An observability blueprint

An observability blueprint serves as the educational center for good observability patterns and practices used at cinch. Its purpose is to offer guidance by way of working examples of common scenarios you will come across whilst developing components.

What this is not

- A set of rules you must follow
- A resource to simply copy & paste
- A guide for engineering best practices
- Datadog documentation

Observability Tooling

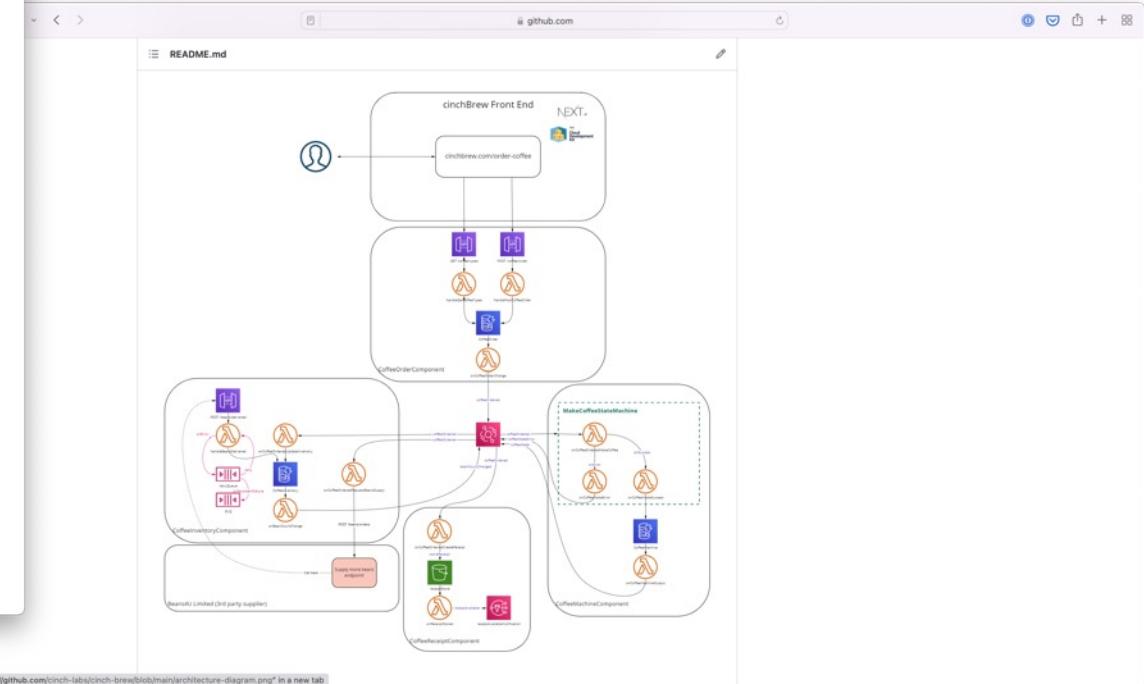
- Frontend
- Backend
- CI and Tests

The cinch brew architecture

The cinch brew architecture is based around common architectural patterns used by many components at cinch. It utilizes technologies and AWS services you may already be familiar with.

The system is made up of multiple components using both synchronous and asynchronous methods of communication. It also features communication with a 3rd party system, a state machine, document storage and queues, all common scenarios you may come across.

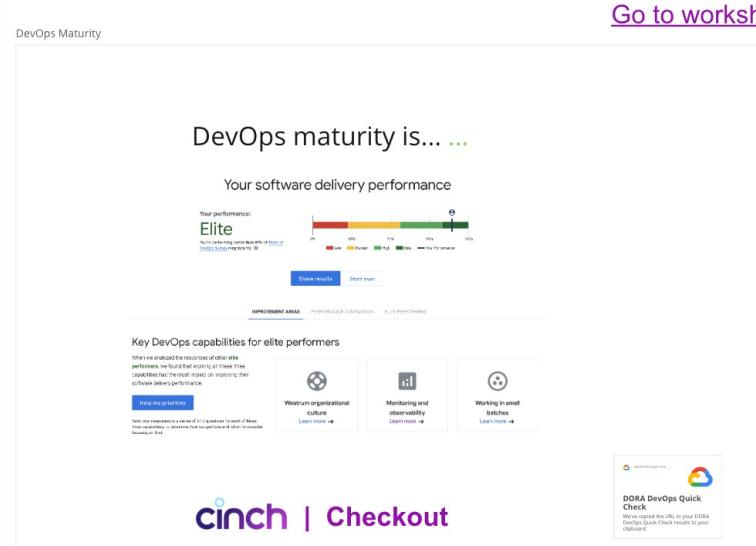
cinchBrew Front End



DevOps Checkpoint

DevOps Checkpoint Summary

16 June 2022



[Go to workshop](#)

Based on this maturity, the capabilities the squad should be focusing on are:



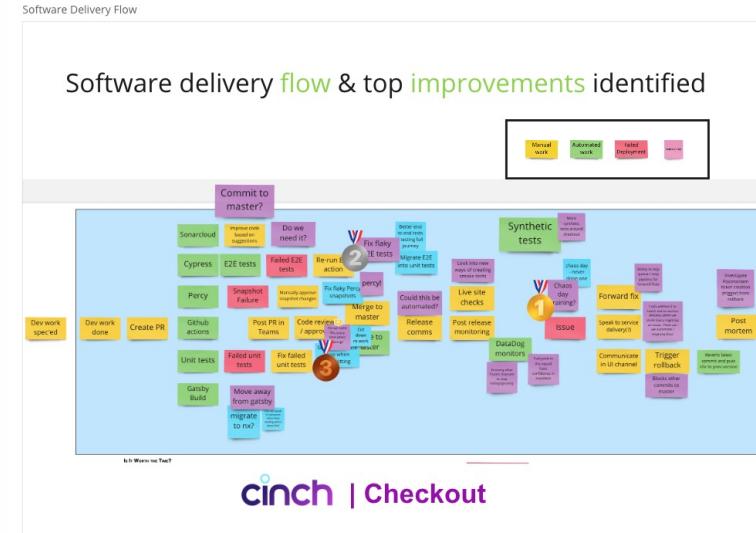
DevOps culture:
Westrum
organizational culture
| Google Cloud



DevOps measurement:
Monitoring and
observability | Google
Cloud



DevOps process: Working in small batches | Google Cloud



cinch | Checkout

Top improvements

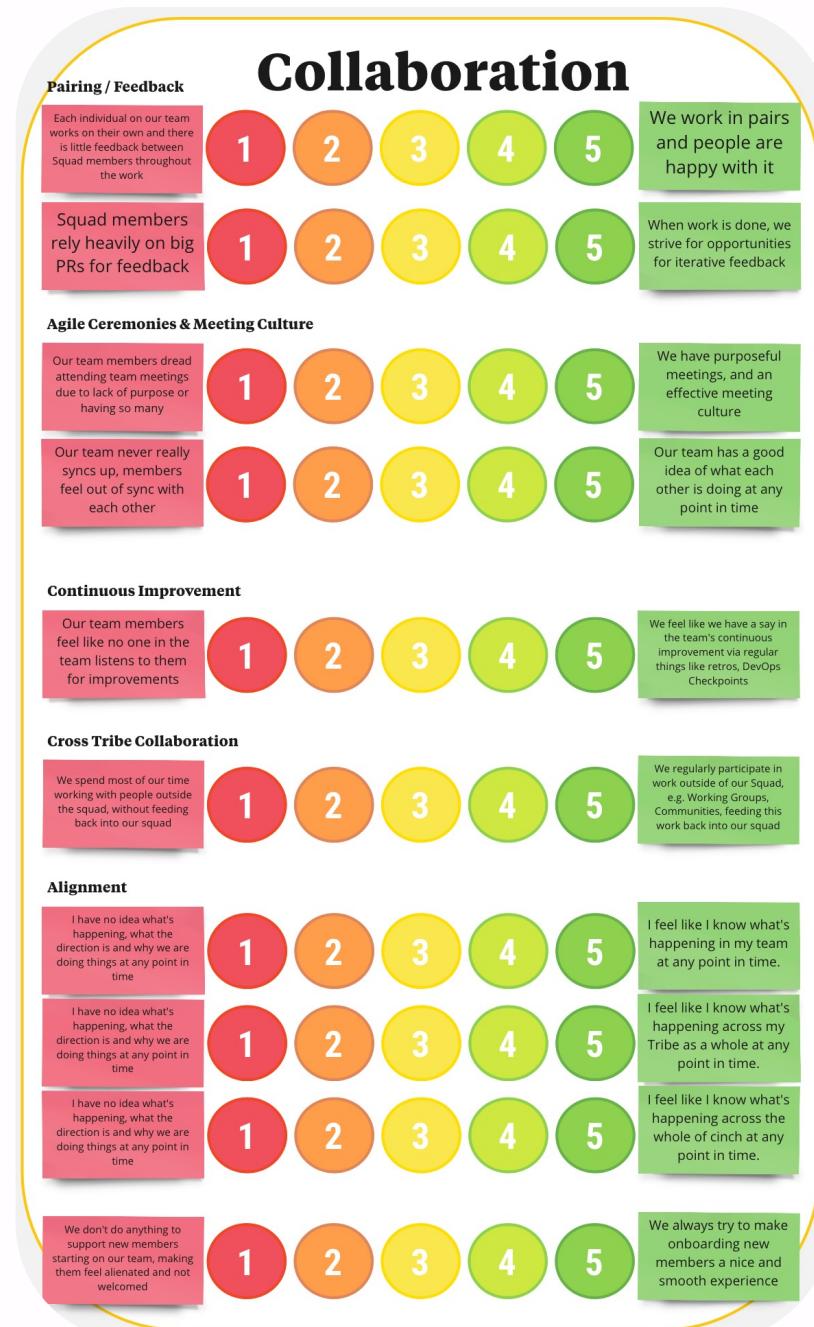
Top **improvements** identified

- 1. Chaos day
- 2. Fix flaky e2e tests
- 3. Do we need PRs every time when pairing?
- 4. Knowing what Teams channel to stop hiding/ignoring
- 5. More synthetic tests around Checkout
- Improve Percy



cinch | Checkout

Engineering Practices Checkpoint

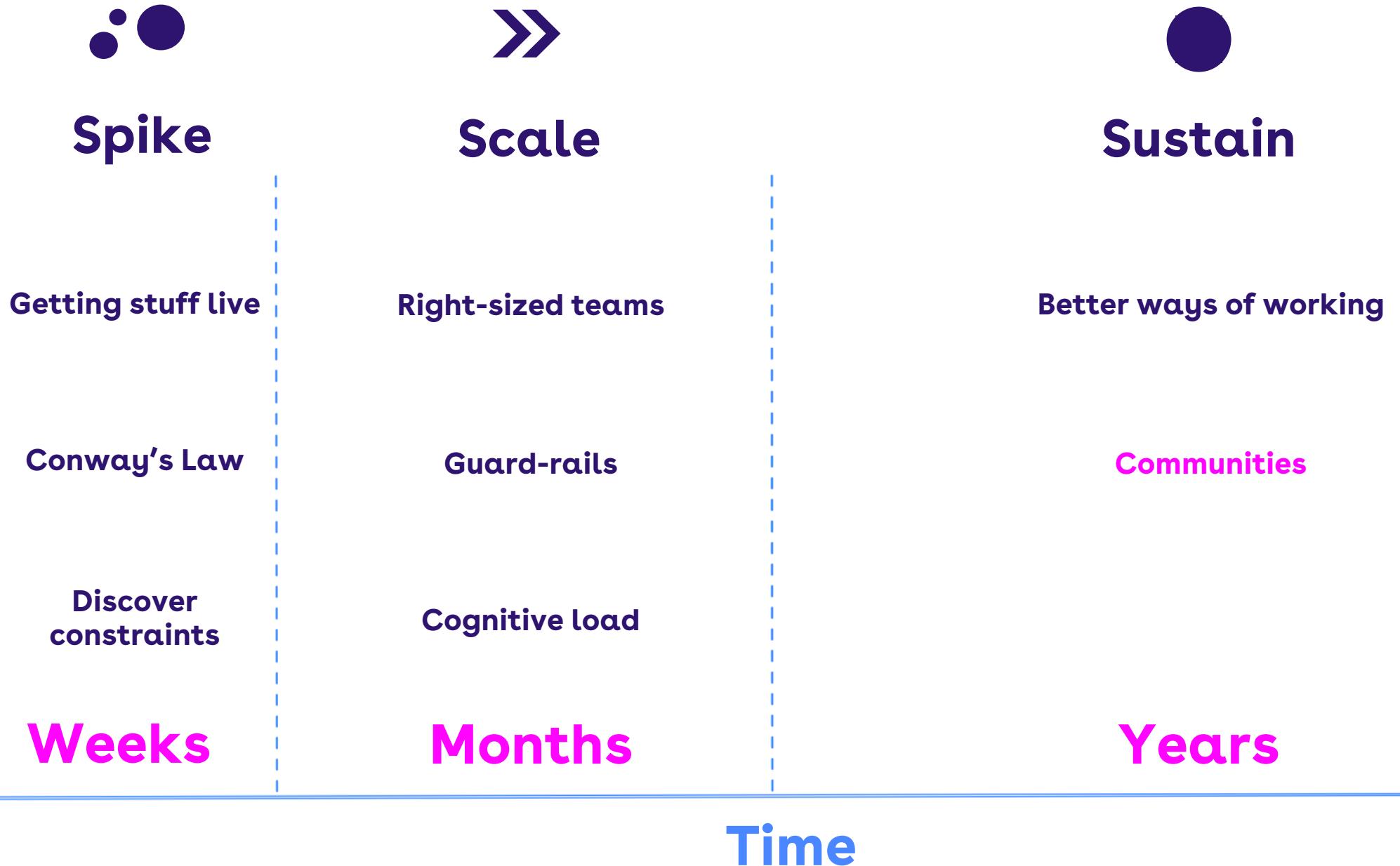


Practice Leads 

Staff Engineer 

Knowledge Management Lead 

Technical Trainer 





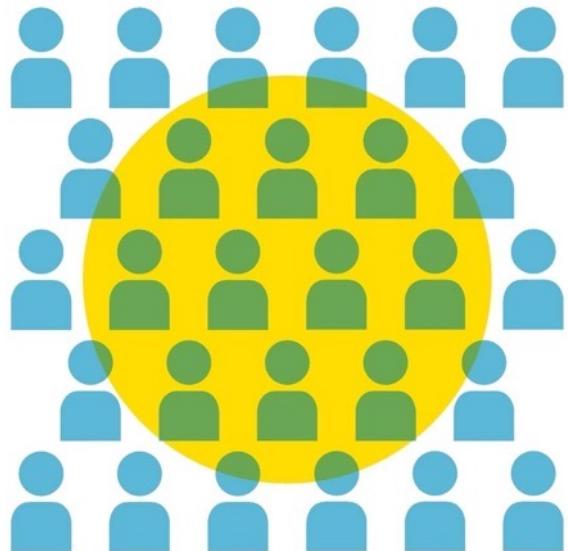
Communities are
powerful things

“Groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly”

- Wenger-Trayner

BUILDING SUCCESSFUL COMMUNITIES OF PRACTICE

Discover how connecting people makes better organisations



Emily Webber



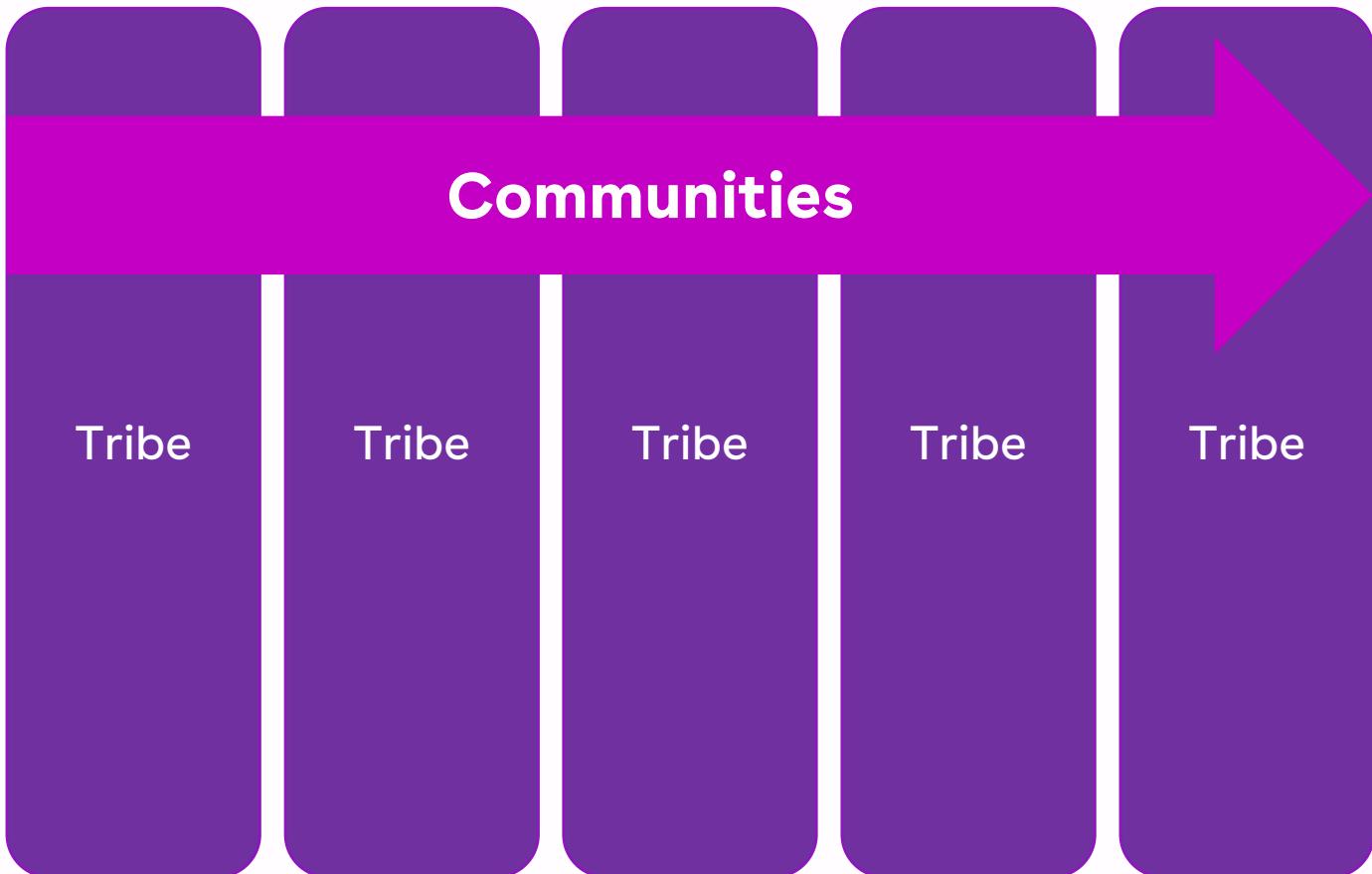
A GUIDE TO MANAGING KNOWLEDGE

CULTIVATING COMMUNITIES OF PRACTICE

ETIENNE WENGER
RICHARD McDERMOTT
WILLIAM SNYDER



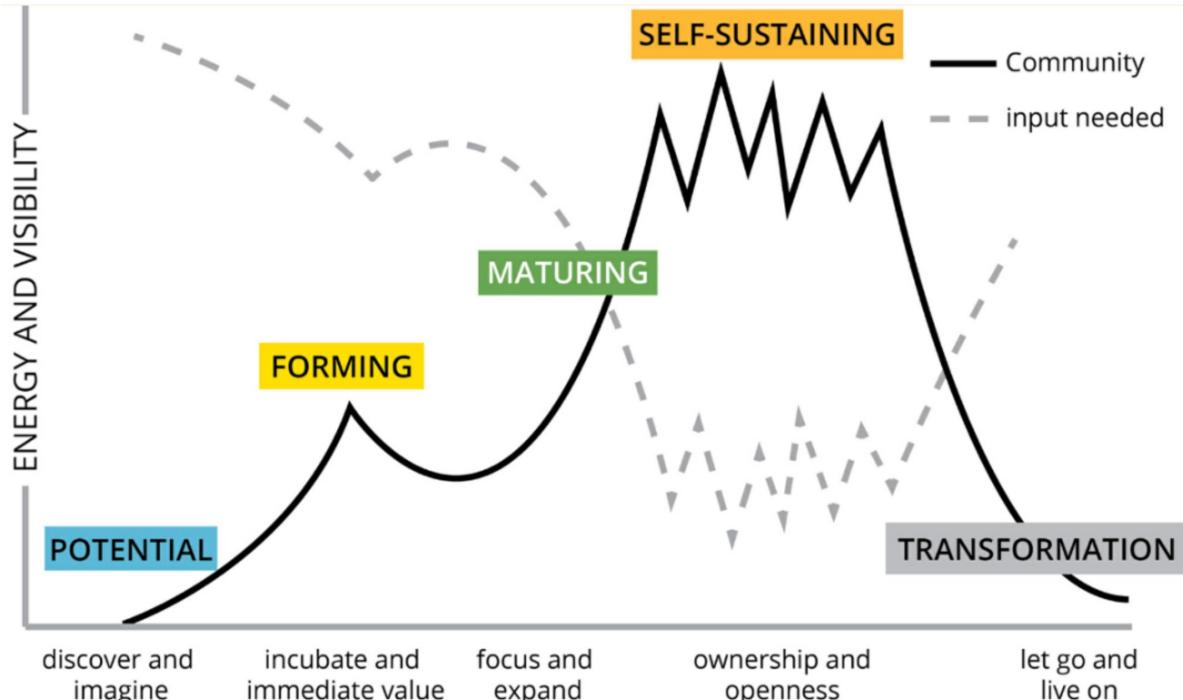
HARVARD BUSINESS SCHOOL PRESS





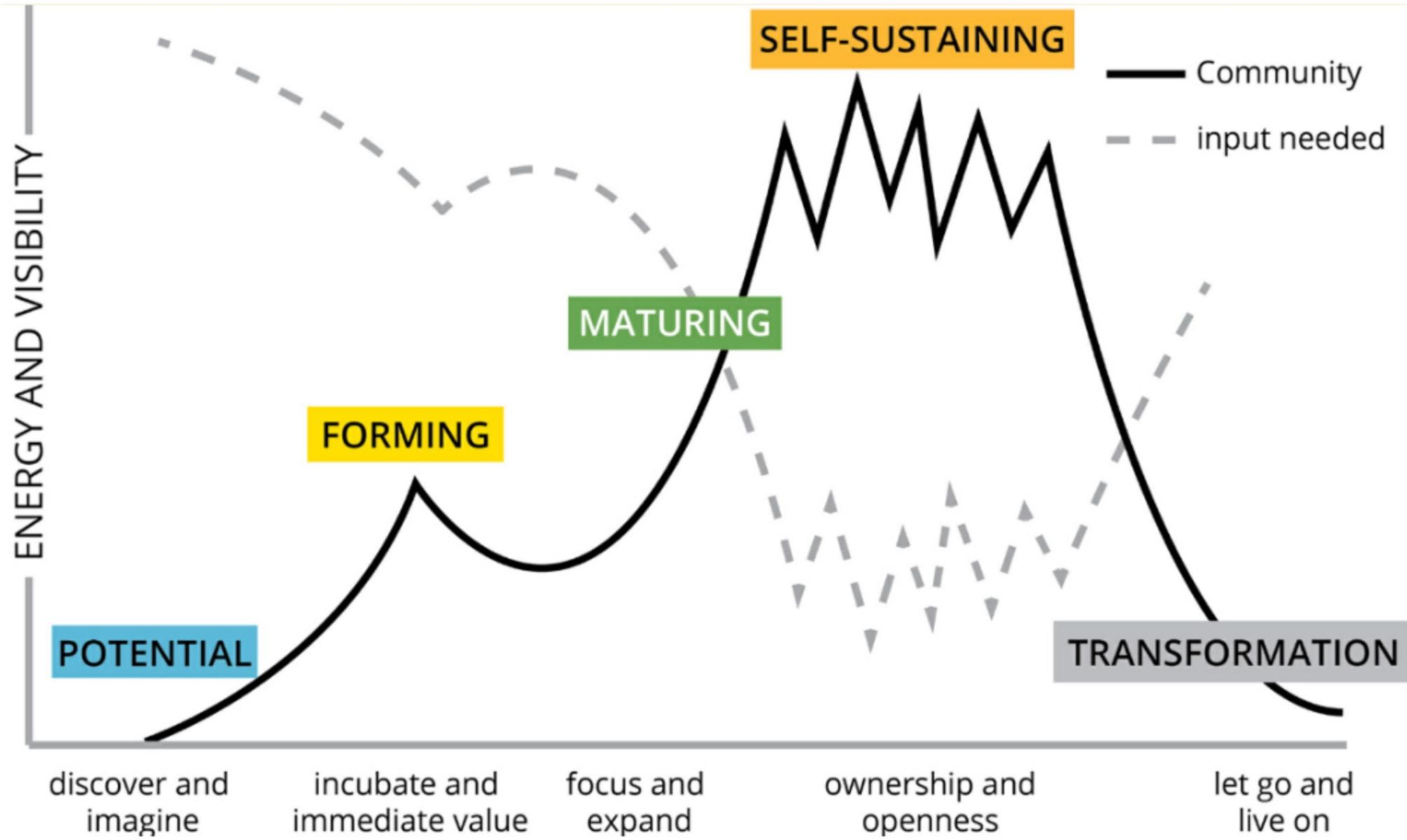
Communities

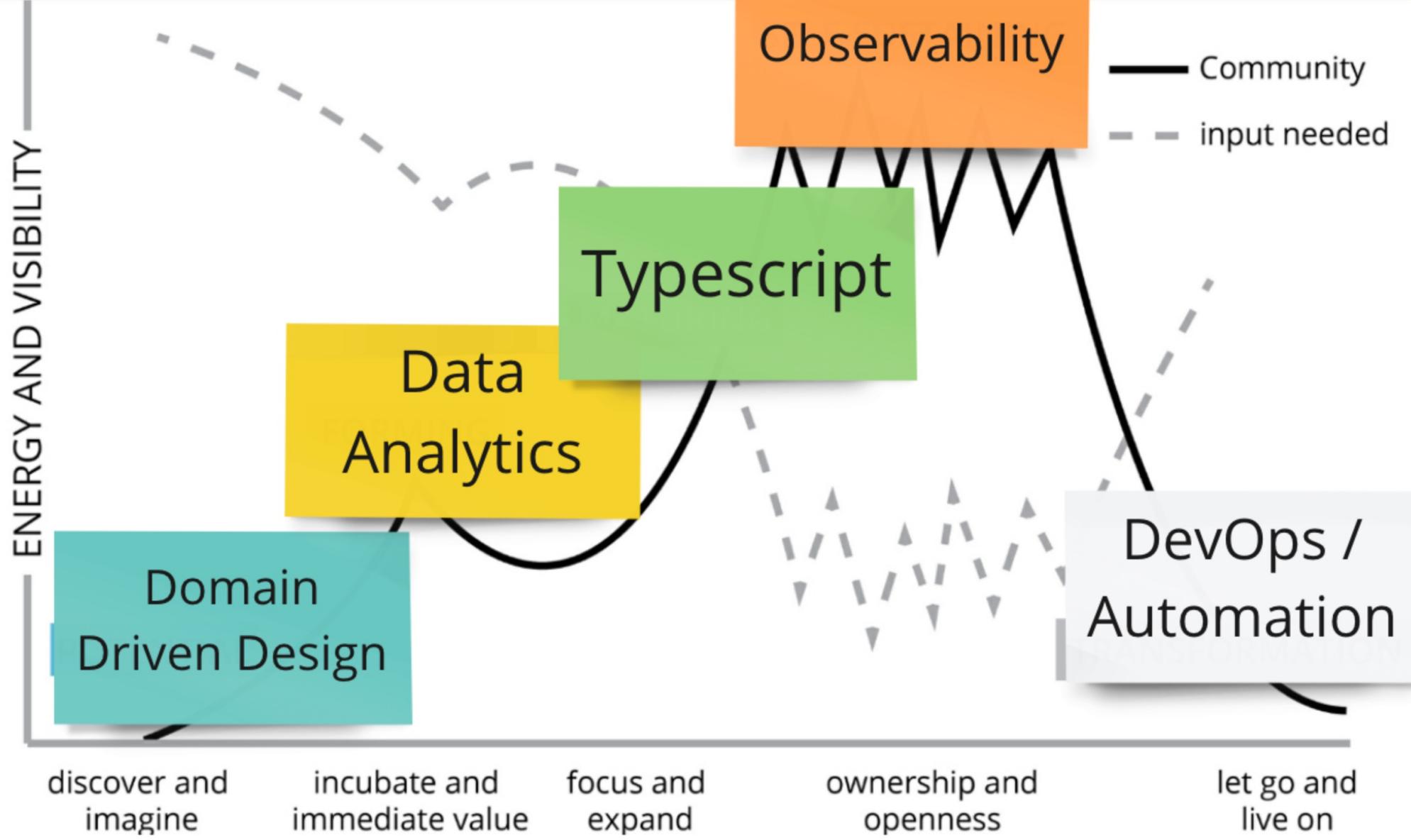
Silos



Emily Webber -
<https://hellotacit.com>







Blogging culture

Megan John Software Engineer Mar 28, 2022 (08:19)

Meg's Mumbles - My Start At Cinch

In General Topics 0 min 12 46 19 0 Follow

I thought I'd add some insight to starting from Square 1 and my first week at cinch.

But first, an introduction to me.

I'm Megan. Or Meg. Or MJ. I don't really mind. I began my journey with cinch last week (21/03/2022) after graduating from a physics degree and then a first job which unfortunately was not a good fit - it happens. My specialities in my physics degree delved into astro and space physics. From a young age I have been really intrigued by the workings of outside of Earth and so decided a physics degree was the thing for me. And it was amazing. I fell further in love with the planets that spin alongside us and how things came to be. But I fell in love with something new too. Coding.

Nasa Image of Jupiter with its polar aurora visible

Through lab experiments and analysing the data received and moving into my dissertation on Jupiter's ionosphere (which I will hopefully go into depths of if anyone is interested), I found a passion for making things work through a computer and so came to a crossroads in my choices after completing my Bachelors degree. Continue being an astrophysicist or become a software engineer. I think you can predict which I

Apostolis Apostolidis Principal Practice Engineer Mar 22, 2022 (10:19)

Autonomy of Schedule

0 2 min 4 34 Edit 9 0 Follow

I have what you would call in [Team-Topologies-speak](#), a number of **streams**. Squads are the unit of software delivery at cinch. As we have established other team types:

- Groups
- Communities (of Practice or Interest)
- Cubs (The DevOps Handbook Book Club)

not why

It about the purpose of all team types, but the 'how' of working outside of your squad. If you belong to a squad or a tribe, that is your core membership. You have made a promise to the people in your squad or tribe, and they take priority over other tribes.

mean that you can't participate in things outside the squad. In fact, we encourage you to do so, but it is vital that you:

- things visible
- communicate with your core membership - your squad

Katie Raby Software Engineer Apr 4, 2022 (12:56)

axe-con 2022 Digital Accessibility Conference

In Events 0 8 min 0 22 6 0 Follow

What is axe-con?

axe-con is a free, open and inclusive digital accessibility (a11y for short) conference held online. The conference focuses around building, testing, and maintaining accessible experiences. axe-con is suitable for developers, designers, business users, and accessibility professionals of all experience levels.

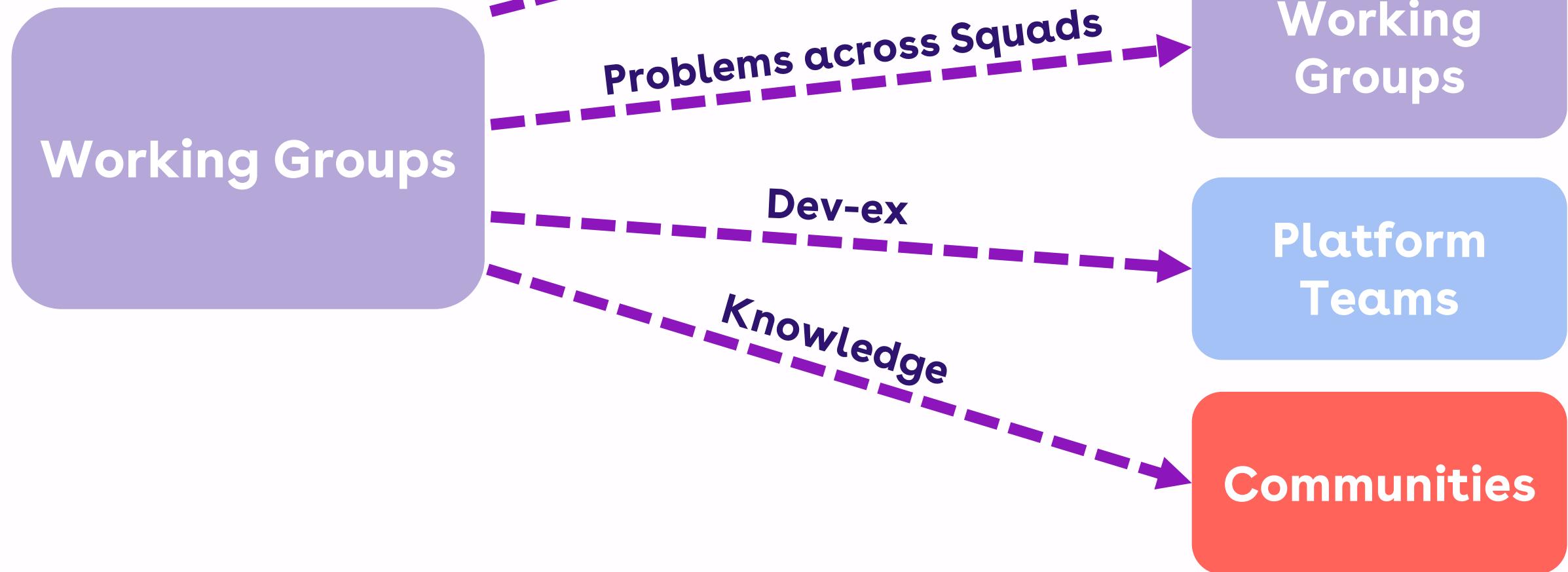
We live-streamed the axe-con sessions in the Manchester office (with lots of snacks - shout out to the sour cream & onion Pringles 😊). We connected with attendees in a11y at cinch, and had fun debating the topics we were watching in real-time.

Takeaways from a few talks we watched

remove semantic meaning

Style Sheets (CSS) is what's used to give websites life and to make them look better, to quote a phrase from a legendary Marvel movie:

What did we do with our Working Groups?







Spike

Getting stuff live

Conway's Law

Discover constraints

Weeks

Scale

Right-sized teams

Guard-rails

Cognitive load

Months

Sustain

Better ways of working

Communities

Observability

Years

Time

Observability is the ability to understand how a software system is behaving internally by evaluating its outputs.

What do we mean by this?

Are the **business transactions** we think are happening, actually happening?

Observe **what we care about**, as a **business**, as a provider of services to a customer

What's our appetite for risk? If we've got good olly practices we should know when it breaks, don't avoid deploying

Working on the (virtual) walls

The image displays two side-by-side screenshots of cloud monitoring dashboards, illustrating how different teams use monitoring tools to track system health and performance.

Orders Squad - Motherboard (Left Dashboard):

- Are orders looking healthy?** (23 widgets)
- Is our part exchange service healthy?** (12 widgets)
- What changes have been released?** (6 widgets)
- Do we have any incidents?** (1 widget)
- Are we seeing errors?** (5 widgets)
- Are my APIs healthy?** (12 widgets)

Search Squad Standup Board (Right Dashboard):

- Search squads Links:** A grid of links to various tools:
 - Jira
 - Repos
 - Miro
 - Dashboards
 - Docs
 - AWS
 - APM - Search
 - APM - Rec
 - CloudTrail
 - RUM
 - Sonar
 - Logs
- Health at a Glance:** A summary section with the following data:
 - Monitor statuses:** Shows 35 monitors in **OK** status.
 - Alerted monitors:** No matching entries found.
 - LCP average:** A chart showing LCP average over time, with values ranging from 0 to 500.
 - Unique user crashes:** A table showing unique user crashes across different categories:

Category	Value
console error: T...	39.00
console error: p...	35.00
console error: ...	11.00
console error: ...	8.00
console error: T...	5.00
console error: E...	2.00
console error: F...	1.00
 - Unique user crashes ...** (Another chart showing unique user crashes over time, with values ranging from 0 to 4).
 - Unique user crashes** (A large orange box displaying the value 32).



Spike

Getting stuff live

Conway's Law

Discover constraints

Weeks

Scale

Right-sized teams

Guard-rails

Cognitive load

Months

Sustain

Better ways of working

Communities

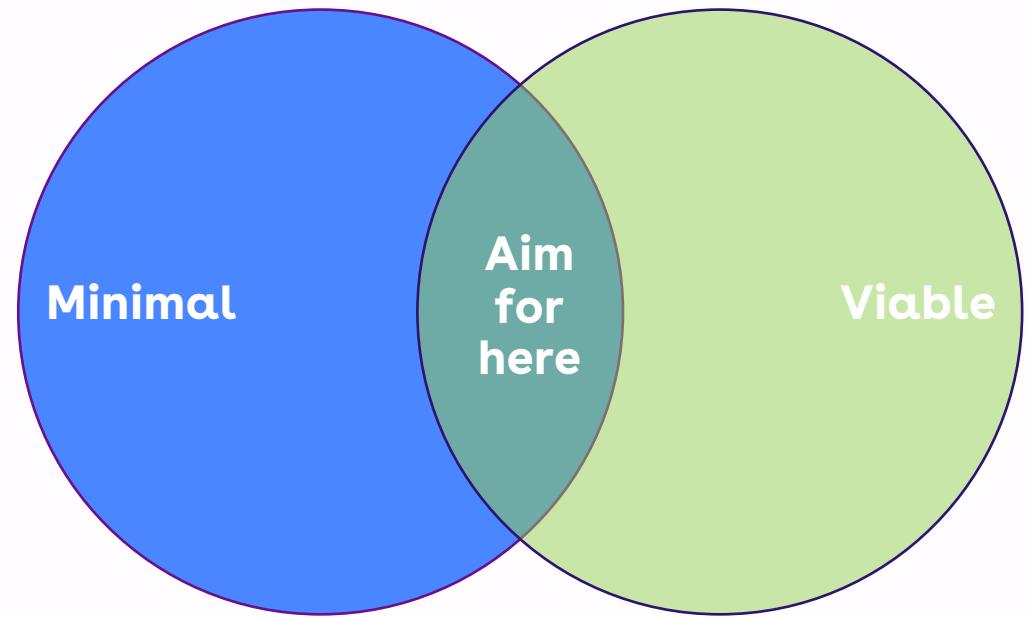
Observability

Years

Time

In summary

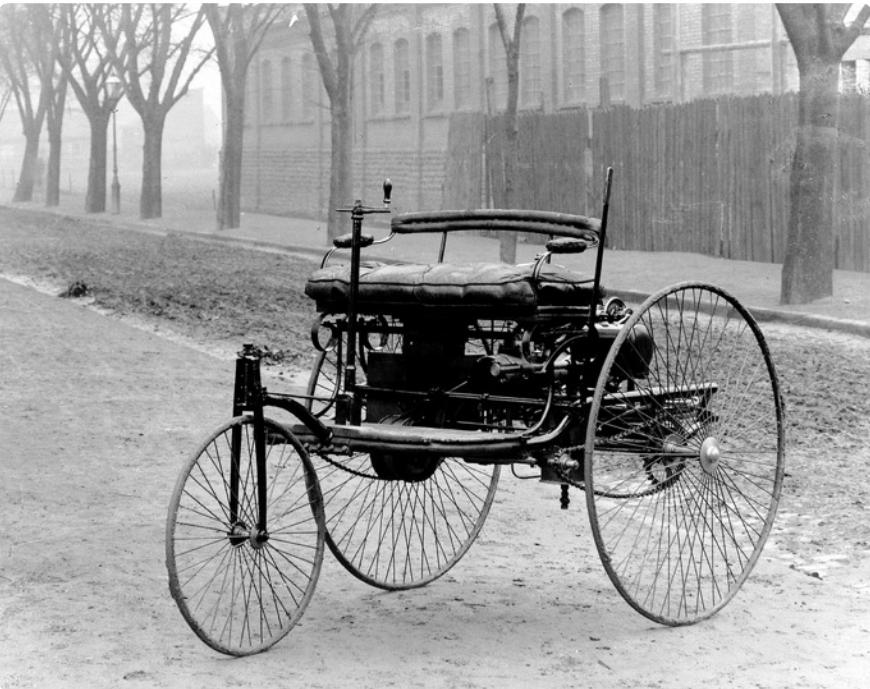
Minimal viable everything.



Feedback is a gift



Change when you need to



Achieving unicorn scale Using 100% hypergrowth metrics
Spike. Scale. Sustain. We're not afraid to fail.

Who are we?

What do we mean by ?

From spike to sustain in 2 years

The 'starting a start-up in 2019' chapter goes

In 2019 cinch was 20 people over 4 teams

In 2022 cinch is over 450 people over 20 teams

The walking skeleton

"We need an MVP"

Spike. Scale. Sustain.

"We need an MVP" was actually "We need an MVO"

Disclaimer
We don't get any referral fees from Team Topologies.

Spike. Scale. Sustain. Weeks Months Years

Spike.

"What is the simplest thing we can program that will convince us we are on the right track?"

Start lean, it's temporary

So how do we define a team?

Discover the business domain together, collaboratively. Understand what the core sub-domains are that support the business domain. Define context boundaries, and form teams around them.

Event Modeling

Bounded Contexts

Yes, cross-functional teams

The 'starting a start-up in 2019' chapter goes

Still not the Spotify model!

"Organisations design systems that mirror their own communication structure"

Expectation Reality

What are you optimising for?

Technology provides lots of positive feedback loops like pointing teams in the direction of practices and ways of working we would like to actively encourage.

Inventory Orders

Inventory Orders

Add teams to the pile.

Every complex system has a constraint. Focusing on anything else is waste.

Or... scale horizontally

Definitely, not the Spotify model!

Disclaimer
We don't get any referral fees from Team Topologies.

Cognitive load
"The amount of mental effort being used in the working memory"

Team Cognitive Load

Team Cognitive Load

Either we decided the technical constraints now, or they'd appear as we scale

Typical
No relational databases, User needs over time, Broad front-end, Ditching the API

Either we decided the technical constraints now, or they'd appear as we scale

Polyglot
Observability, Large data sets, User needs over time, Task-based TDD, Microservices

Orchestration
Observability, Large data sets, User needs over time, Task-based TDD, Event-driven

100% capacity

"If we stress the team beyond its cognitive load capacity, performance and output will drop significantly. This is known as cognitive overload. It's important to understand balance without the space to consider if those are in the team's best interests."

Stream-aligned teams

What are types in TypeScript?

What's your AWS configuration?

How to make payment of a car easier?

Focus on selling cars

Always learning the good bits

Intrinsic

Space

Space

100% capacity

Stream-aligned teams

Enabling teams

Platform teams

Complexified sub-system teams

Minerals dataset - 650 Business*

X-as-a-service

Facilitating collaboration

Measuring our platform

Becoming knowledge stewards

Technology Blueprint

Exemplars

DevOps Checkpoint

Engineering Practices Checkpoint

Practice Leads ✓ Staff Engineer ✓

Knowledge Management Lead

Technical Trainer

Focus on the gaps

What do we mean by this?

Observability is the ability to understand how a software system is behaving internally by evaluating its outputs.

Working on the (virtual) walls

In summary

Minimal viable everything.

Feedback is a gift

Change when you need to

Thank you.

We focused on getting stuff done, using **Contextual** for our teams, using **Concerns** about **The Theory of Constraints**, which **Engineering** had optimal cognitive load, **Design** had great **Observability**, **Marketing** had **Communities** that care about things like **Observability**.



Spike

Getting stuff live

Conway's Law

Discover constraints

Weeks

Scale

Right-sized teams

Guard-rails

Cognitive load

Months

Sustain

Better ways of working

Communities

Observability

Years

Time

So, how did we scale a ?

One size doesn't fit all.

Lean Agile

**Team
Topologies**

**Event-Driven
Architecture**

Wardley mapping

DevOps

Domain-Driven Design

The Spotify Model™

**The Theory of
Constraints**

**Throw all of that together and what did
we get?**

We focused on getting stuff live, using Conway's Law for our teams, thinking about The Theory of Constraints, whilst making sure that right-sized teams with guard-rails had optimal cognitive load, better ways of workings fueled by communities that care about things like Observability.

Thank you.

Toli Apostolidis - Engineering Practice Lead

@apostolis09

Andy Norton - Head of Engineering Practice

@andyjnorton