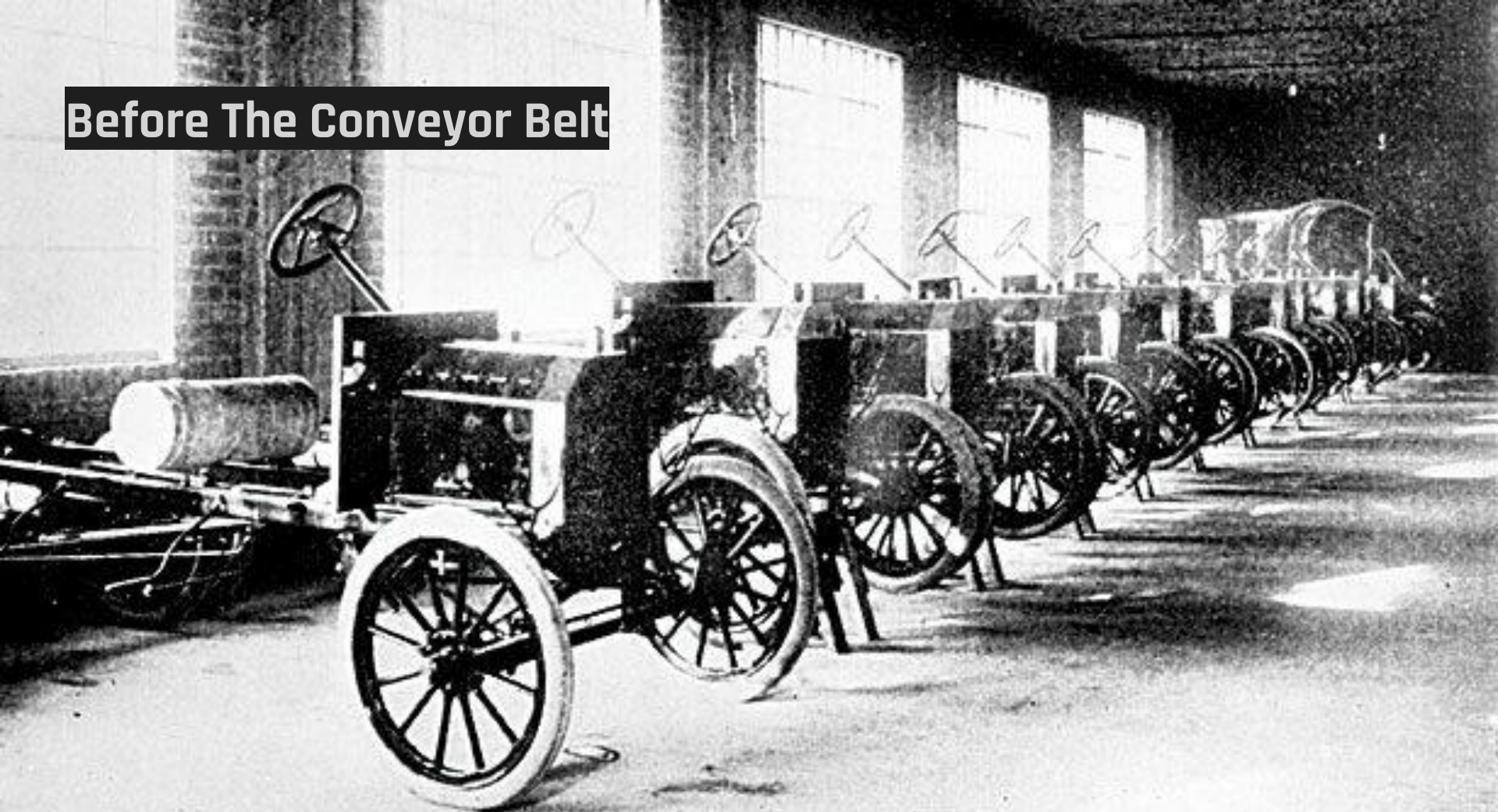


Tickets Create Silos

How to Unlock True Interdisciplinary Collaboration

Before The Conveyor Belt



With The Conveyor Belt



**Building Software Is a Design
Process,
Not an Assembly Line!**

Facing Reality

**“Agile now means, we do
half of Scrum poorly and
Use Jira”**

(Andy Hunt, “The Pragmatic Programmer”)





Coders Had a Vision

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.

Even Small Teams Always End Up With Waterfall

Agile/Scrum

No Product Owner

External Stakeholders

Predictability

Roadmaps

Business Pressure

Efficiency

Work on my own

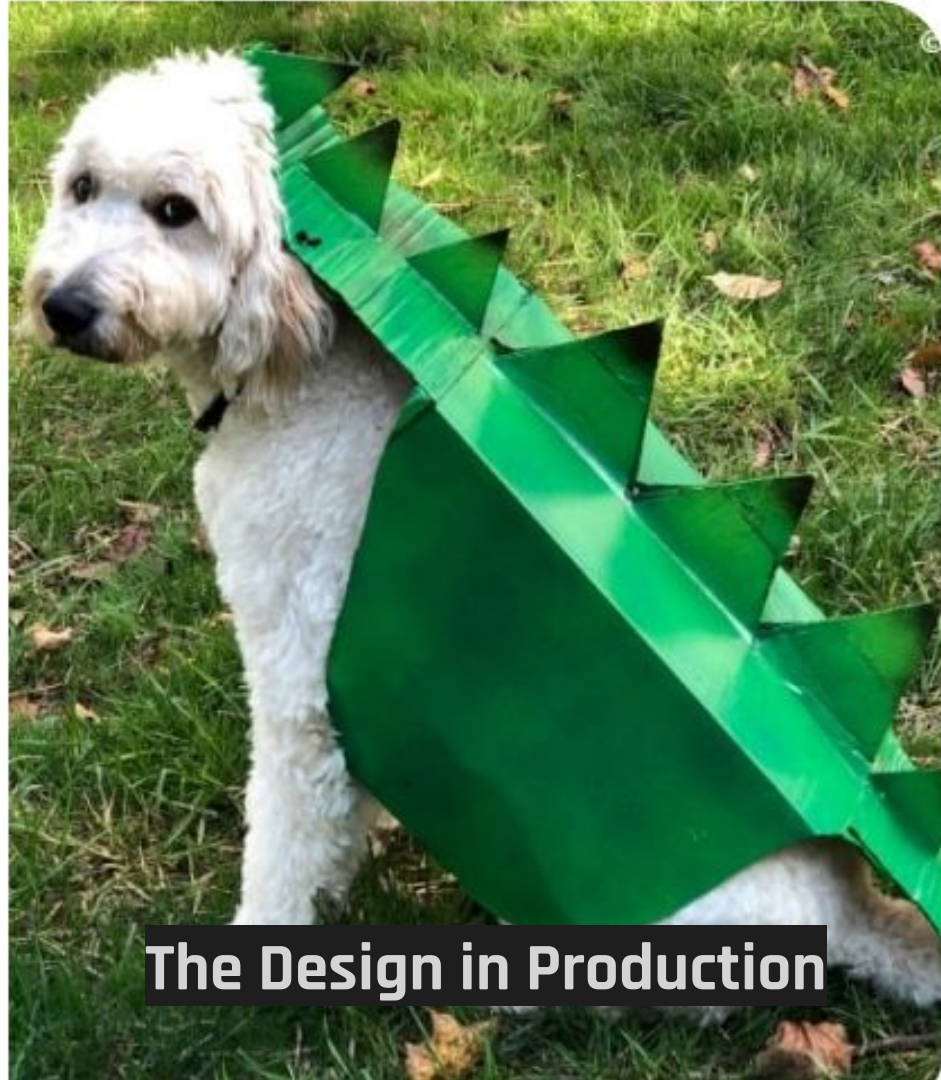
Different Dev &
Product Orgs

Scrumfall

The Design in Figma



https://www.reddit.com/r/FigmaDesign/comments/10rh6k4/is_it_just_me_or_lightbox/



The Design in Production

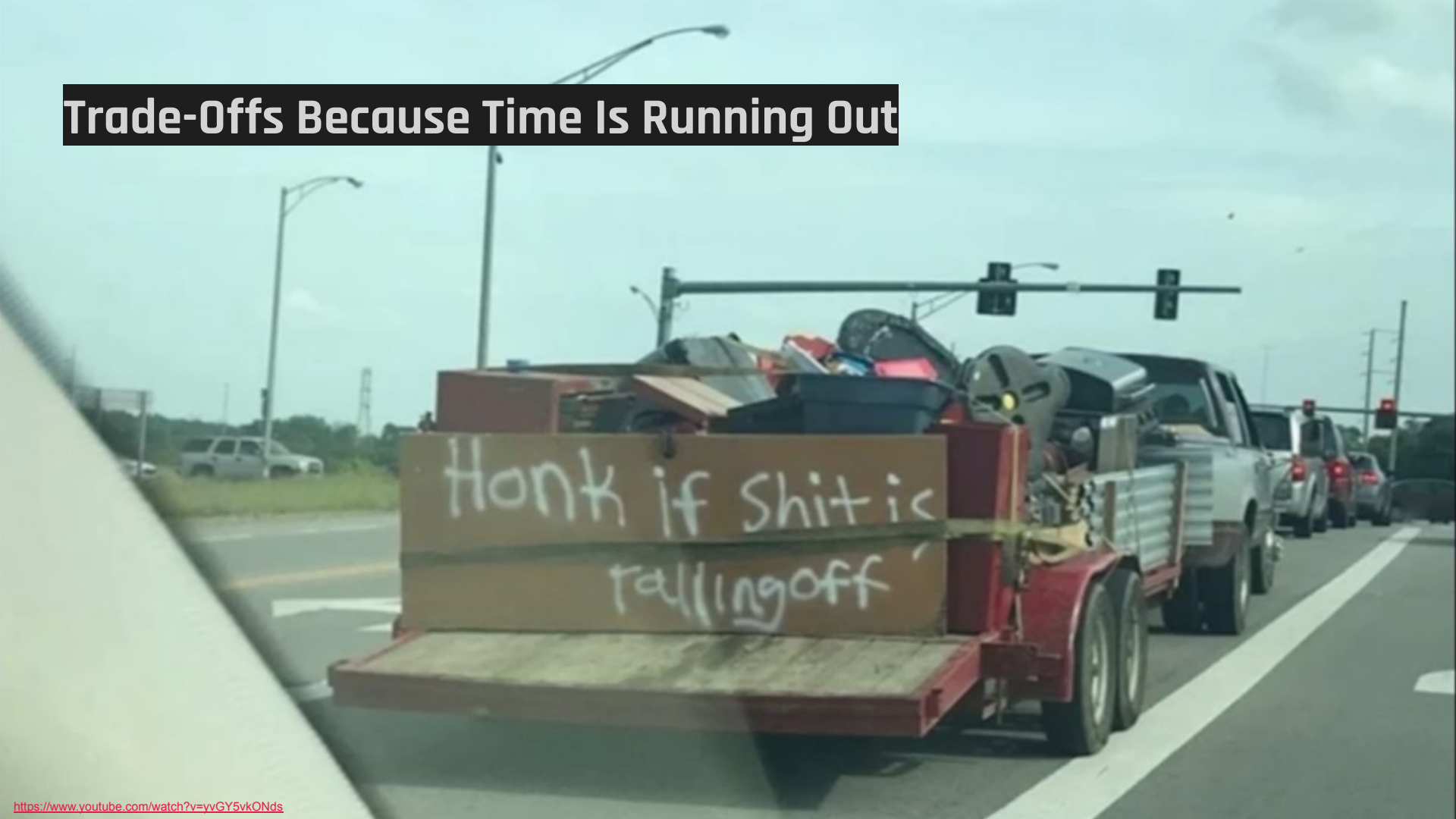
Yeah, If You Could Fill Out a Jira Ticket

That Would Be Great

Hyperactive Hive Mind



Trade-Offs Because Time Is Running Out



**Building Software =
Making Decisions in Continuously
Evolving Systems.**

Slicing Work

Slicing Work

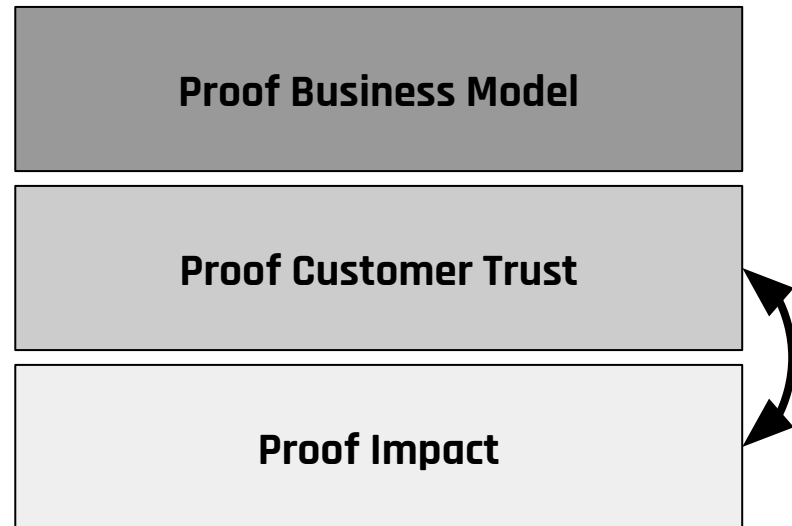
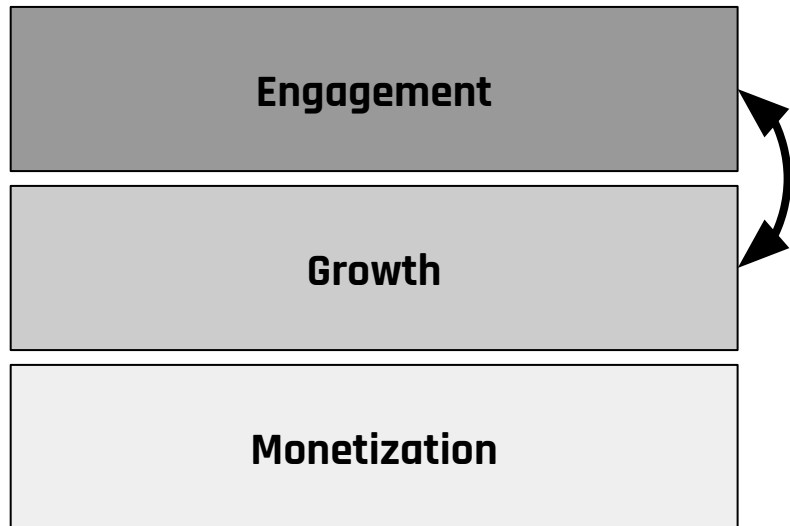
Slicing Objectives

Slicing Problems

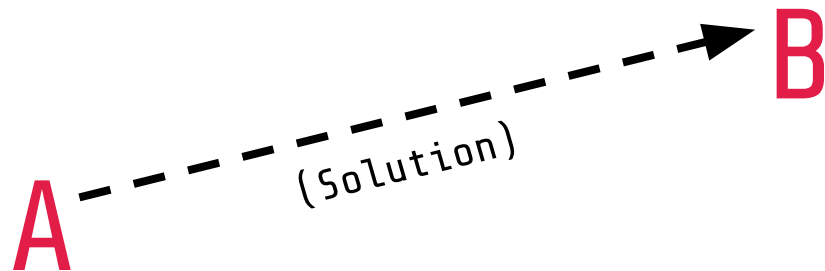
Slicing Solutions

Slicing Delivery

Slicing Objectives: Force Ranked



Slicing Problems: Precise Context & Outcome Framing



Current Context:

...
...
...

Desired Outcome:

...
...
...

Problem: Manual license checks creating risk and extra effort.

Strategic Context

> What belongs here?

- **Customer Pressure:** [] has stated they need a solution. If not addressed, they may need to adopt a secondary system, **introducing the risk of new or specific [] customers will be moved to that system.**
- **USP Misalignment:** Our product is positioned as an **automated compliance solution**, but the current handling of manual control drivers contradicts that.
- **Compliance Risks:** M is overdue, **creating**

Desired Outcomes

> What belongs here?

1. **Automated Workflows** – The system should ensure that **manual control drivers are revalidated within the control interval**, with an automated process to support compliance.
2. **Reduced Customer Dependence on External Tracking** – Customers should no longer need to track manual control checks outside our system.

Slicing Problems: Appetite as a Strategic Allocation

Appetite

Fixed Time
Variable Scope

Slicing Problems: Appetite as a Strategic Allocation

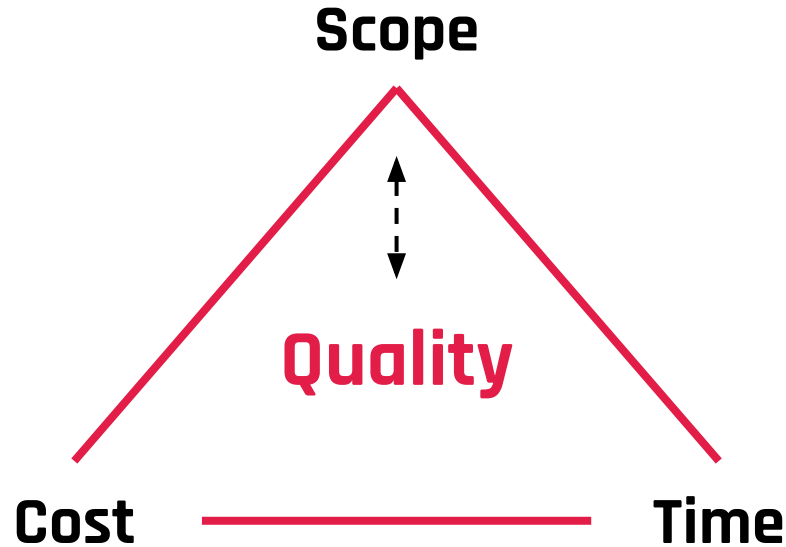
Appetite

Fixed Time
Variable Scope

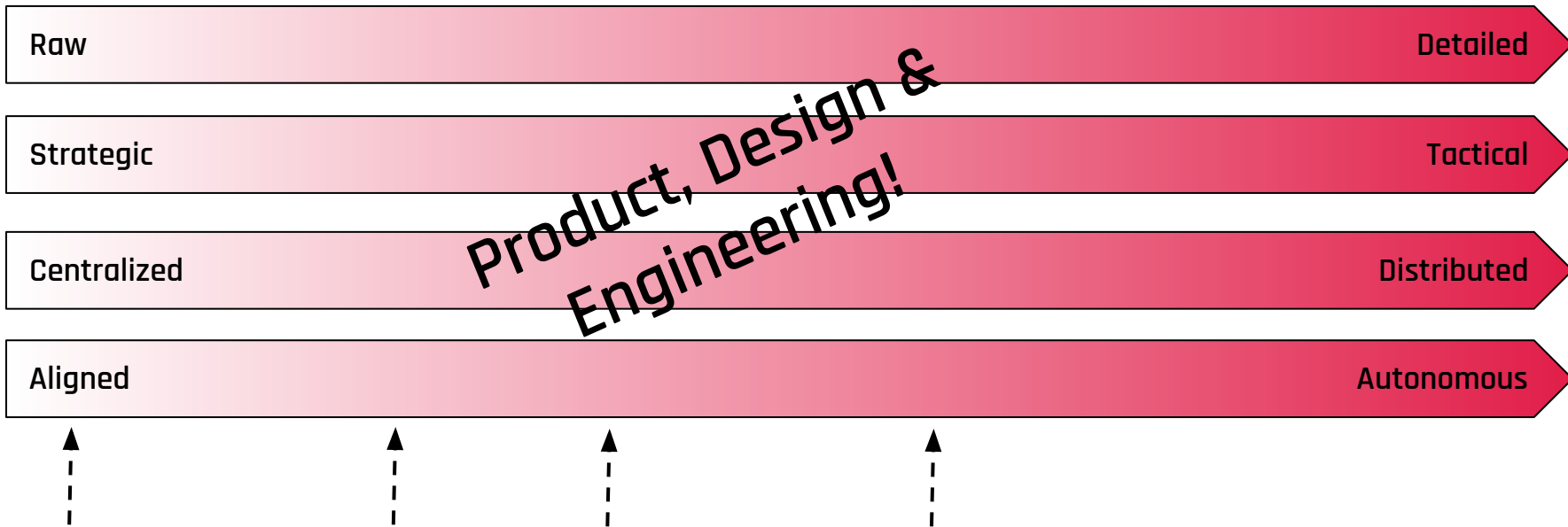
Estimates

Fixed Scope
Variable Time

Variable Scope Guarantees Quality!



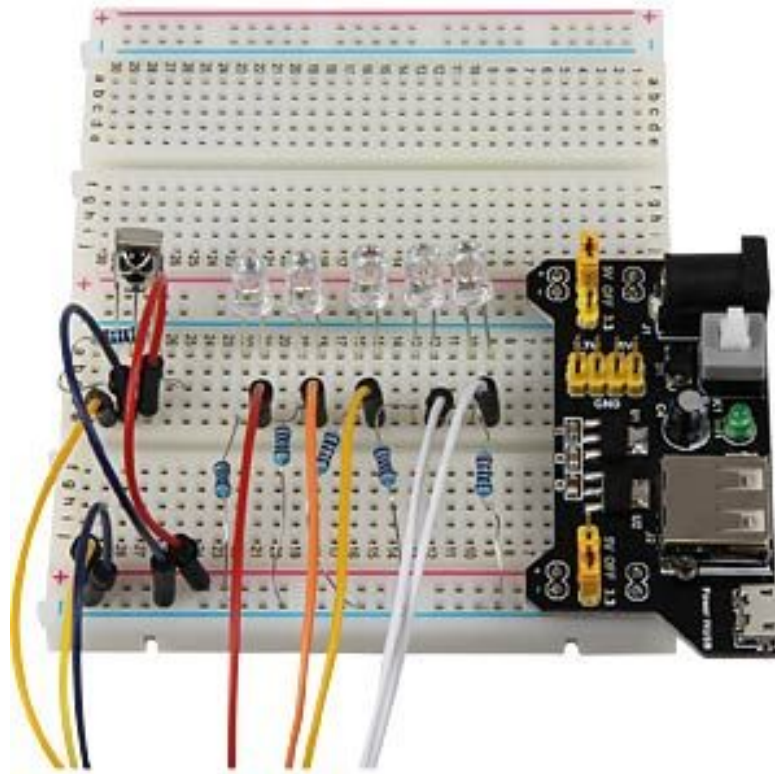
Slicing Solutions: Collaborating on Decisions



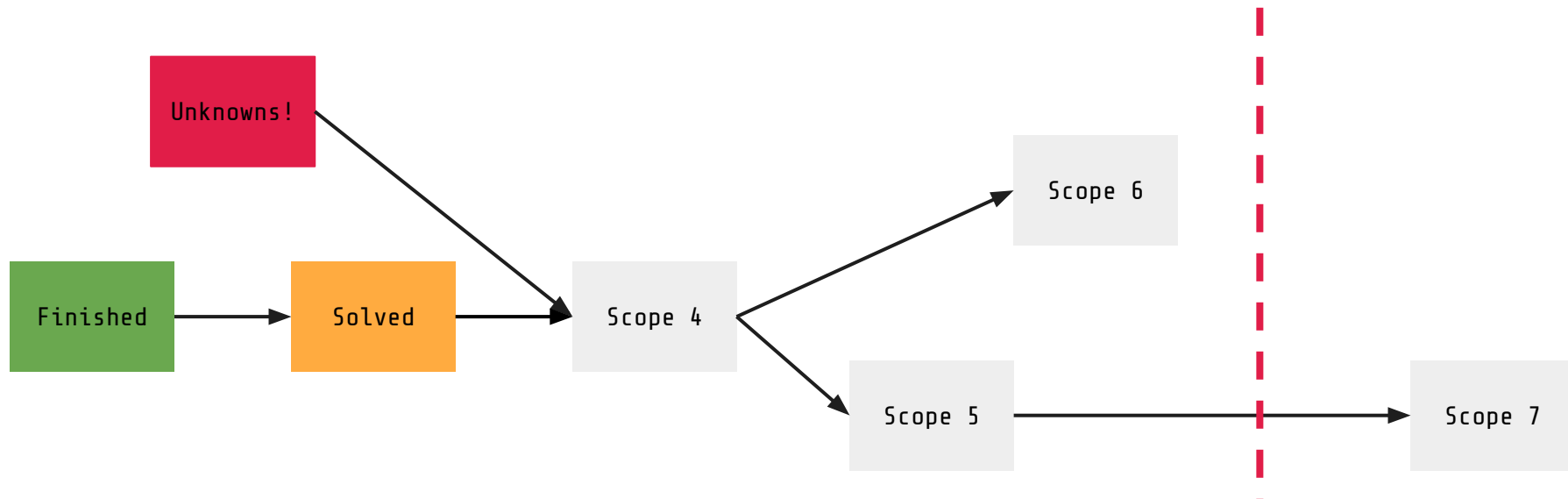
Start at a Blank Slate & collaborate on a Problem



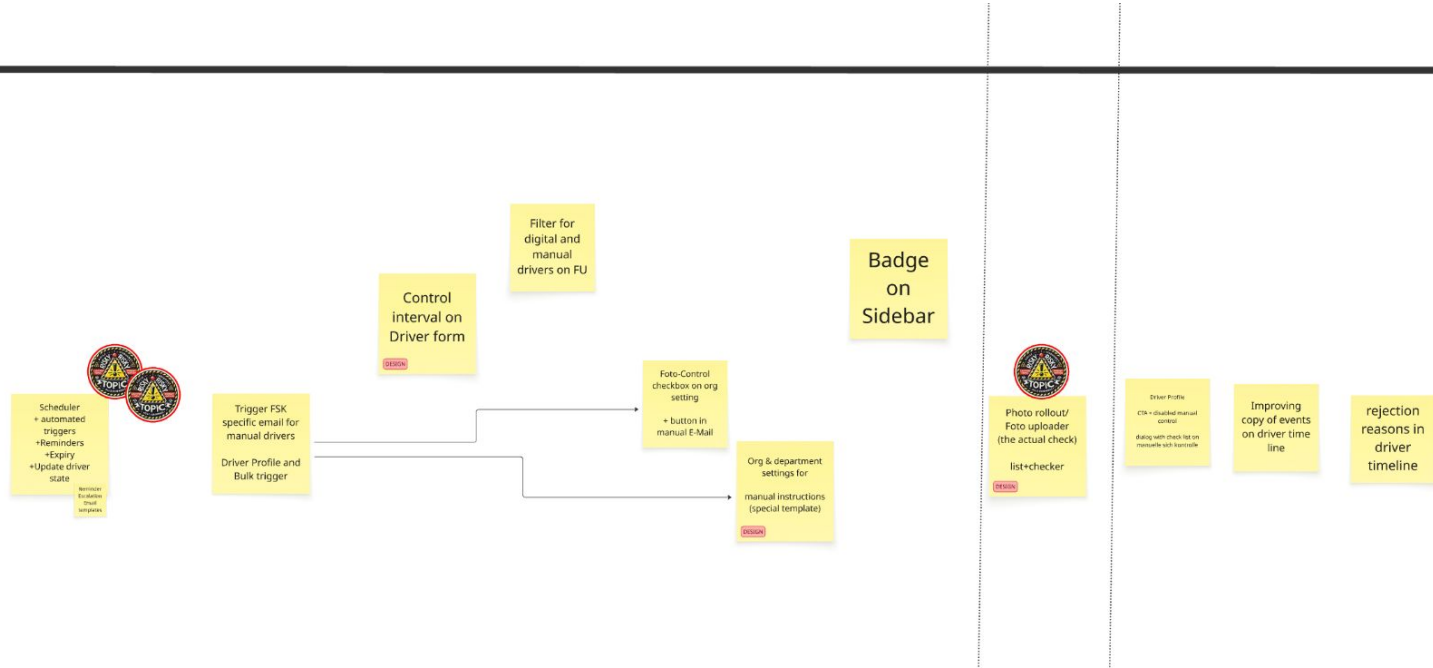
Breadboarding: Finding the Elements



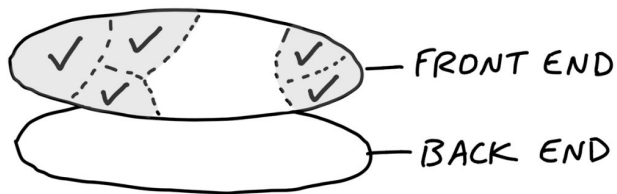
Slicing Solutions: Always Ready to Cut!



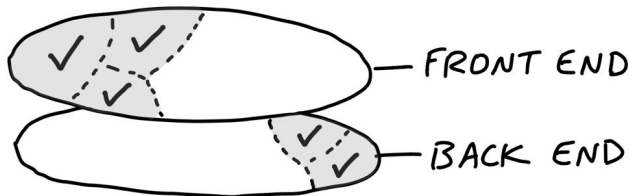
Scoping



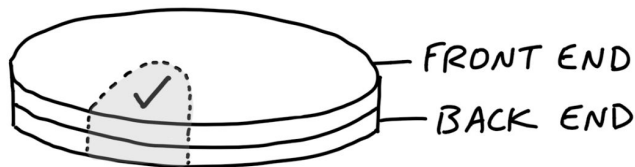
Slicing Delivery: End to End!



NOTHING WORKS



NOTHING WORKS



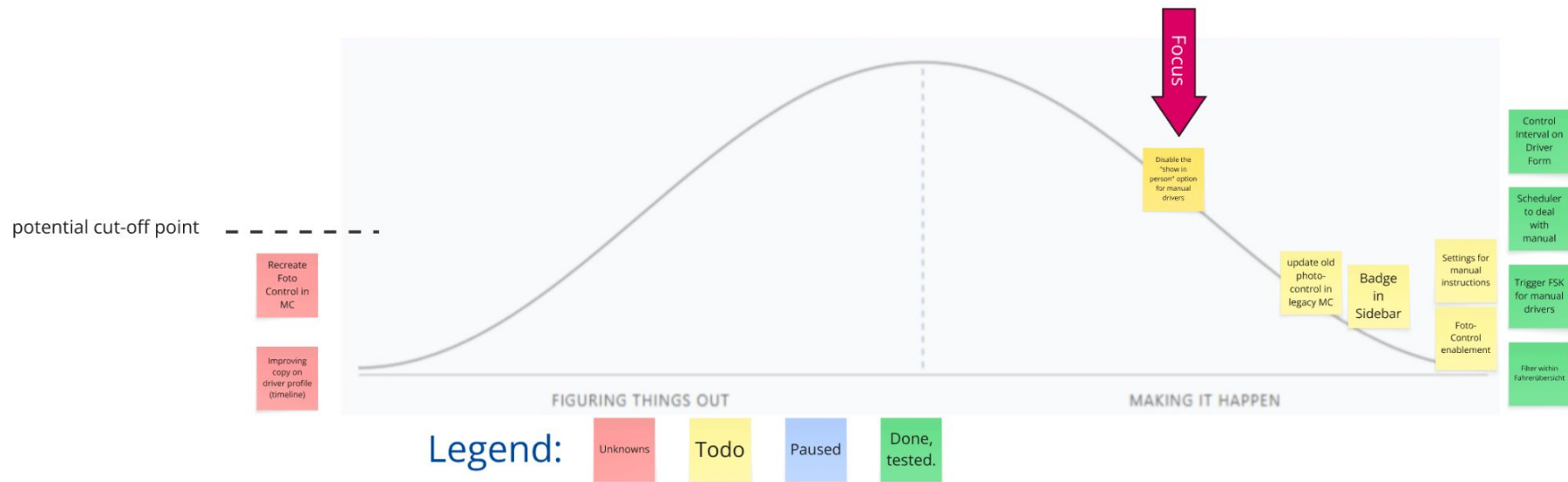
SOMETHING WORKS!

Implement by structure, not by person!

Slicing Delivery: A Piece of Cake!



Reporting on Progress



Slicing Work

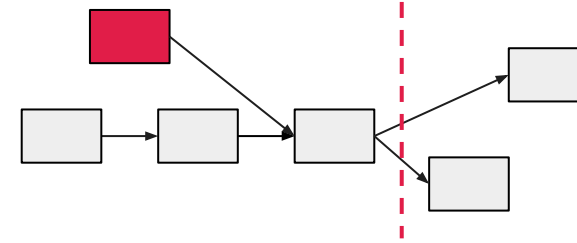
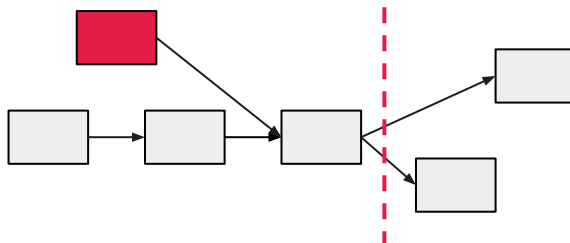
Slice Objectives



Slice Problems



Slice Solutions

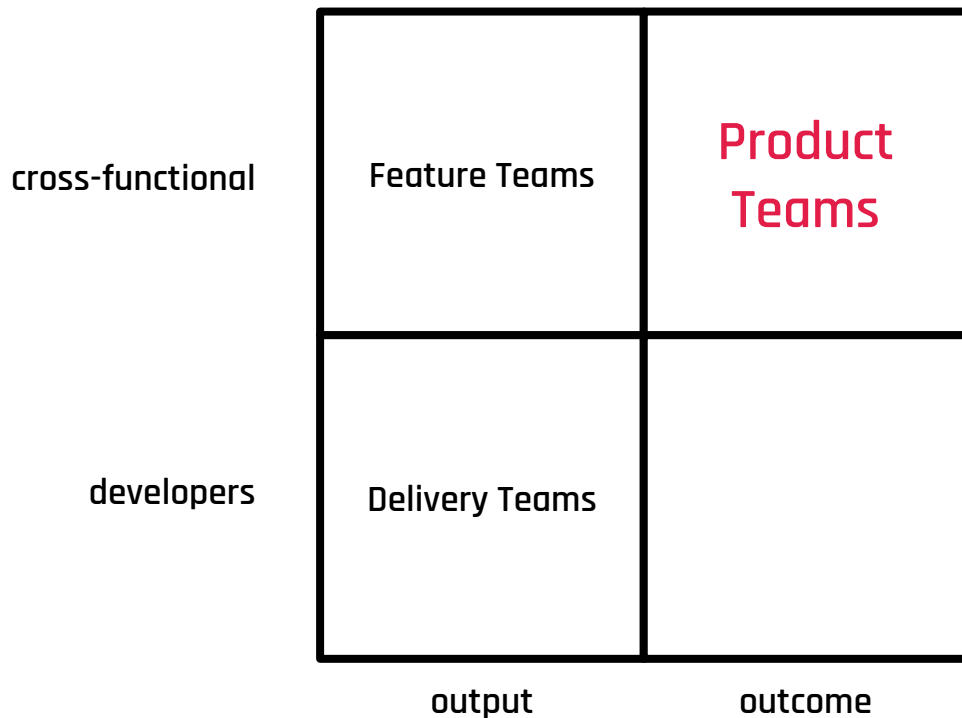


Slice Delivery



Aligning Teams

Empowered Product Teams



Product Teams: Everything They Need

Product Manager (Not a PO!!)

Responsible for customer value + business viability risk

Designer

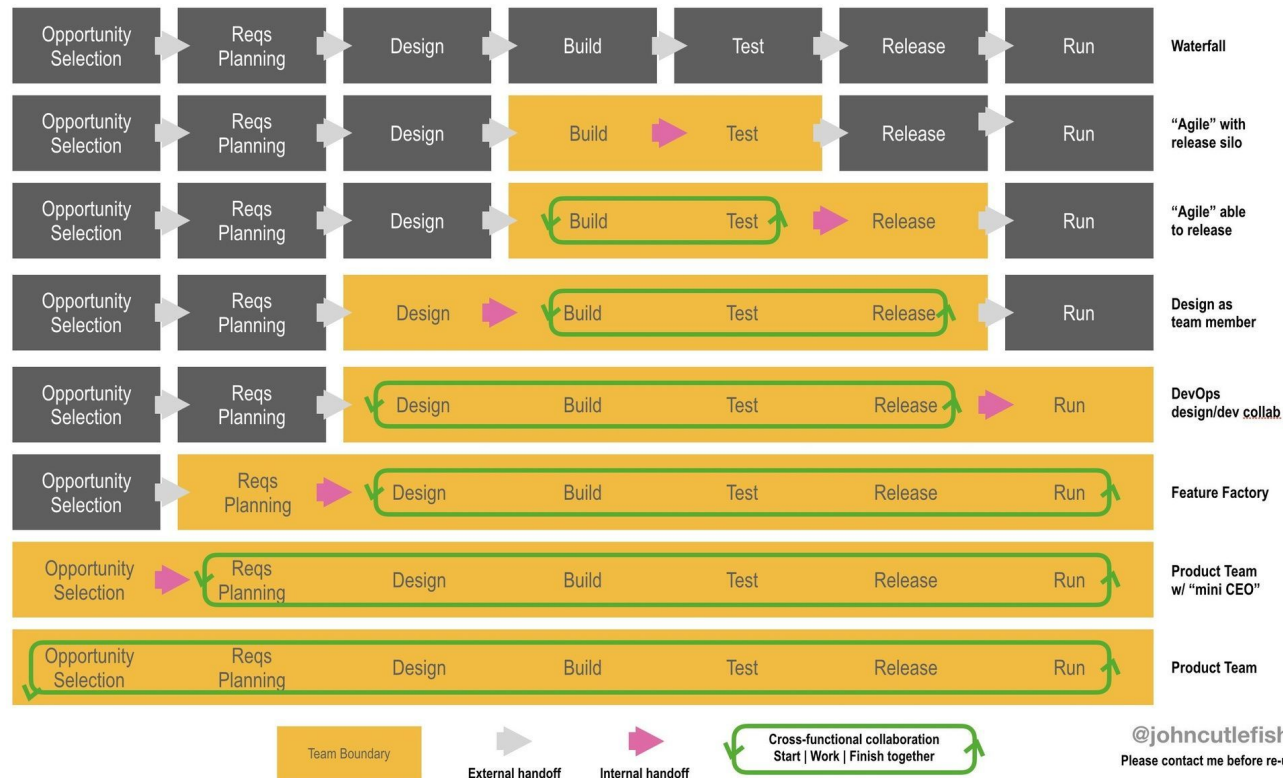
Responsible for usability risk + Experience

Insourced Engineers

Responsible for feasibility risk + Delivery

**PO Is a Role in Scrum,
Not a Job Description!**

Intentionally Designed Boundaries



Smallest Unit of Planning for Outcomes

Objectives

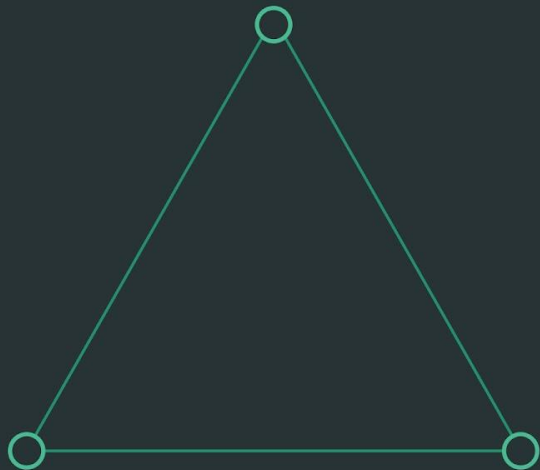


Discovery (Problems + Solutions)

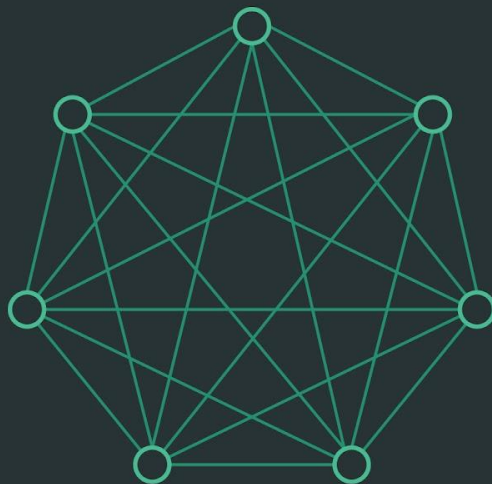


Delivery

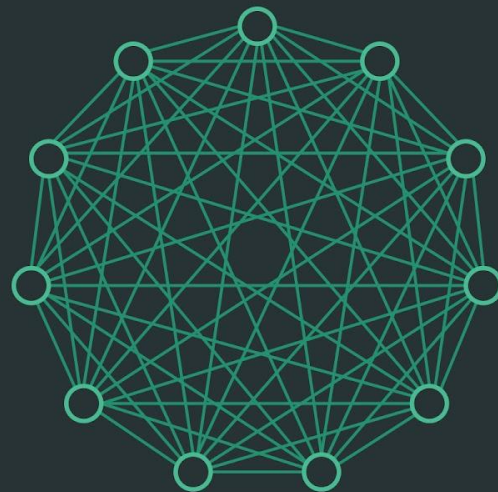
Brooks's Law: Communication Scales Exponentially



3 PEOPLE
3 lines



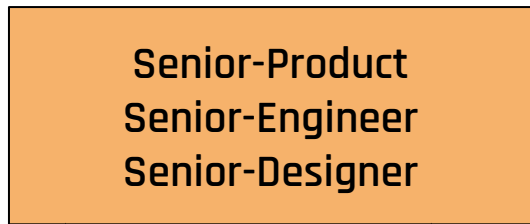
7 PEOPLE
21 lines



11 PEOPLE
55 lines

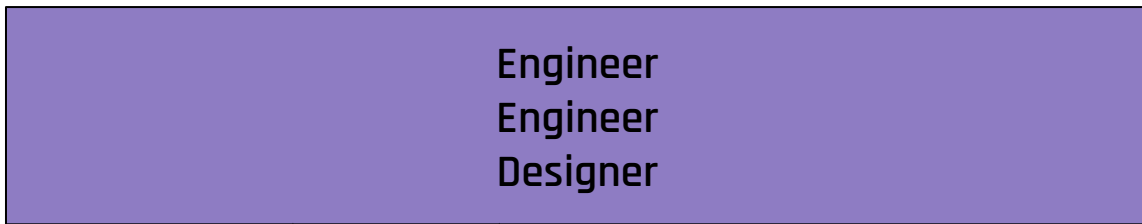
Optimize for Small Teams in Long Cycles

De-Risking: 2-3



2w

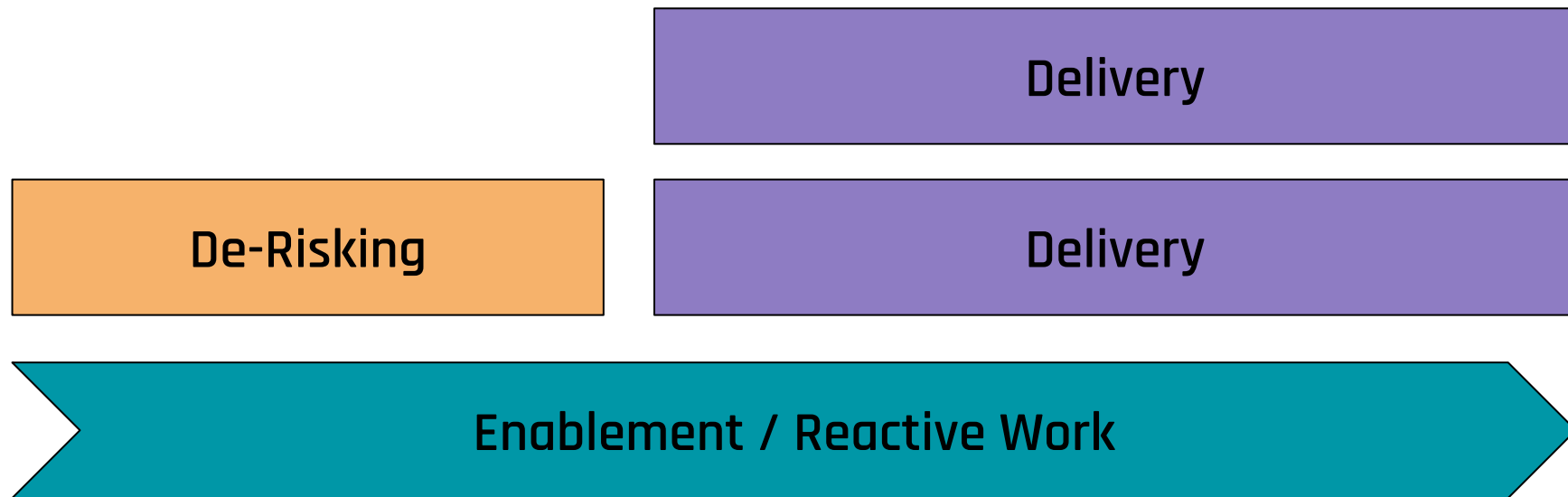
Solution Delivery: 2-3



6w

Trade-offs all the time!

Separate Enablement From Strategic Allocation



Happy WIP Limits everywhere!



Startups: Small Teams in Long Cycles



Emancipating People

Don't empower, emancipate!



Commitments Come From the Team, Not the Manager



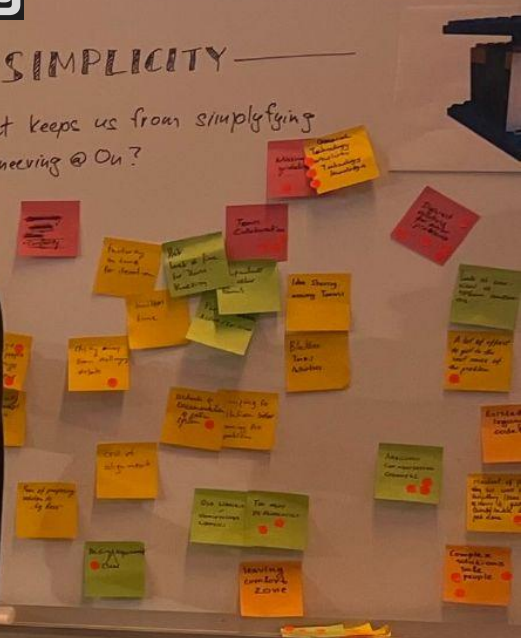
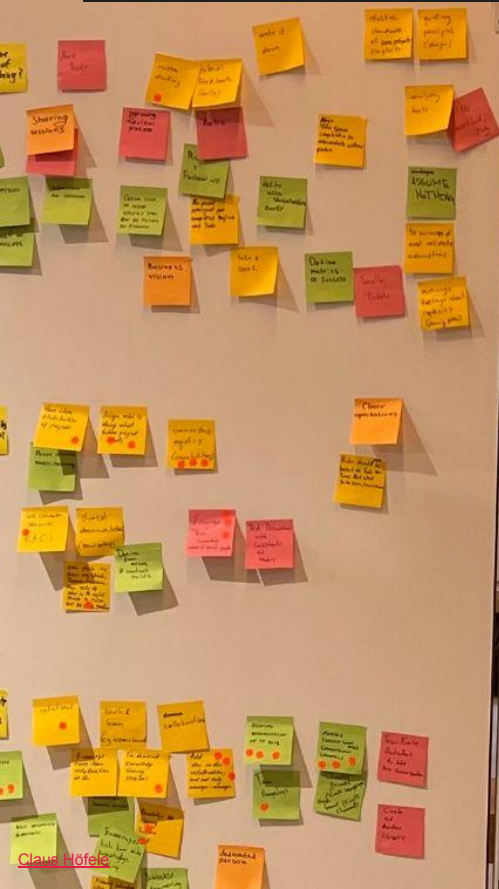
Don't Bring Answers – Facilitate Shared Learning

What?	Who?	When?
Hourup Guild docs		
Publisc Guide	Villanovs	Friday Feb
Neige Team docs Miro	Hannes & Claus	Friday Feb 23
Coffee Chat	Unut	Friday Feb 23

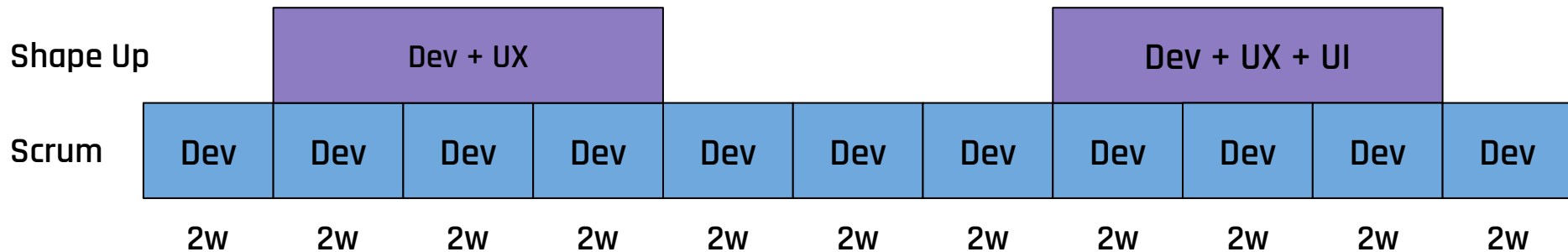
—SIMPLICITY—

What keeps us from simplifying engineering @ On?

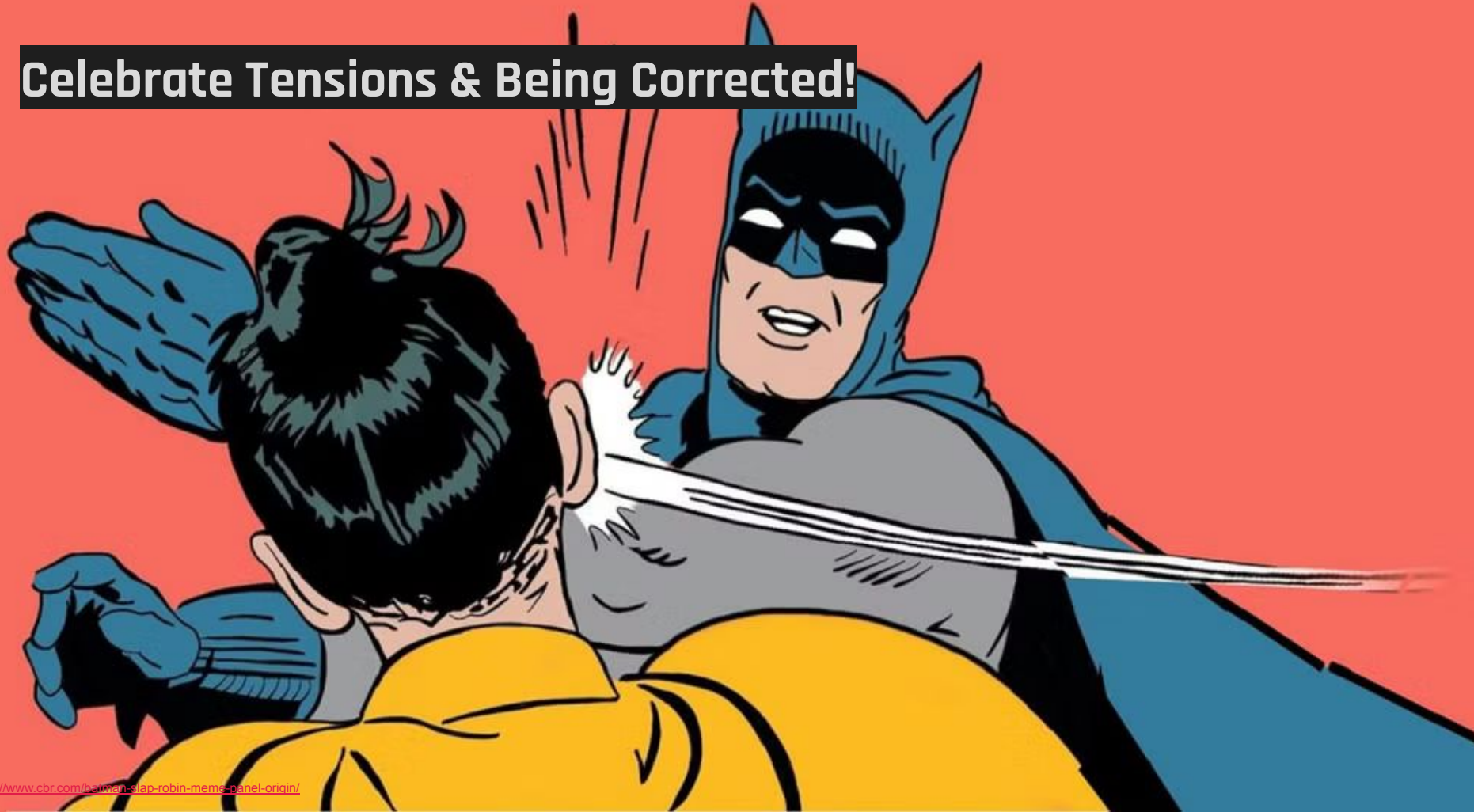
Whiteboard



Try: De-Risk Changes & Experiment



Celebrate Tensions & Being Corrected!



Try: Identify & Enforce Memes!



Break Silos, Not People!



Optimize for Decision Making.

Limit People Involved.

A portrait of a man with a beard and blue eyes, wearing a black hoodie, standing outdoors in front of a blurred city background. The text 'KLAUS BREYER' is overlaid on the left side of the image.

KLAUS BREYER

v01.io



EOF

Stream Aligned Teams

4 fundamental topologies



Stream-aligned team



Enabling team



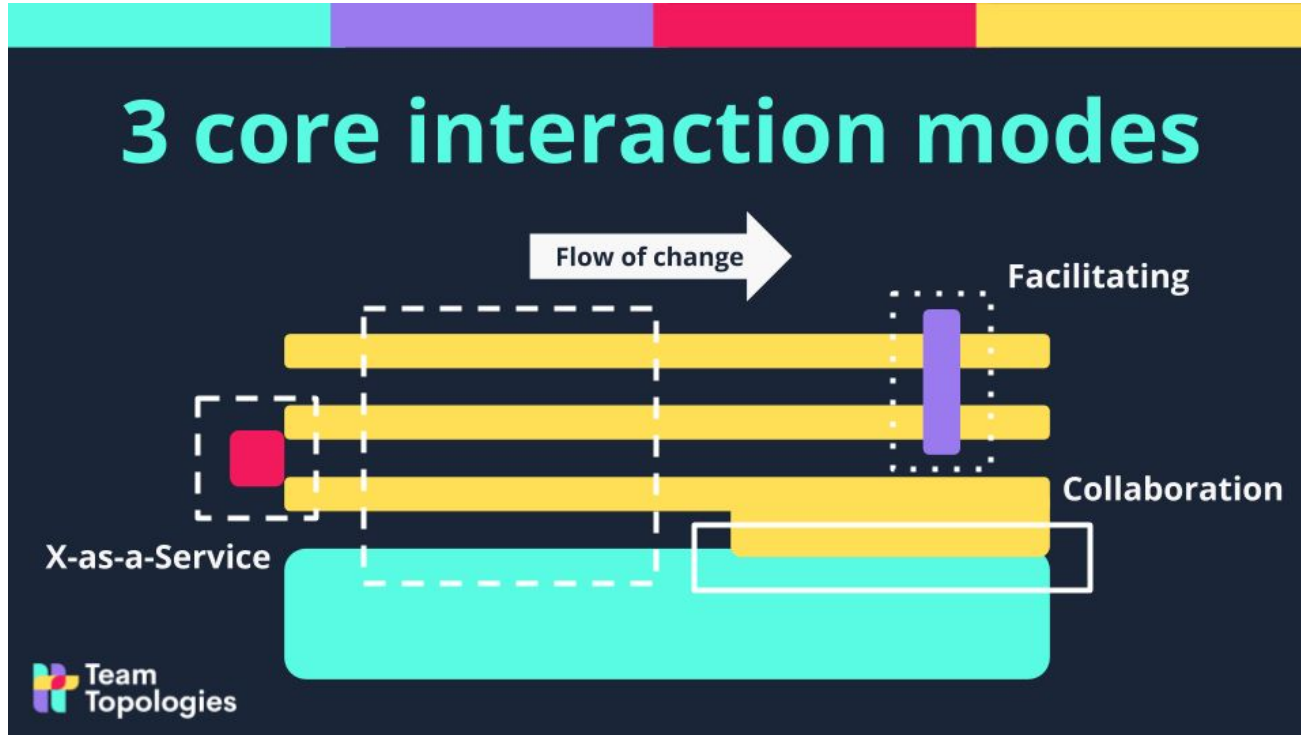
Complicated Subsystem team



Platform team



Stream Aligned Teams



Feature Team

- Serve the business
- Deliver features
- Not involved in product discovery
- Not responsible for impact
- “Mercenaries”

Product Team

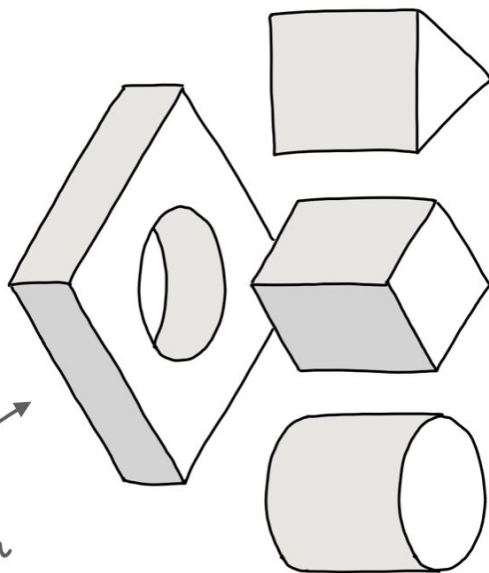
- Serve the customer
- Solve Problems
- Involved in product discovery
- Empowered and responsible for impact
- “Missionaries”

Slicing Problems: Context & Outcome

Framing

- What problem are we trying to solve?
- What slice of the problem are we addressing now?
- What will change when we're done and how will we know?

Narrowed-down
problem



Shaping

- What technical and design options do we have?
- What decisions need to be made before we can start the project?
- What are all the moving parts we need to consider?

Multiple
paths

Outcome > Output

Problem A

Problem B

1. Improving a feature
2. Getting more people to use it (increase % adoption).
3. Getting people to use it more (increase frequency).
4. New feature/product to support a new workflow (increase customers)

The AI UX Gap

WHO ARE WE?



CHATBOTS!!!



WHAT DO WE DO?



Sorry, I don't understand what you are trying to say.

Hello

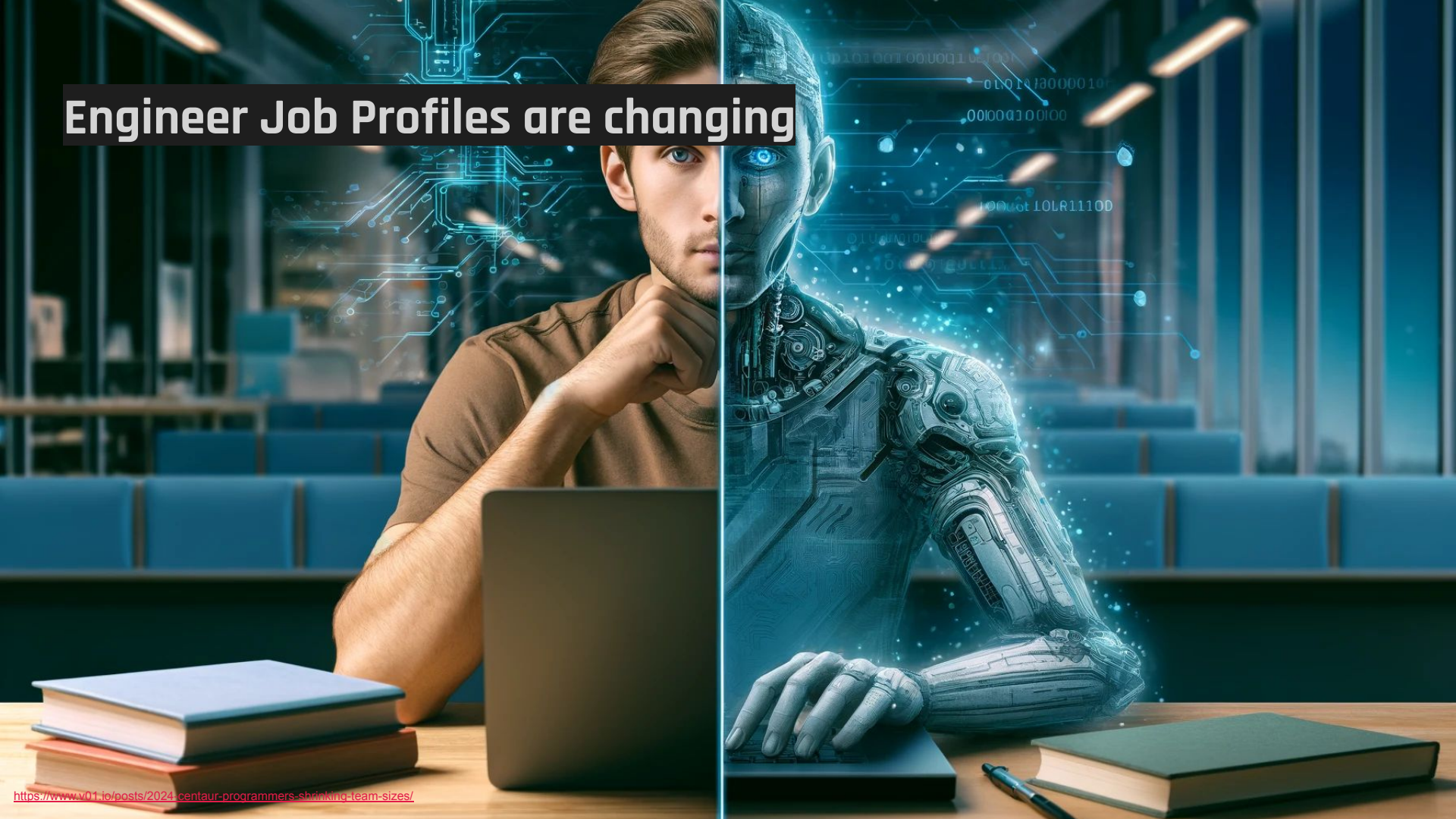
Sorry, I don't understand what you are trying to say.



Never wait for final requirements

(there aren't any!)

Engineer Job Profiles are changing



Enablement by oncall rotation

Product
Dev
Design

Shaping

Building

Dev
Design



30-40 Figma screens are a sign for silos



Principles > Process

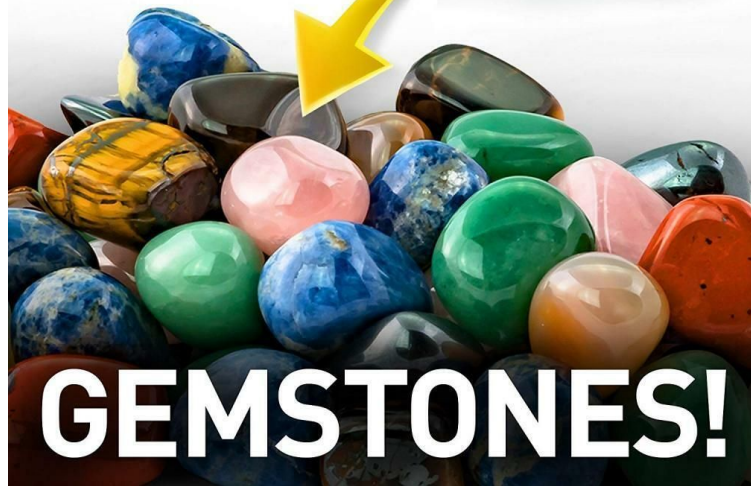
Join Discovery Conversations



Tensions are a good thing!



**TURN
ROCKS
INTO...**



GEMSTONES!

Objectives are outcomes!

Fat marker sketches as inclusive methodology

1



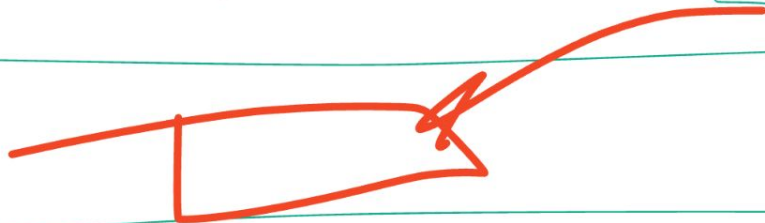
2



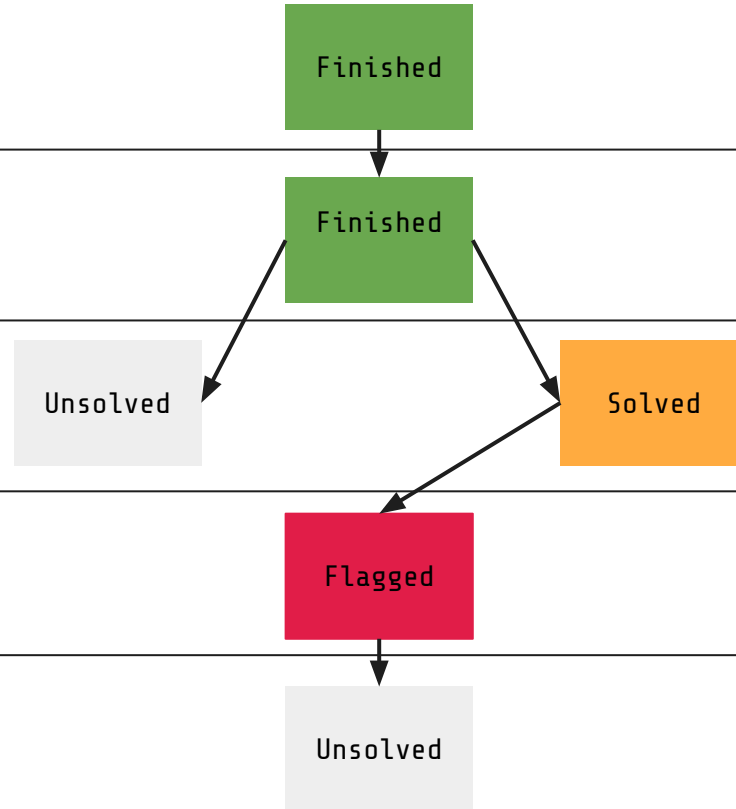
3



4



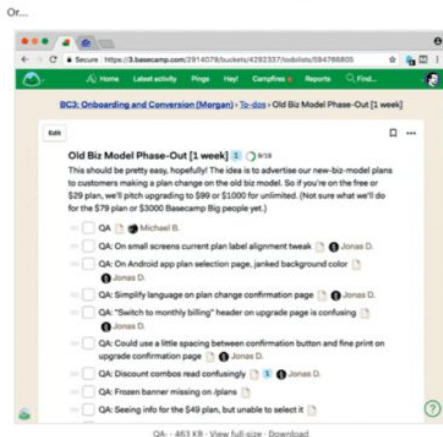
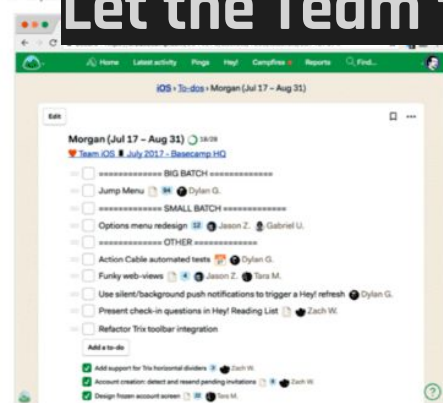
Identify, prioritize & enforce scopes



To-Do Groups

Jason Fried
Sep 12, 2017 · Notified 3 people

For 13 years



What we're trying to do is create simple groups of to-dos within a single list. So we're hacking it. We're either creating "=====Artificial Dividers=====" or prepending a batch of to-dos with a label like "QA" or our old standby, the trusty - which is code for "maybe".

groups first class citizens without adding noticeable complexity to the purity of today's to-do lists. We're pleased with where we landed, and we'd like to put it forward as a big batch project this cycle.

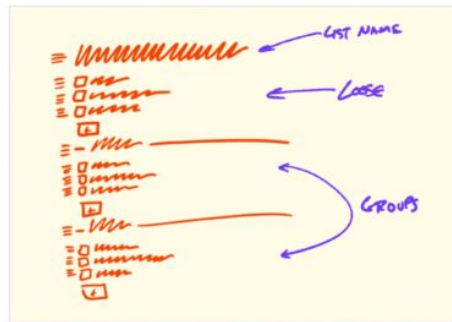
Here's how it works:

First, to-dos without groups will look identical to today's to-dos. We won't add any additional UI around them, no new buttons, etc...



A basic list remains the same · 185 KB · View full size · Download

But, if you add a group (or groups), it'll look like this...



A list with some loose to-dos plus two groups of to-dos · 230 KB · View full size · Download

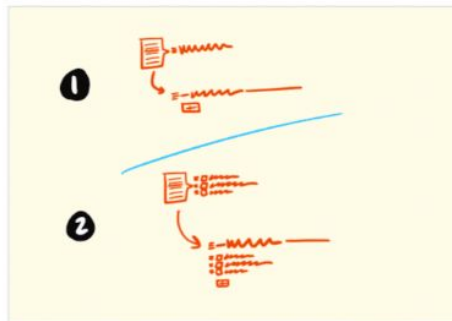
This means now you could make a list for a specific scope that had a section for design-specific to-dos, and QA-specific to-dos.



With labels · 189 KB · View full size · Download

A group is defined by a divider. You can name a divider anything you want. It can't be checked off, it's not a to-do itself. Any to-dos you put below the divider become part of that group. And critically, groups always go below any ungrouped (loose) to-dos. This simplifies a bunch of behaviors, and prevents loose to-dos from being lost between groups. If it's loose, it's at the top.

You can create a divider in two ways...



Two ways to make a divider · 152 KB · View full size · Download

1. Click the hamburger menu next to the list title. You'll get a menu. One of the items will be "Create a group" (or add a divider, exact language TBD). It'll then add a blank divider at the end of the loose to-dos, and scroll right to that point so you can name it.
2. You can multi-select (shift key) multiple loose to-dos, and instantly create a group from those to-dos. If you do this, an unnamed divider will be created, and the to-dos will be placed under that divider.

Key to this concept is that we'll add "Add a to-do" buttons to the bottom of each list (and the loose to-dos at the top). So once a divider exists, you can add to-dos to that section. If we didn't do this, and we only had one add to-do button at the bottom, you'd have to add and then move into place. That's a hassle. It's much better to add it place when we have places to add.

Completed to-dos will still be grouped into a single collection at the bottom of the list. If they were part of a group when you checked them off, we'll prep the group name before the to-do like so...



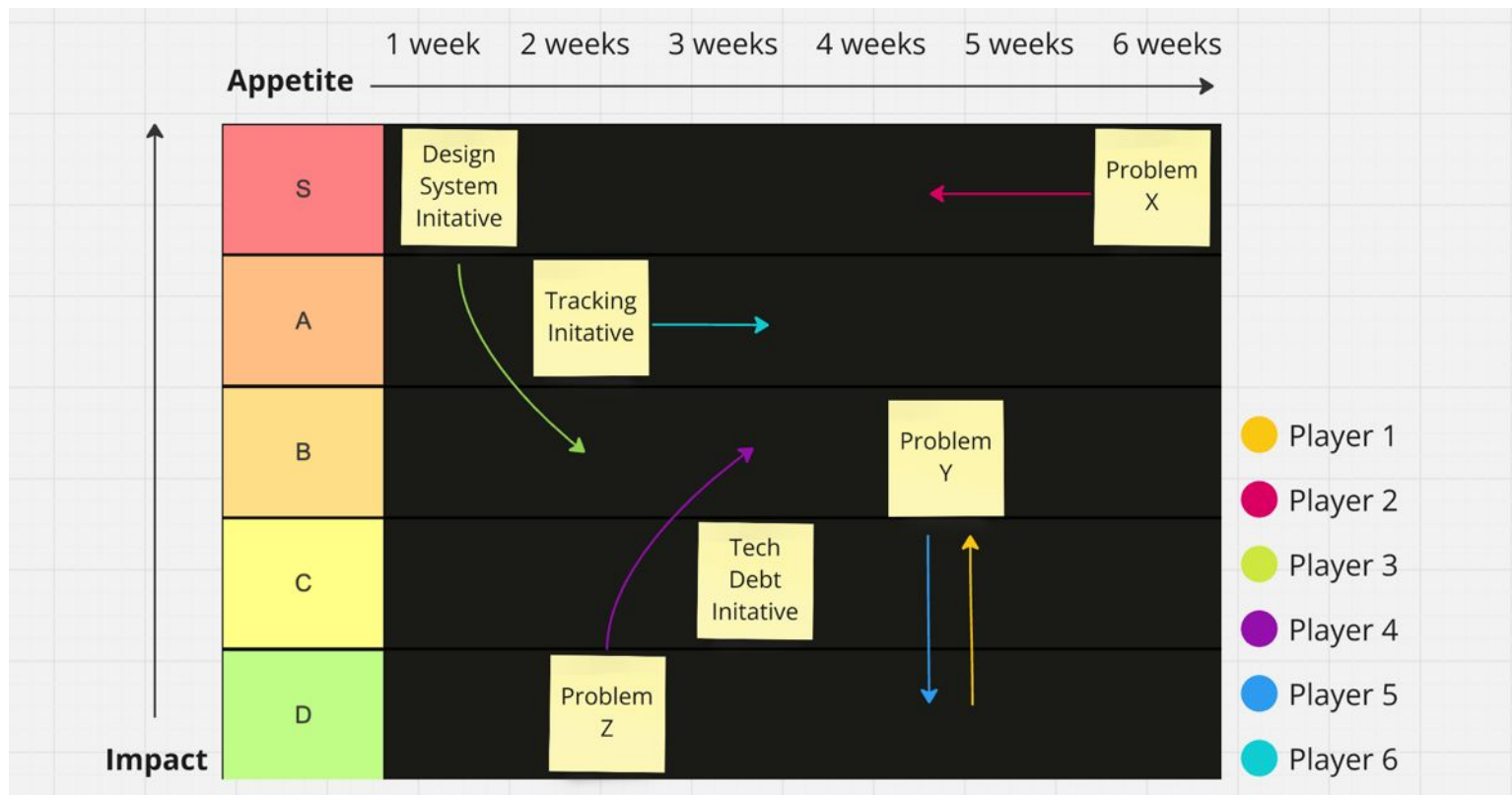
If you uncheck a completed to-do from a group, it could go back up to the group (assuming the group hasn't been deleted). · 185 KB · View full size · Download

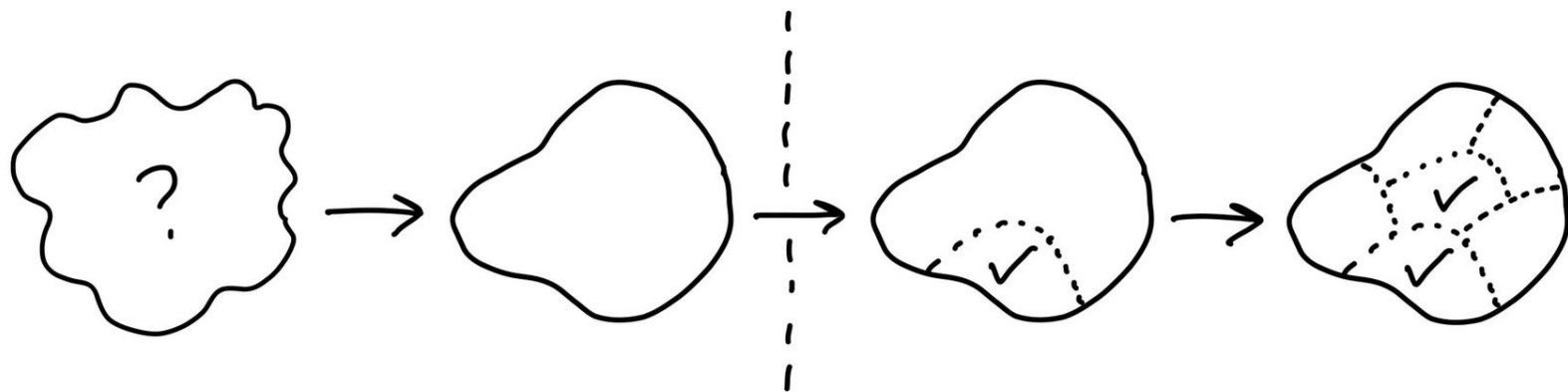
Groups will also have their own perms. So you can click the title of a group/divider and you'll see a new page with just the to-dos in that group...



Group perms for QA · 195 KB · View full size · Download

Let the whole team align on their priorities





Shape Up v 1.8, 2019 edition

Stop Running in Circles and
Ship Work that Matters

by Ryan Singer

**Fixed Time,
Variable Scope.**

**Shape Up in
a Nutshell.**



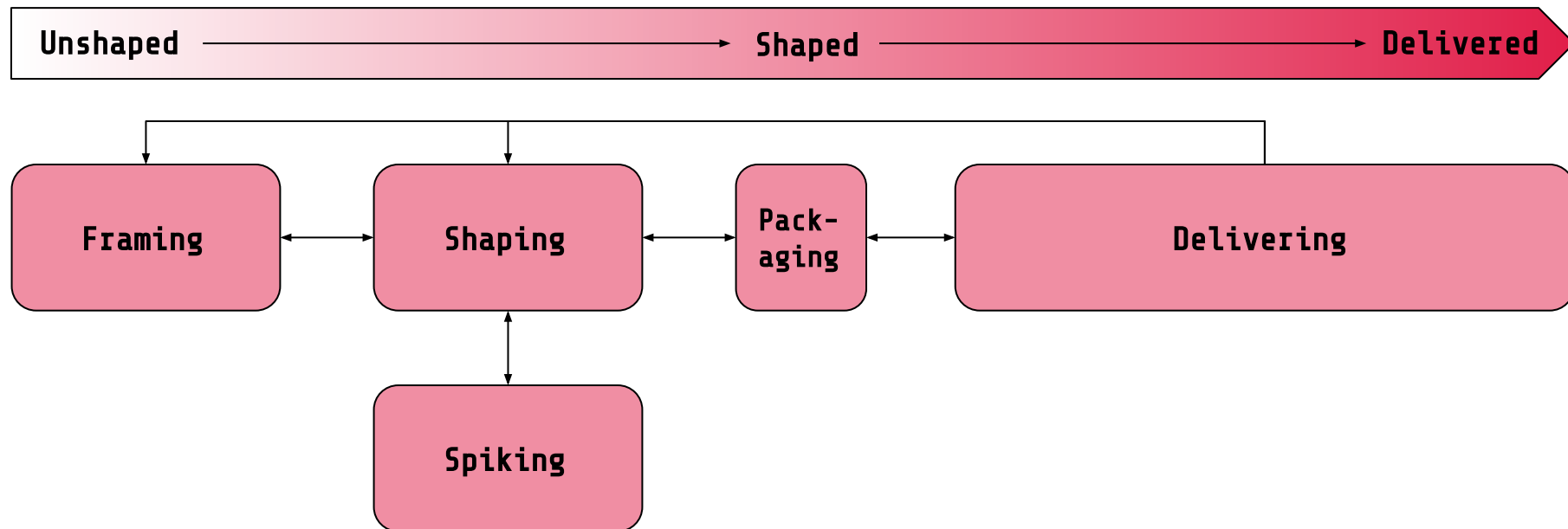
Organize by structure, not by person.

Start at the Epicenter

**Only real work
uncovers real needs.**



Pursuing constant de-risking



Framing

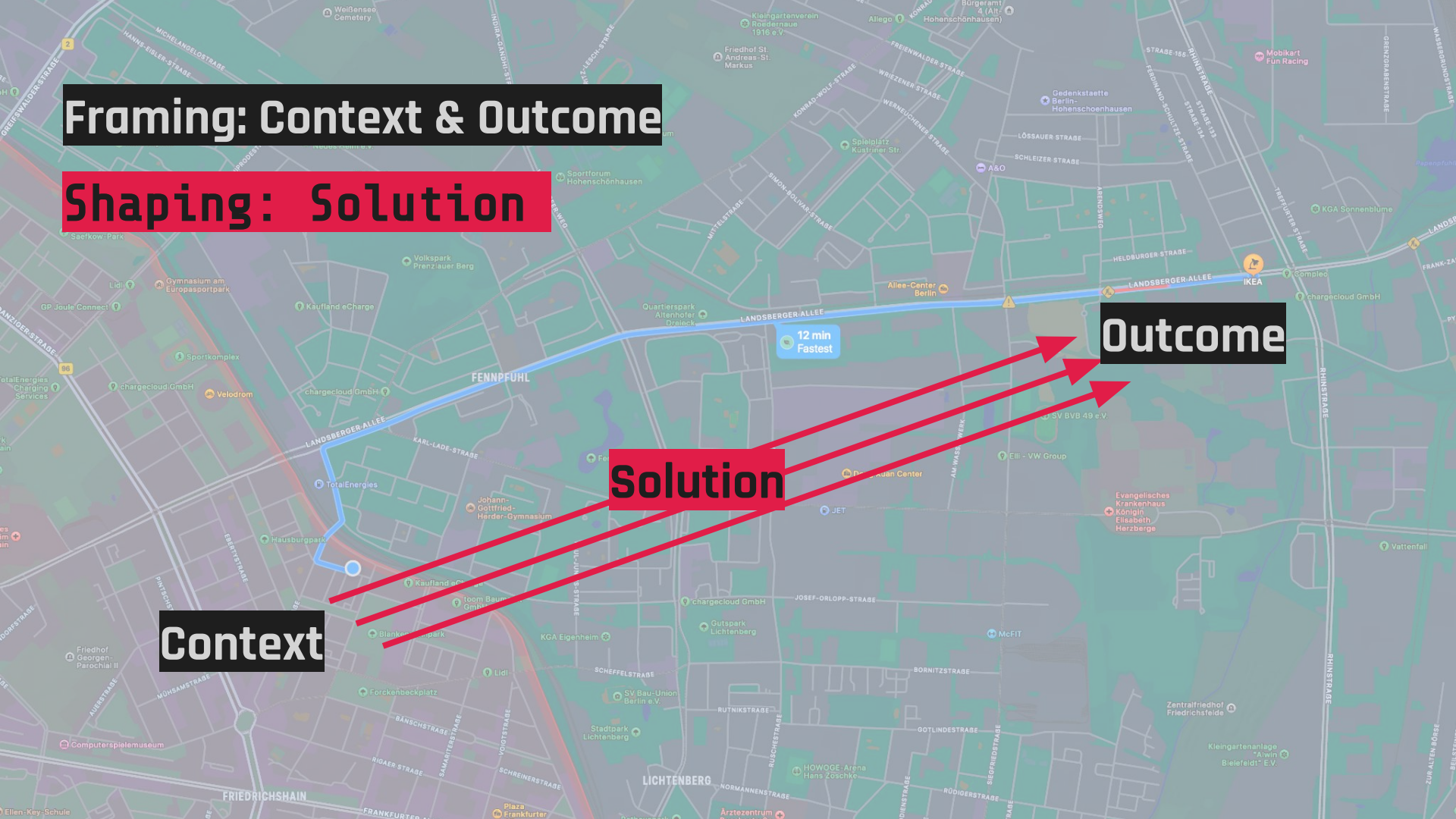
Framing: Context & Outcome

Shaping: Solution

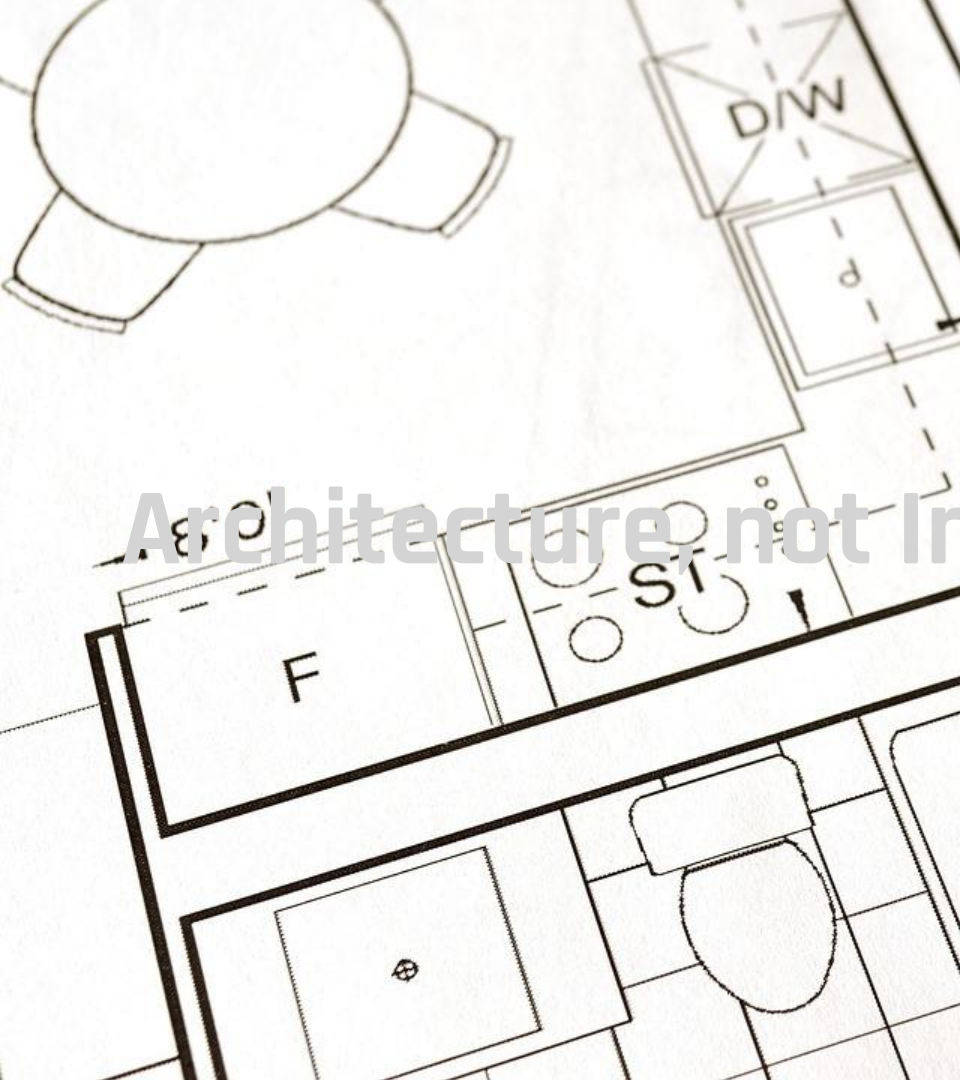
Outcome

Solution

Context



Shaping



Architecture, not Interior!



Playing Field

Frame

Context

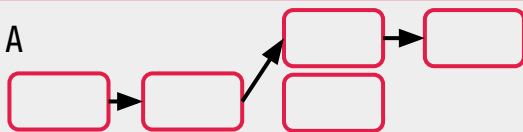
Outcome

Fit

	A	B	C
Requirement 1..	✓	✓	
Requirement 2..		✓	✓

Paths

A



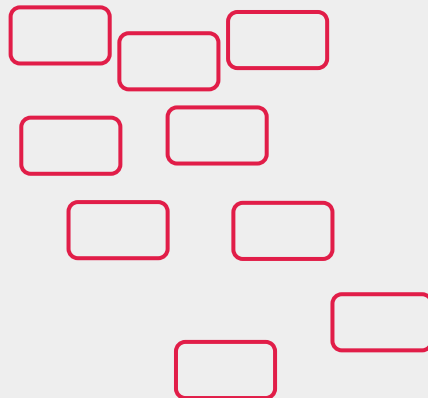
B



C



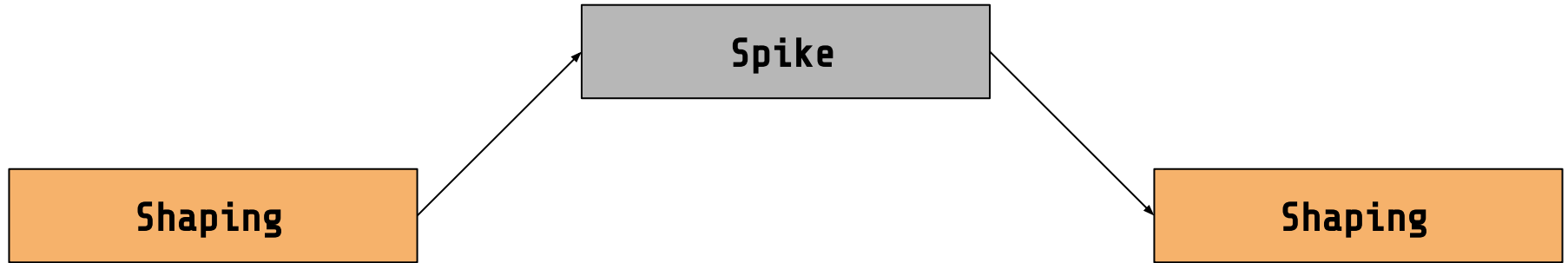
Parts



"Find the best path that fits the frame and won't blow up in delivery."

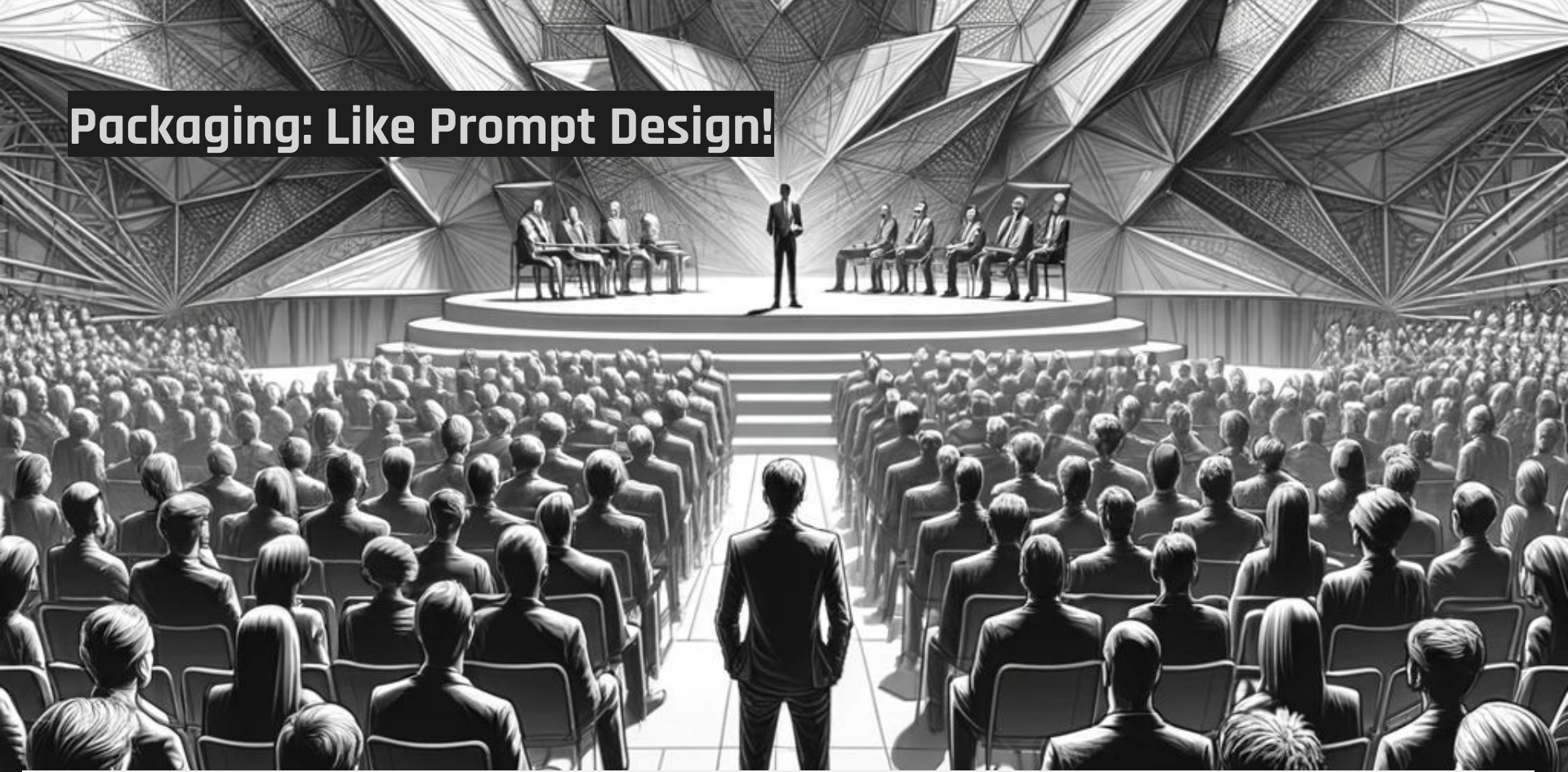
Spiking

Spiking



Packaging

Packaging: Like Prompt Design!



“draw me a painting of mc escher, grey, where a male person stand in front of a conference crowd - with geometric architecture structures”

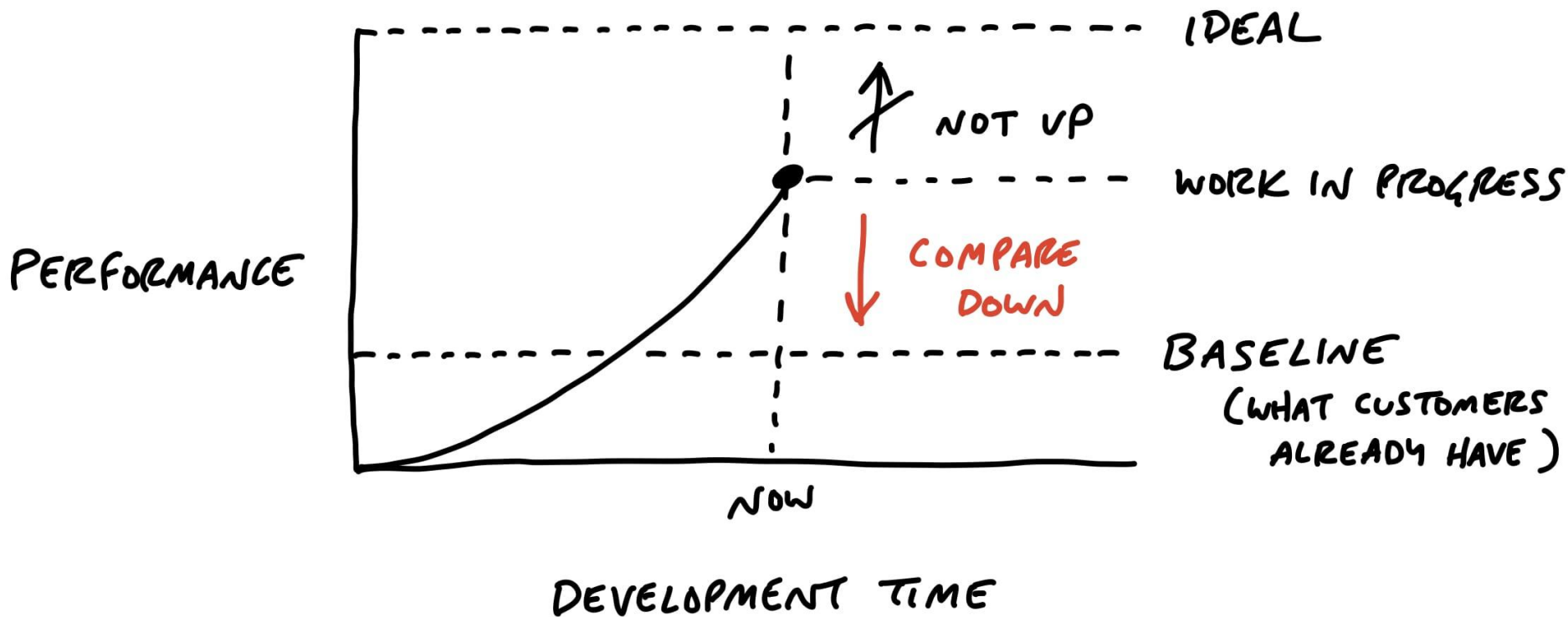
Building

Autonomous Teams!

Thinking is
like REM Sleep.



Always compare to baseline



Tooling

Todo-Tools are made to account for work, not figure out what the work is.

Time Left

Objectives

Accepted

14 days

27%

280/367

Epics

17%

Features

56%

Stories

76%

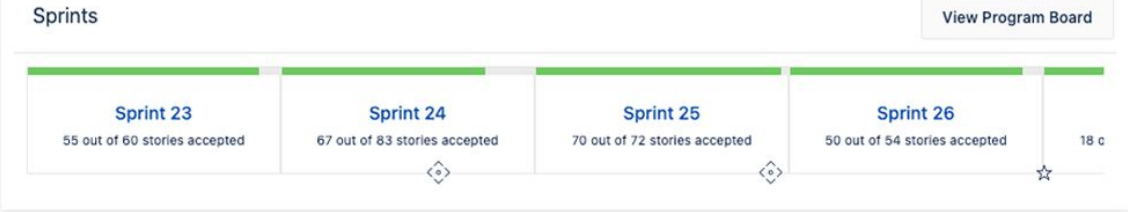
Dependencies

63/73

View report

Team Progress

	Points	Count
Cowboys	78.1%	
Washington	90.2%	
Baltimore	86%	
Houston	62%	
Tiger	88.8%	



Feature

Search by ID, Nar

PrioritizeQuick FiltersPlanning Interval BurnUp

ID	Title	Feature Progress	Points	MMF	Status	Owner
5426	Implement Live Call Monitoring	<div></div>	12	No	IN PROGRESS	MD
4273	G12 : Team innovation for AC5	<div></div>	12	No	READY TO START	
4209	Chatbot JSON integrations	<div></div>	1	No	IN PROGRESS	NO
3422	Intelligence Assisted Chatbot	<div></div>	24	No	IN PROGRESS	
5386	G12 : AP2 machines stabilization	<div></div>	1	Yes	ACCEPTED	
4207	G12: Automation	<div></div>	1	No	ACCEPTED	
5387	G12: ESRs for AC5	<div></div>	24	No	ACCEPTED	

Dependencies

Cha

G12: DB c...

Delivered

Cha

Mirrored ...

Delivered

G12: Hadoo...

Delivered

V2.1 SAT - ...

Delivered

GD

V2.1 SAT ...

Delivered

GD

V2.1 SAT ...

Delivered

Interface: P...

Delivered

Objectives:

G13 Family Feature for...

In Progress

Increase user retentio...

In Progress

Grow iOS Daily Active ...

In Progress

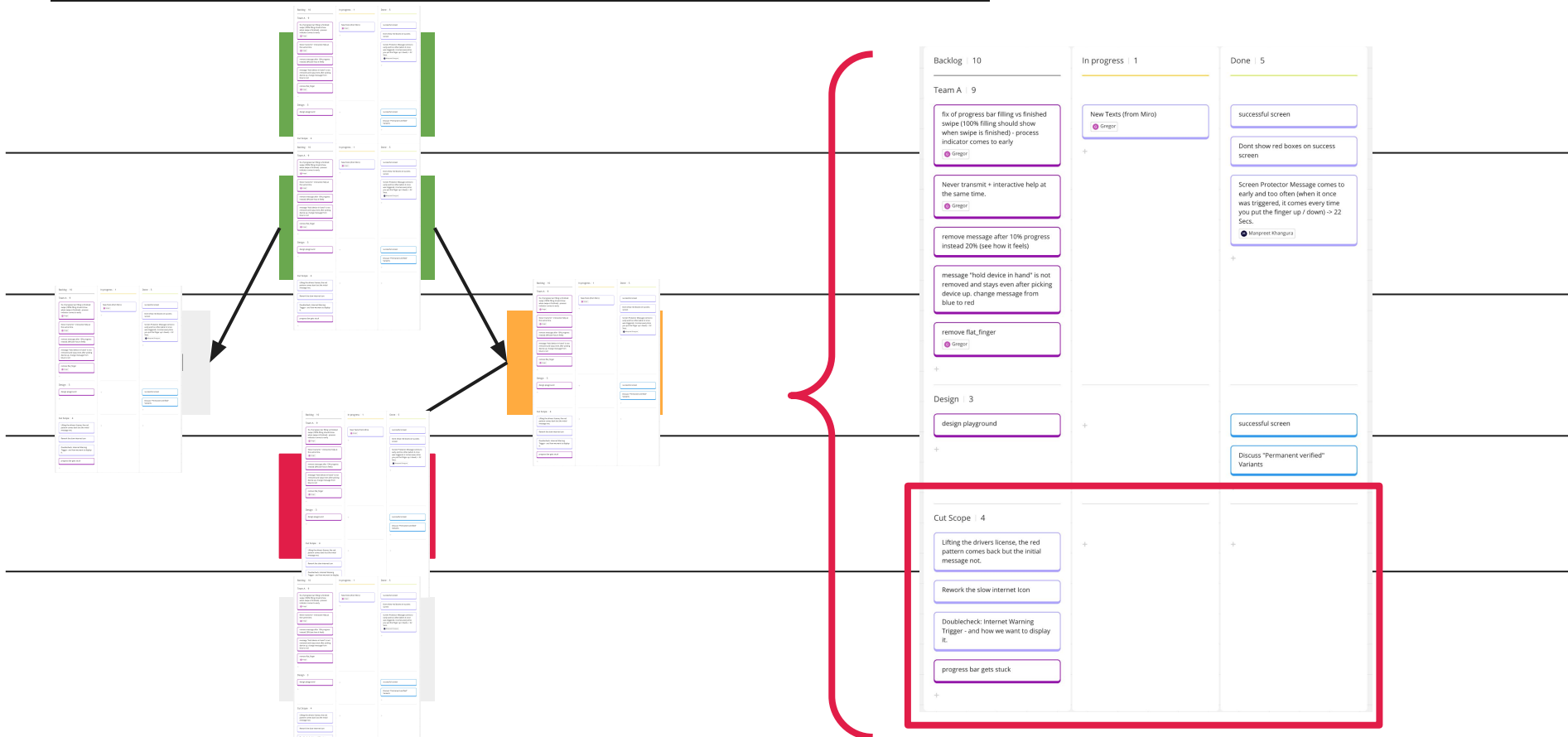
Design Survey for Call...

In Progress

Increase user retentio...

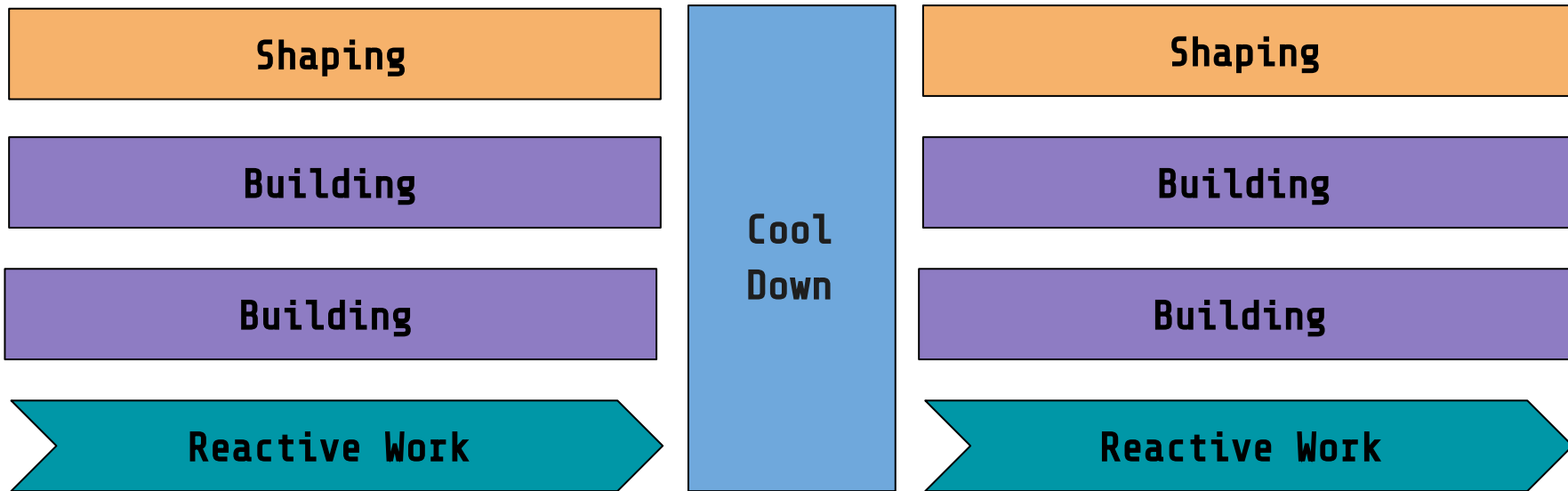
In Progress

Always cut scope - with appetite in mind!

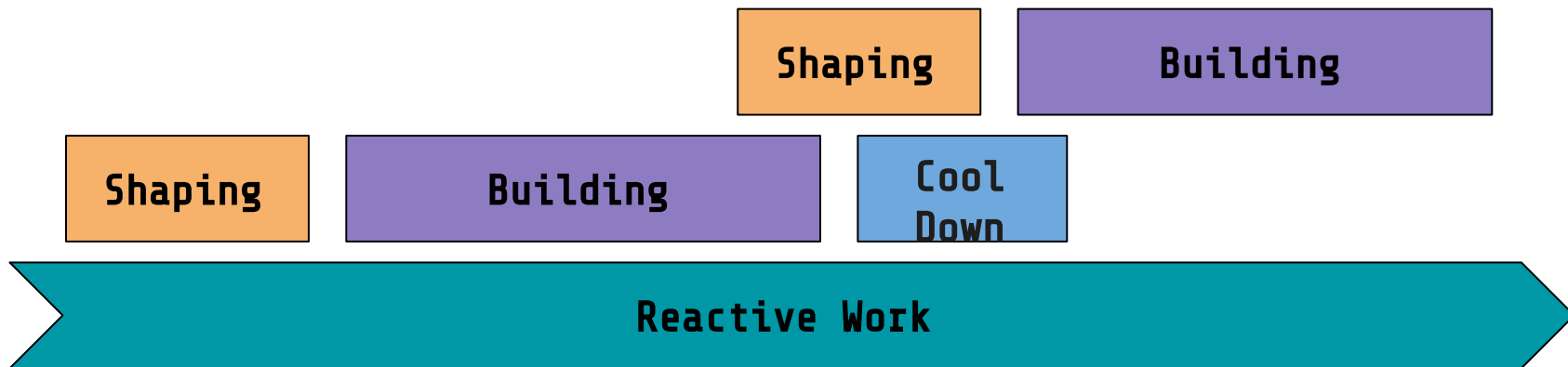


Implementation

Oceanliner Model (Synchronized Cycles)

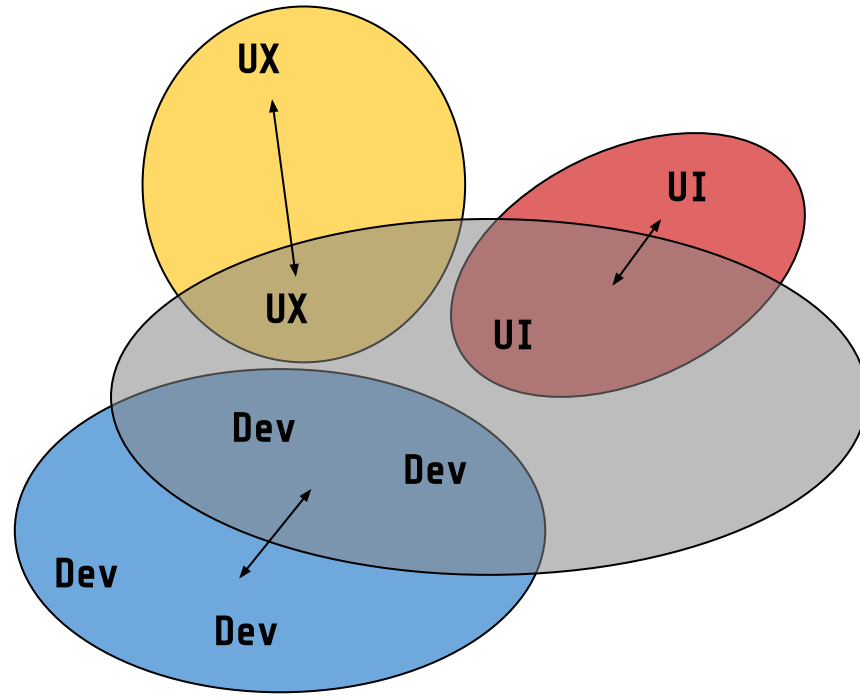


Speedboat Model (Asynchronous Cycles)



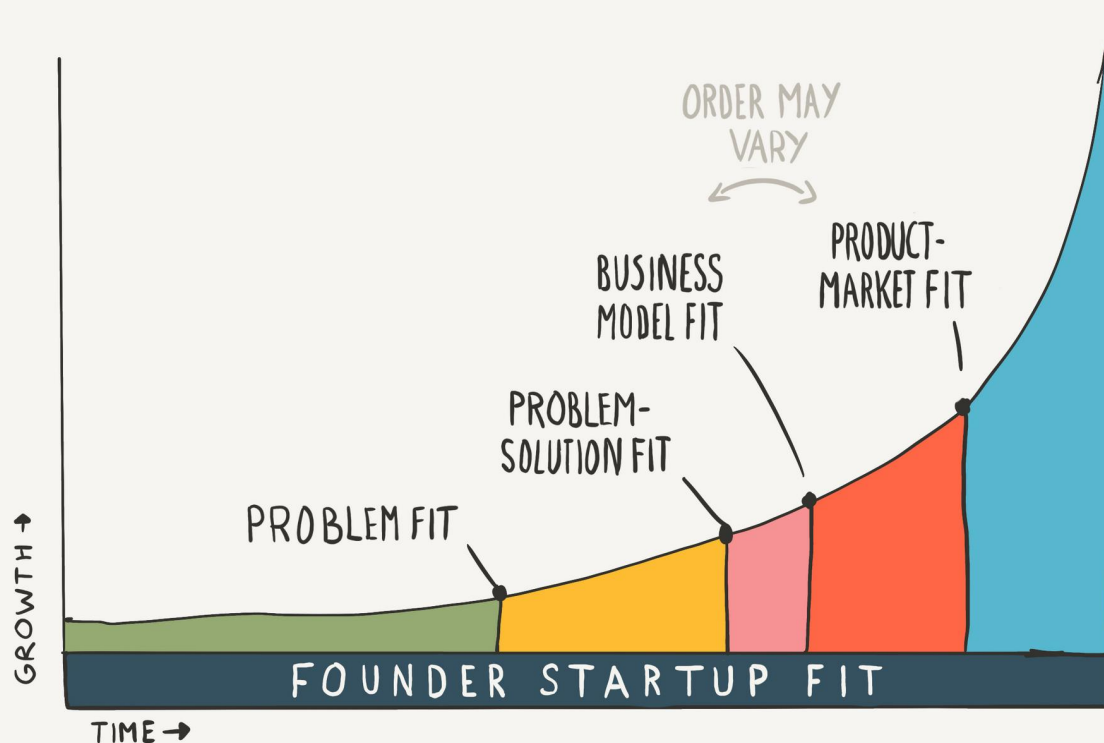


Builders as a “temporary squad”



Bigger Picture

Investing in Product (Teams): Scale with Proof



Pursue constant de-risking

Unshaped

Shaped

Delivered

Interviews

Framing

Shaping

Pack
-agi
ng

Delivering

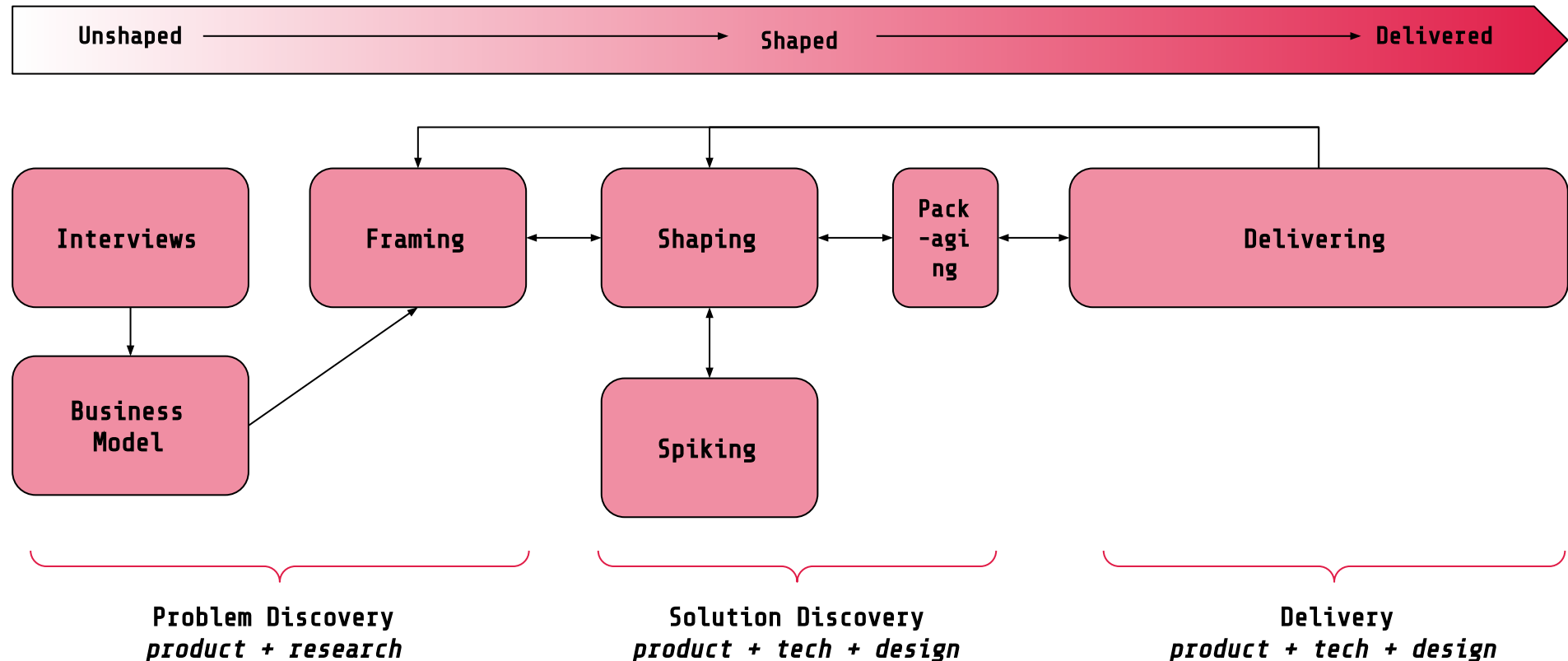
Business
Model

Spiking

Problem Discovery
product + research

Solution Discovery
product + tech + design

Delivery
product + tech + design



Reasoning

Real feedback from a real team

ui feedback did not
disturb, but felt
helpful.

feeling of
being closer
to each other
(XD - DEV) b

Awesome team!
The teamwork
feeling is back!

UX, UI, Dev, PO all
working
simultaneously on
one topic, being on
the same page and
in the right mindset

Everybody
contributing to find
the best solution -
not only UX

Fixed Time

Variable Scope

Autonomous Teams

Fight Details early on

Defuse Timebombs

Start at the Epicenter

Organize by structure

One scope at a time

Done means deployed

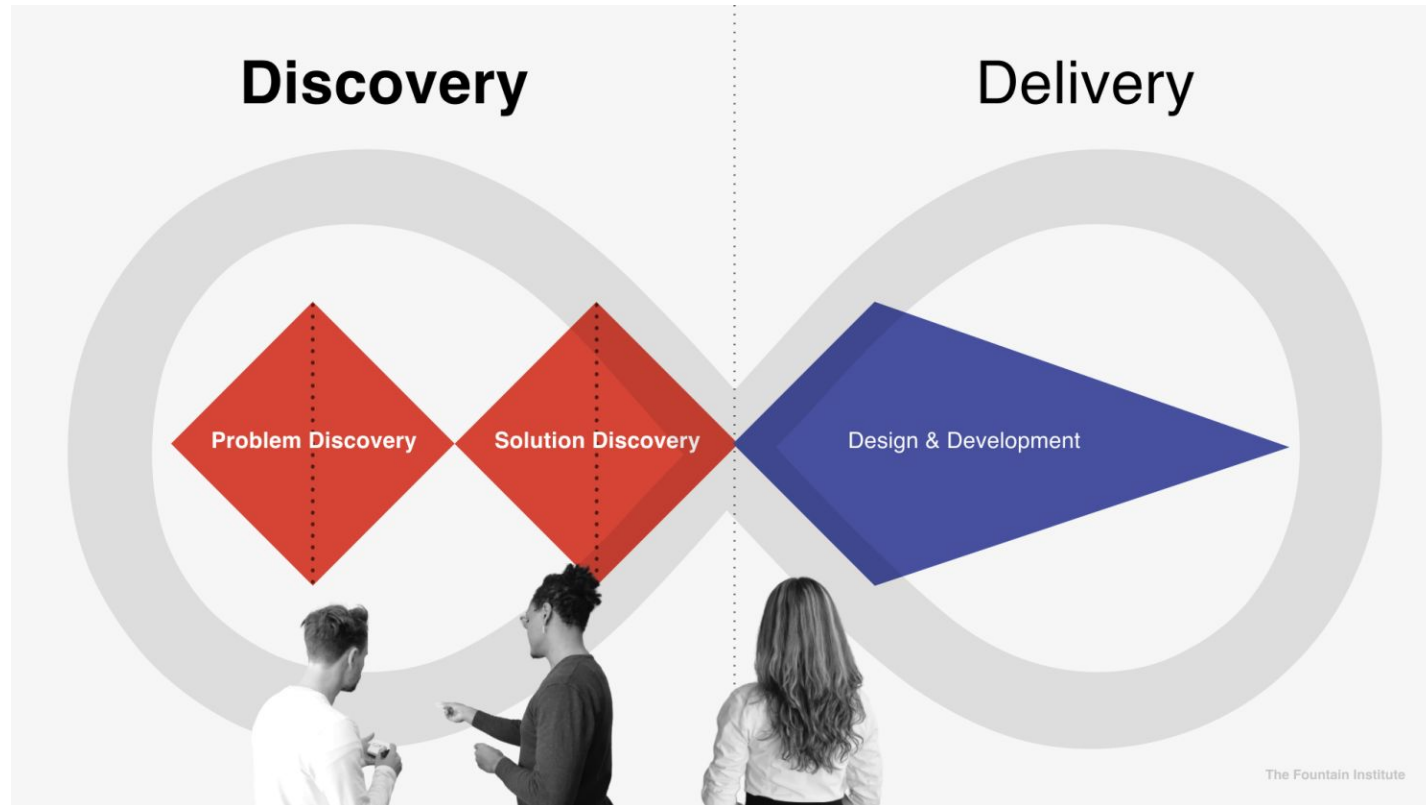
Requirements for Shape Up (by the book)

- Ongoing Product Development (Vision should be established)
- Empowered Product Team
- Team of a certain size
- Fixed staffing for an extended period of time
- Minimum: 2-3 devs, 1-2 Designer, 1 Product \geq 50% booked

(The Tools + Principles can be valuable, nonetheless)

Backup

Discovery > Delivery



When to use Shape Up?

Shape Up is for Features.

Reactive work needs separate, dedicated capacity.

Not for cross-party integrations with dependencies

Why use Shape Up?

To Scale: Run multiple parallel product teams.

When slowing down: Get focused and back to shipping fast.

Possible Experiments

One six-week experiment

Start with cycles

Start with shaping

How to build in small autonomous teams?

How to shape (derisk) collaboratively?

How to frame problems hypotheses?

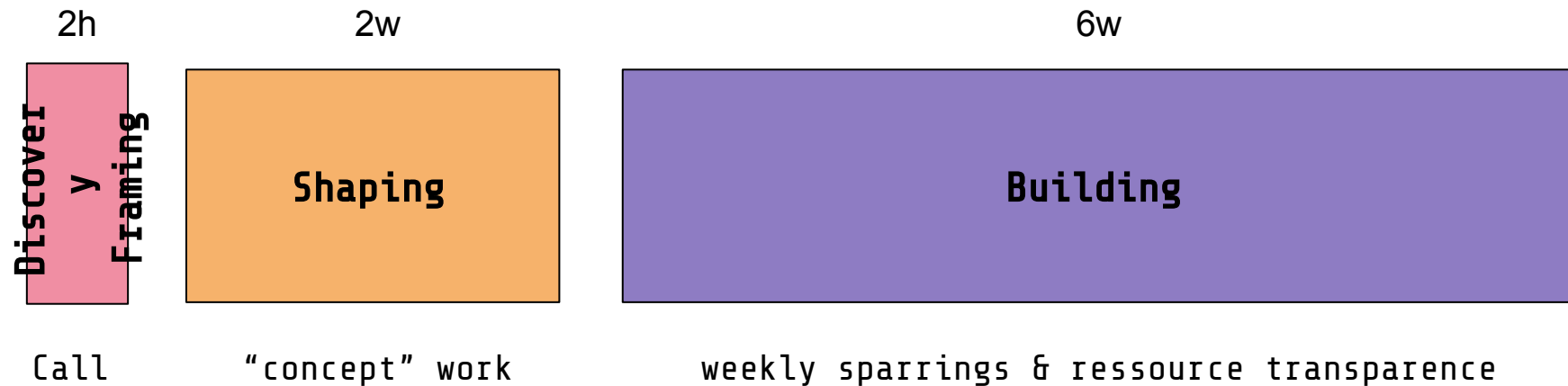
How to involve the client?

Starting Points

Shape Up is born remote.

Balancing synchronous and asynchronous work.

Agency Model: Bring the Customer Along!



Scrum Bashing

Martin Fowler



Bernd Schiffer

@berndschiffer



Shitty Agile For Enterprises -- @martinfowler about
what SAFe stands for #agileaus

[Tweet übersetzen](#)

8:57 vorm. · 17. Juni 2014 · Twitter for iPhone

Allen Holub



Allen Holub
@allenholub



Was asked what I don't like about Scrum. Not much, really. Just:

- Backlogs
- Sprints
- Daily Scrums
- Sprint Reviews
- POs
- SMs
- Commitments
- Accountabilities
- Standardization on an inflexible framework
- Certs

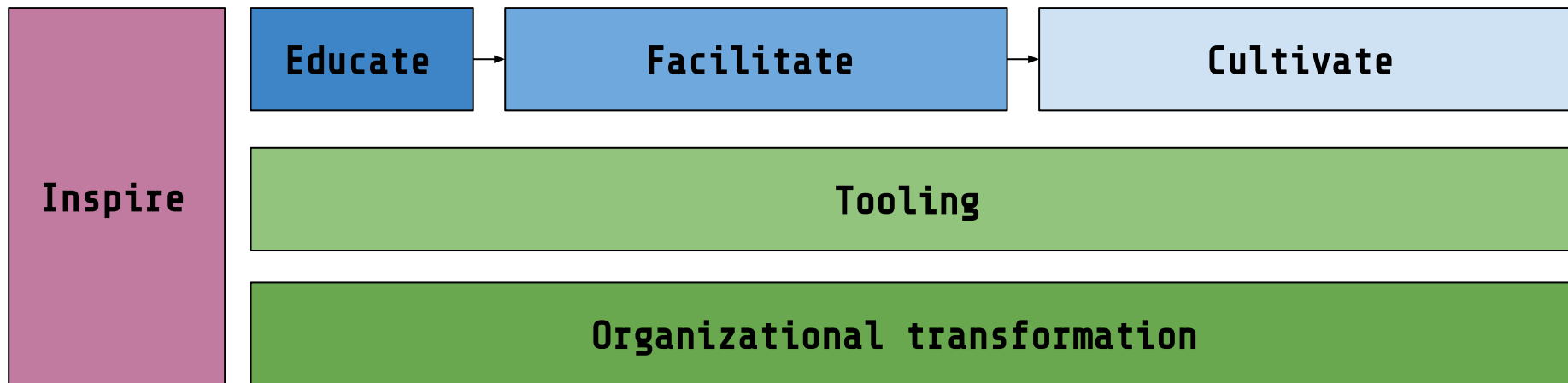
Just little stuff like that. The rest is okay.

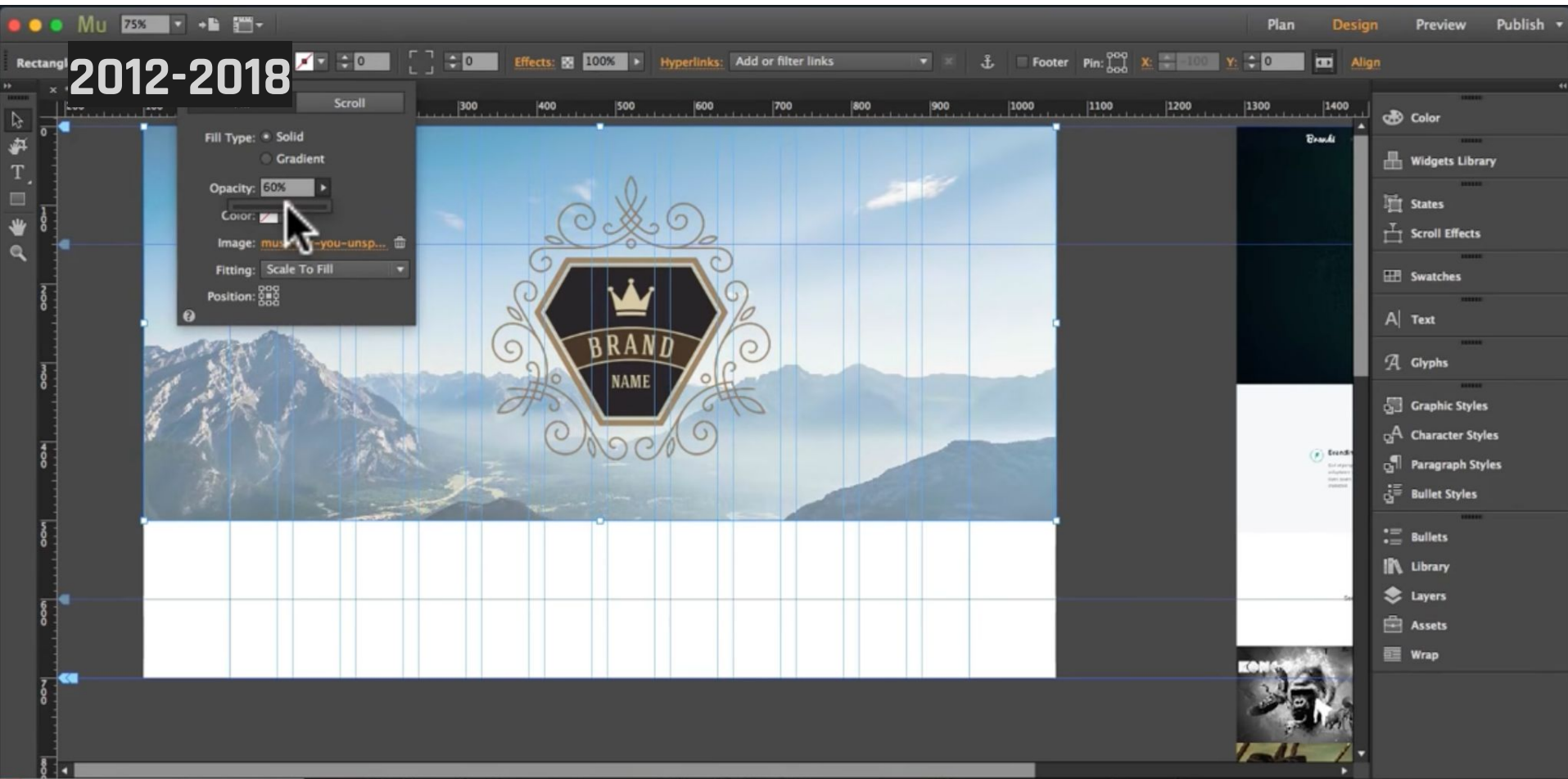
[Tweet übersetzen](#)

5:24 vorm. · 15. Apr. 2022 · chirr.app

My Offer

My offer for product orgs and agencies





Good people attract more good people.

The recipe: Engaging challenges with a dash of repetition.

Champion journeymanship

Balance fresh talent with experienced hands during each cycle.

Start gradually

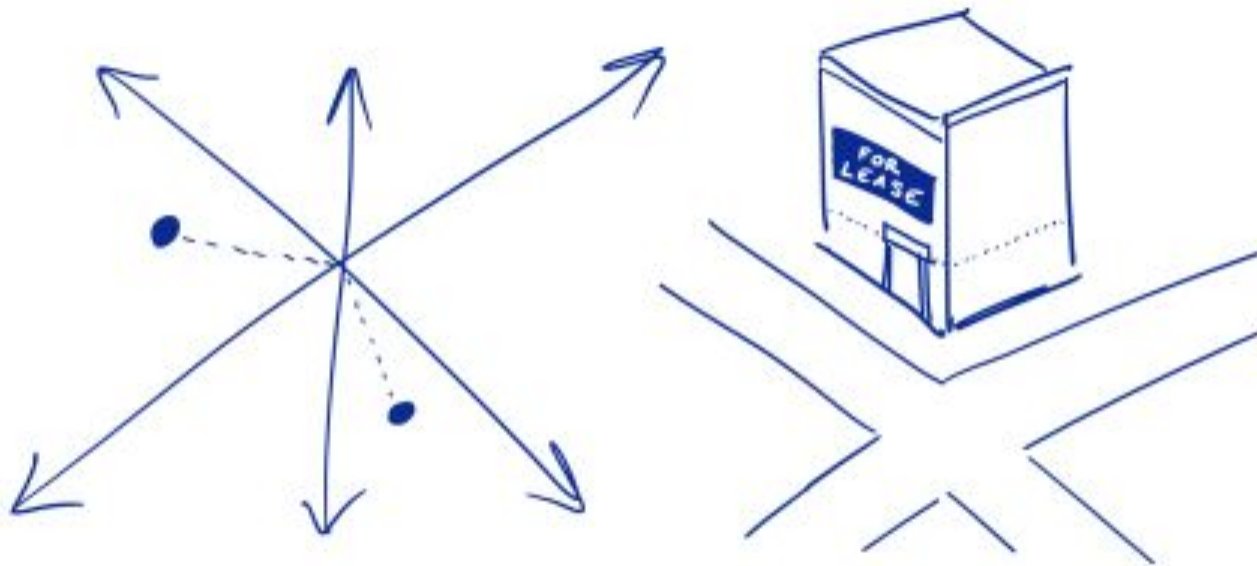
Start with a one time experiment.

Start with shaping.

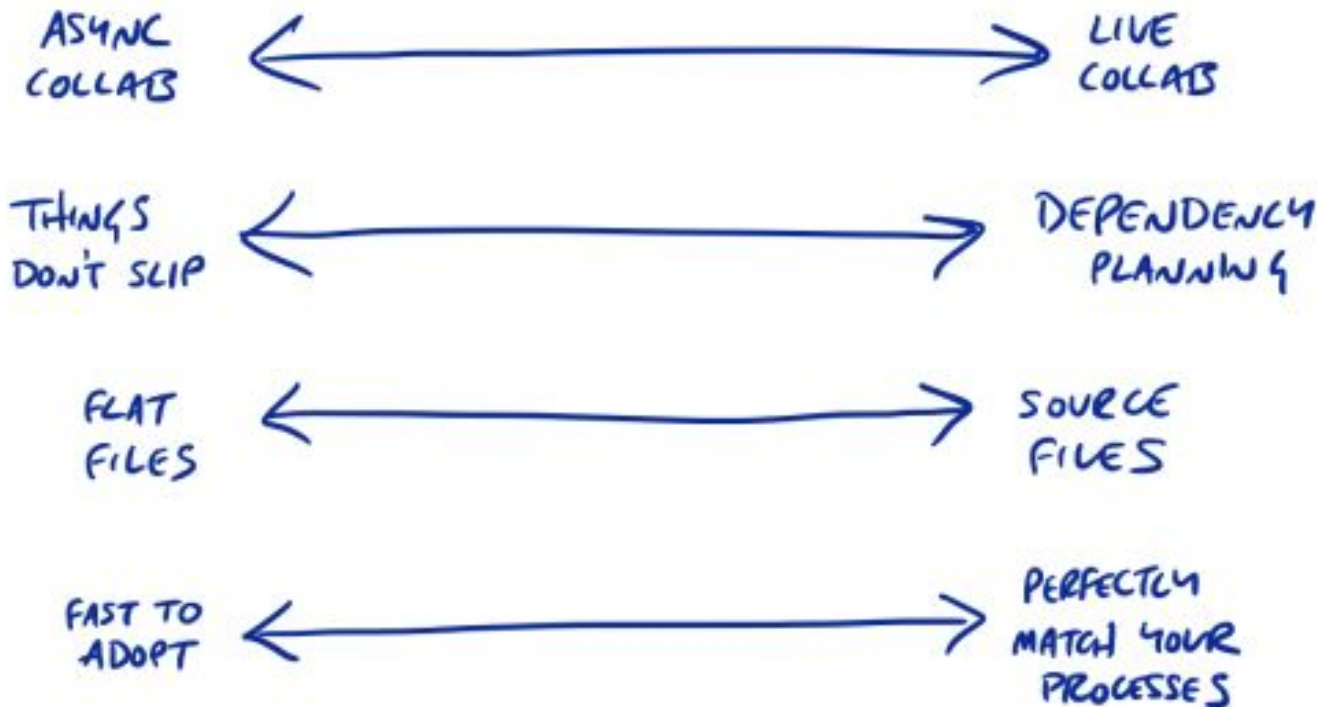
Start with cycles.

Strategy: Positioning

Position, Position, Position!



Position, Position, Position!



Position, Position, Position!

Less about	More about	Comment
Up to the minute	On your own time	Prefer async, bundled notifications, long form
Millimeter planning	Nothing gets forgotten	"Not dropping" something is more important than dependency management
Creating the work	Discussing the work	Write that copy yourself and post for feedback, instead of live collab with someone
Source files	Renderings	In progress PSDs go elsewhere. Code goes in GitHub. Share an intermediate piece for feedback, or upload the final version for posterity.
Embodying specific processes	Easy to adopt	You can make a better bug tracking process in Jira, more precise creative direction with InVision etc, but everybody will figure out Basecamp.
Personal tasks	Coordinated tasks	If other people don't need to know about it, it doesn't need to be in Basecamp

Strategic Intent

Strategic Intent

I want to change the world!

inspirational

vision
mission

**strategic
intent**

To get everyone in the UK online by the end of 2012.

bland

values

quarterly
objective

Increase profits
5% over last
years results!

Leadership!
Innovation!
Teamwork!

general

concrete

Strategic Intent

Stop wordsmithing and start deciding.

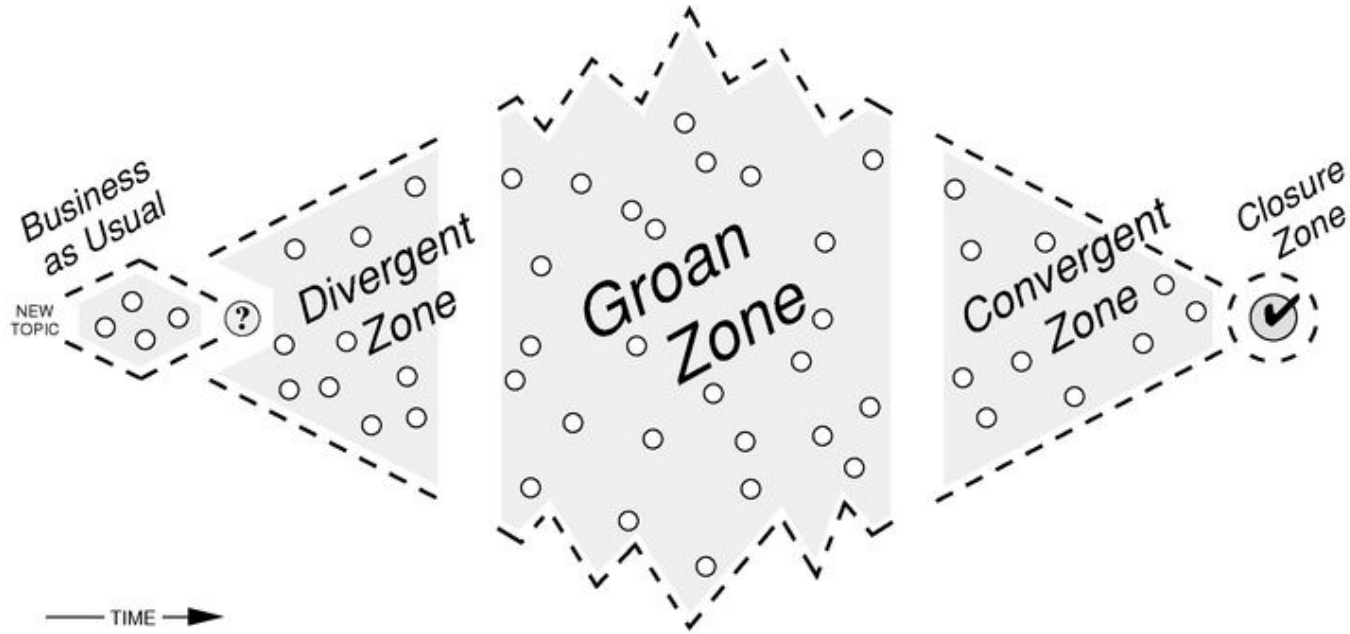
Ask: Will we know when we are done?

Narratable: From the new intern to the investor.

Work in Progress: Today is only version 1!

The End!

The Diamond of Participation



Spikes: Seek & share insights!

Don't ask yes/no
questions.

Prototype to Learn

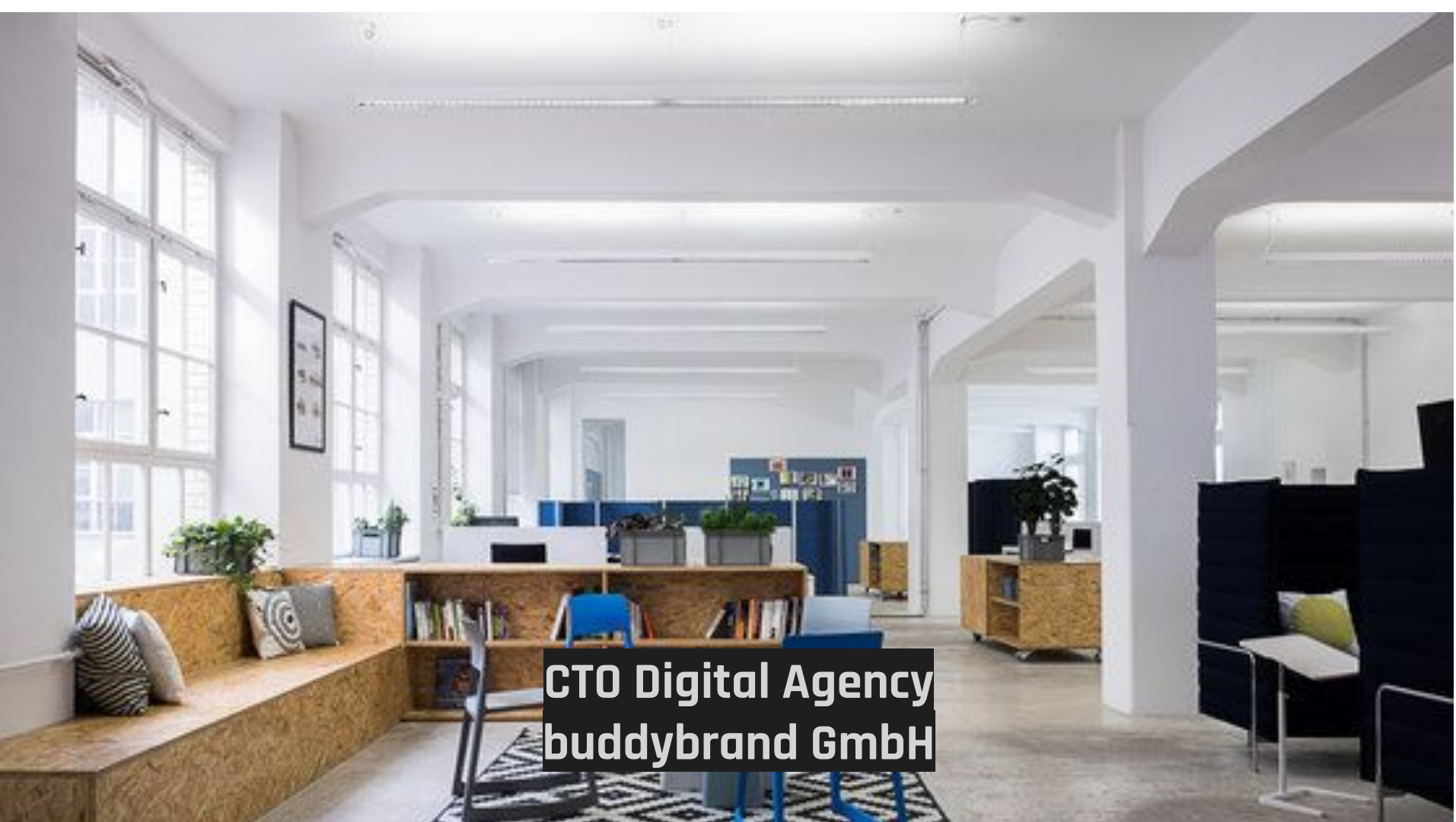
Challenge assumptions with data!



RFC for Collaboration

Company	Is There a “Central” Methodology?	What Project Management “Methodology” Is Typically* Used for Engineering Projects?	Who Typically Leads Engineering Projects?
Amazon	No, teams can choose	Plan (6-pager)->Build (iterate)->Ship	Tech lead
Apple	No, teams can choose	Plan->Build (iterate)->Ship	Tech lead
Datadog	No, teams can choose	Plan (RFC)->Build (iterate)->Ship	Tech lead or an engineer
Facebook	No, teams can choose	Plan->Build (iterate)->Ship	Tech lead or an engineer
Google	No, teams can choose	Plan (Design Doc)->Build (iterate)->Ship	Tech lead or an engineer
Netflix	No, teams can choose	Plan->Build (iterate)->Ship	Tech lead or an engineer
Shopify	No, teams can choose	GSD (Get Shit Done, 6-week cycles)	Tech lead or an engineer
Spotify	No, teams can choose	Plan->Build (iterate)->Ship	Tech lead or an engineer
Uber	No, teams can choose	Plan (ERD)->Build (iterate)->Ship	Tech lead or an engineer

About me



**CTO Digital Agency
buddybrand GmbH**



CPO & CTO Influencer Marketplace BuzzBird GmbH



Inv.: 2008940-0

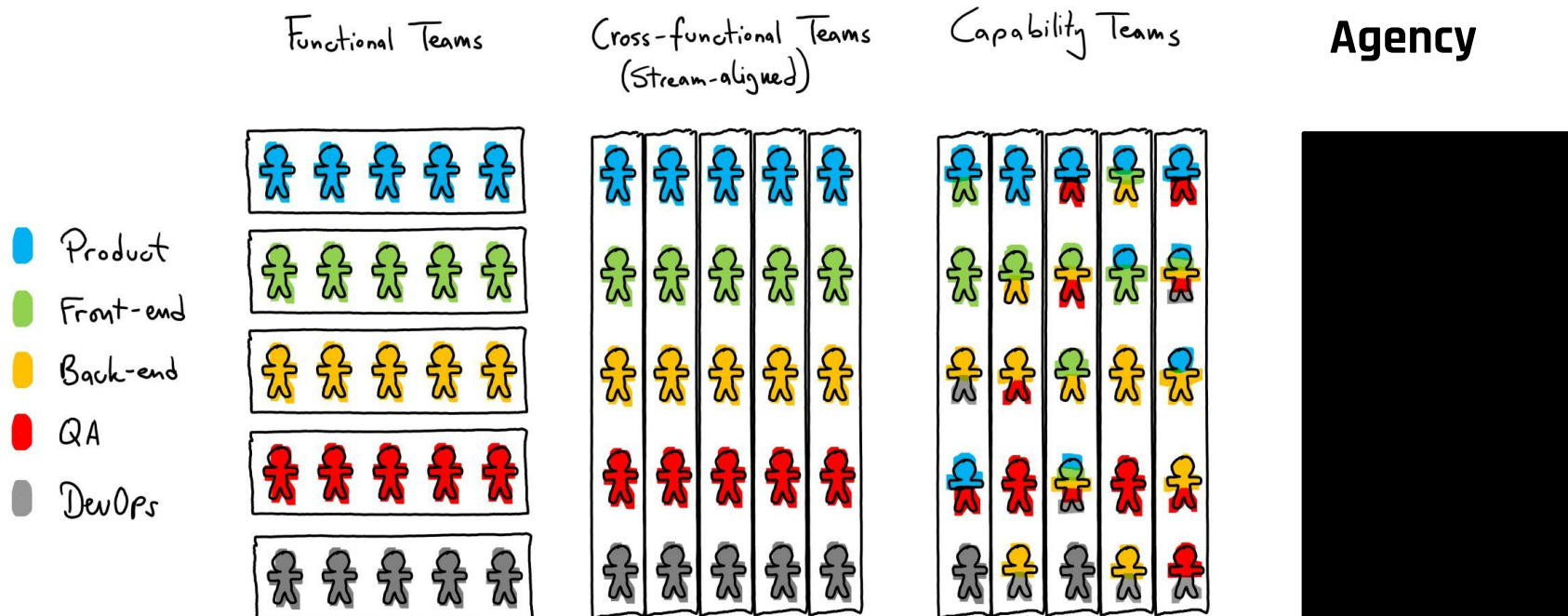
VOITH

Built Ilt Business Unit (Interim)
Voith GmbH & Co. KGaA



Leading Product & Tech (Interim)
edding AG

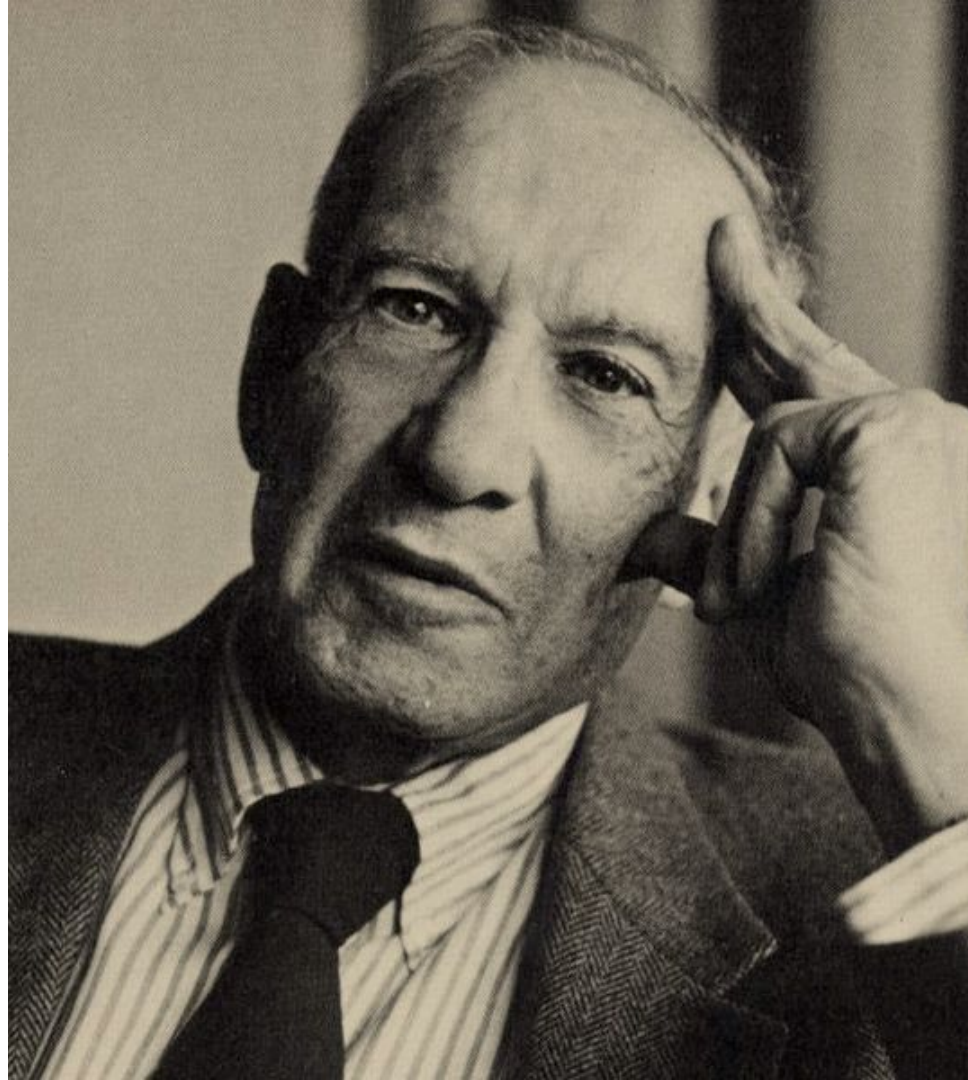
Teams: Insourced Engineering



Scaling Orgs

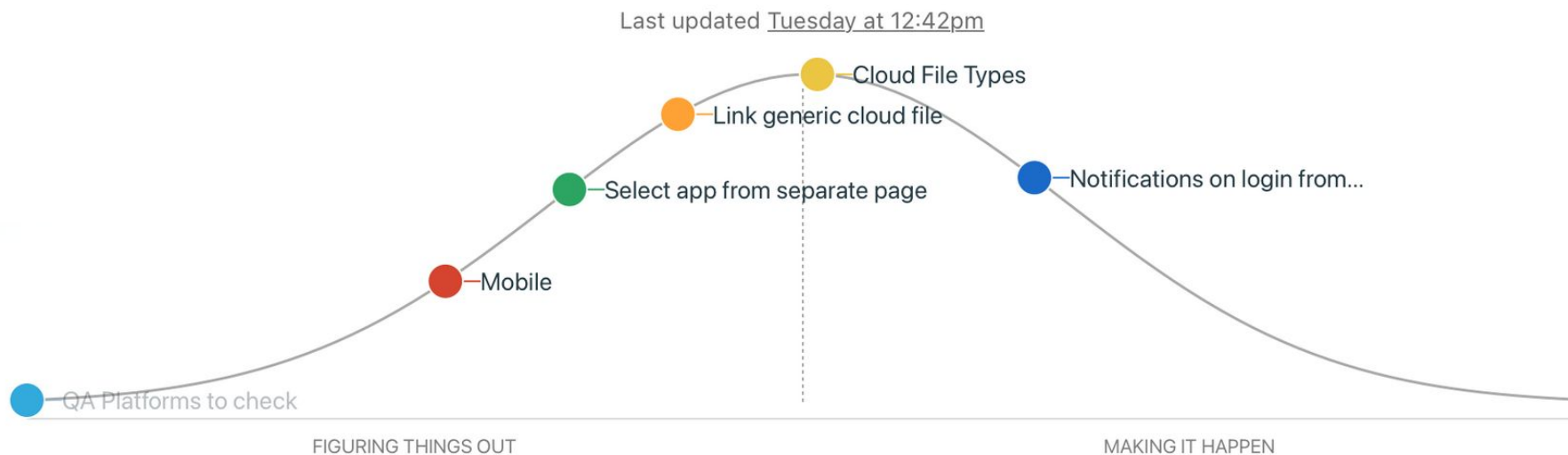
**“[only] what gets
measured, gets
managed.”**

(Peter Drucker)





Try: Scope Progress With Hillcharts



Progress is more like a hill than a straight line

Try: Moving the Needle



How far along are we?



How's the project going?

☒ On track ☐ Some risk ☐ Concerned

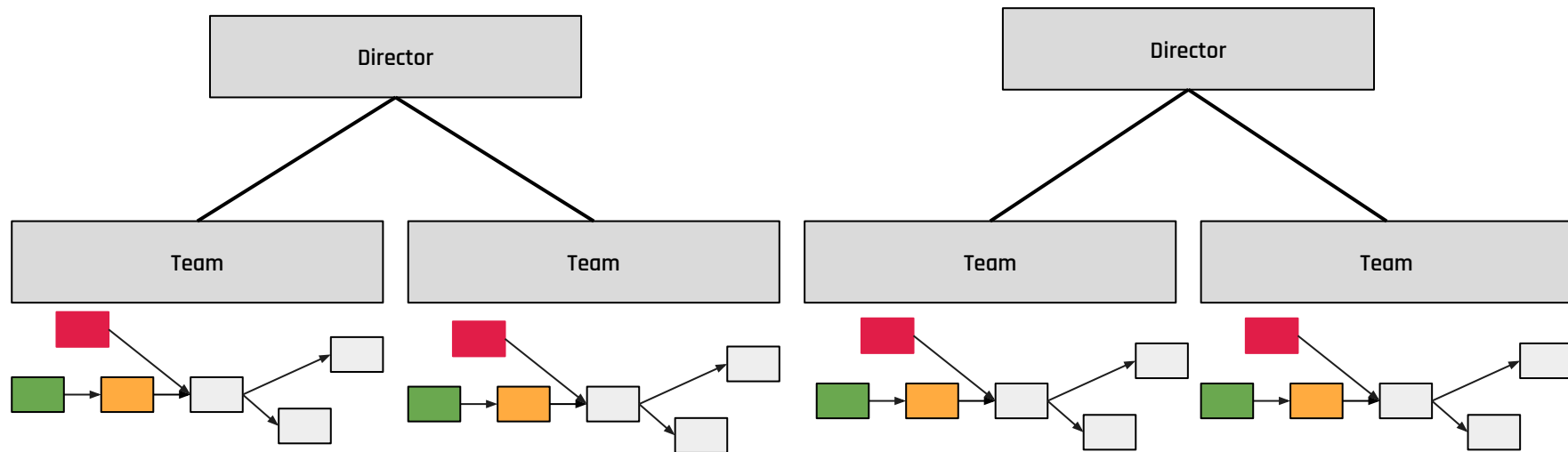
Describe why you moved the needle:

B *I*

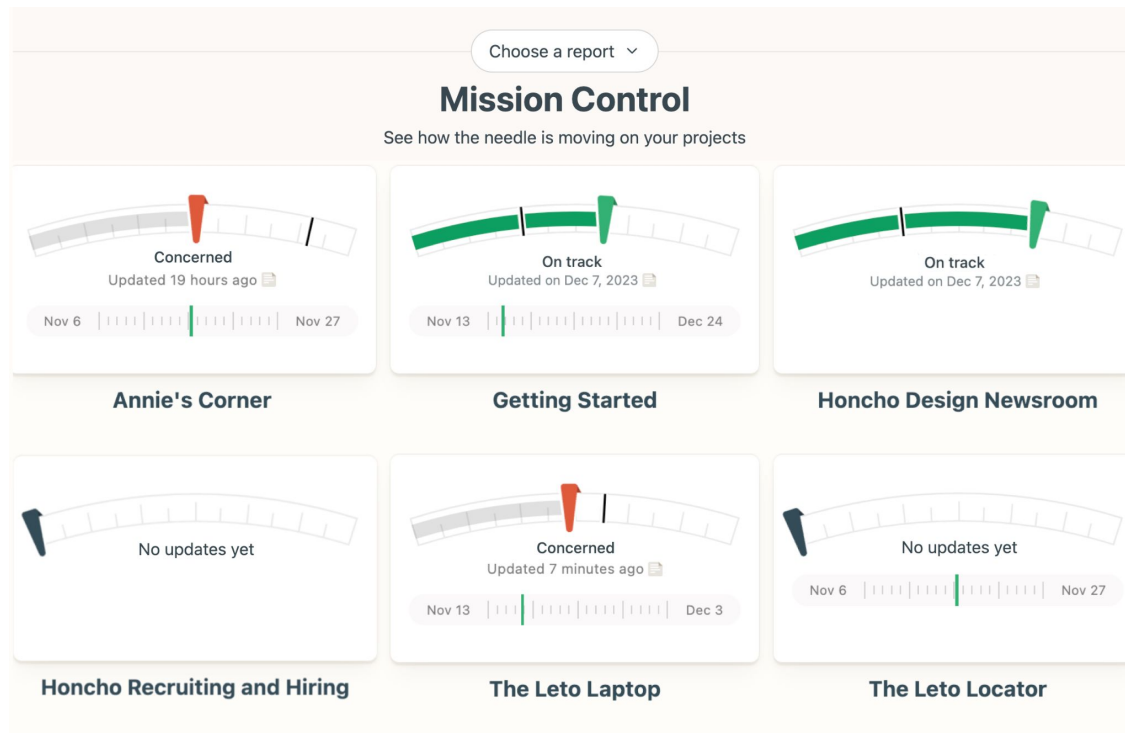
Gina's reviewed the three options and we're leaning strongly towards two of them. We'll iterate and go through the next round of reviews!

Progress vs. fixed timebox (appetite!)

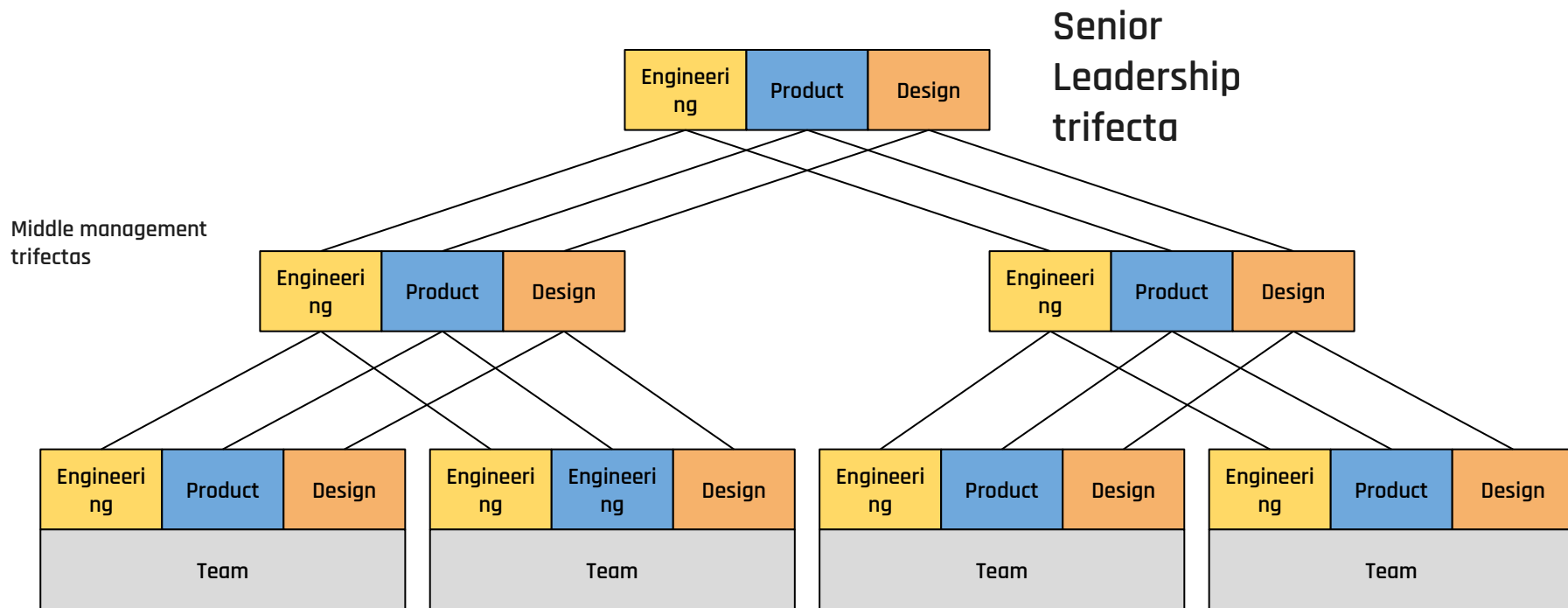
Implement Reporting That Enforces Habits



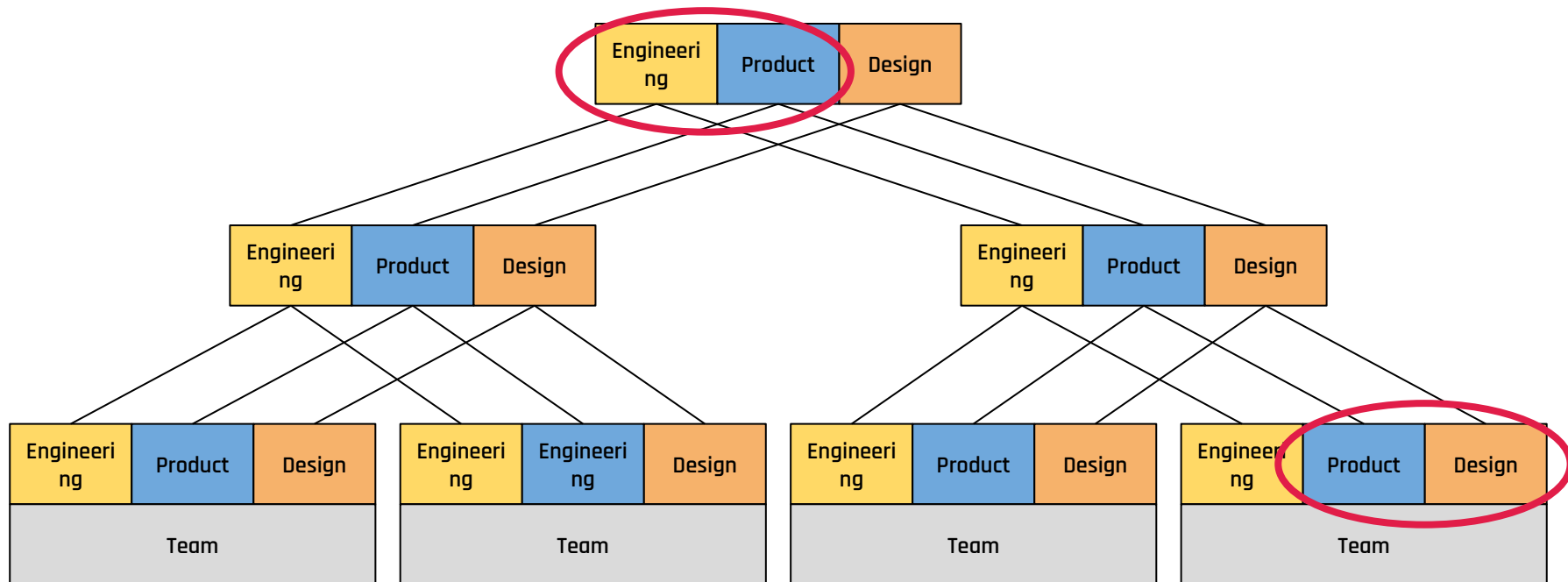
Try: Roll-Up Dashboards for Progress



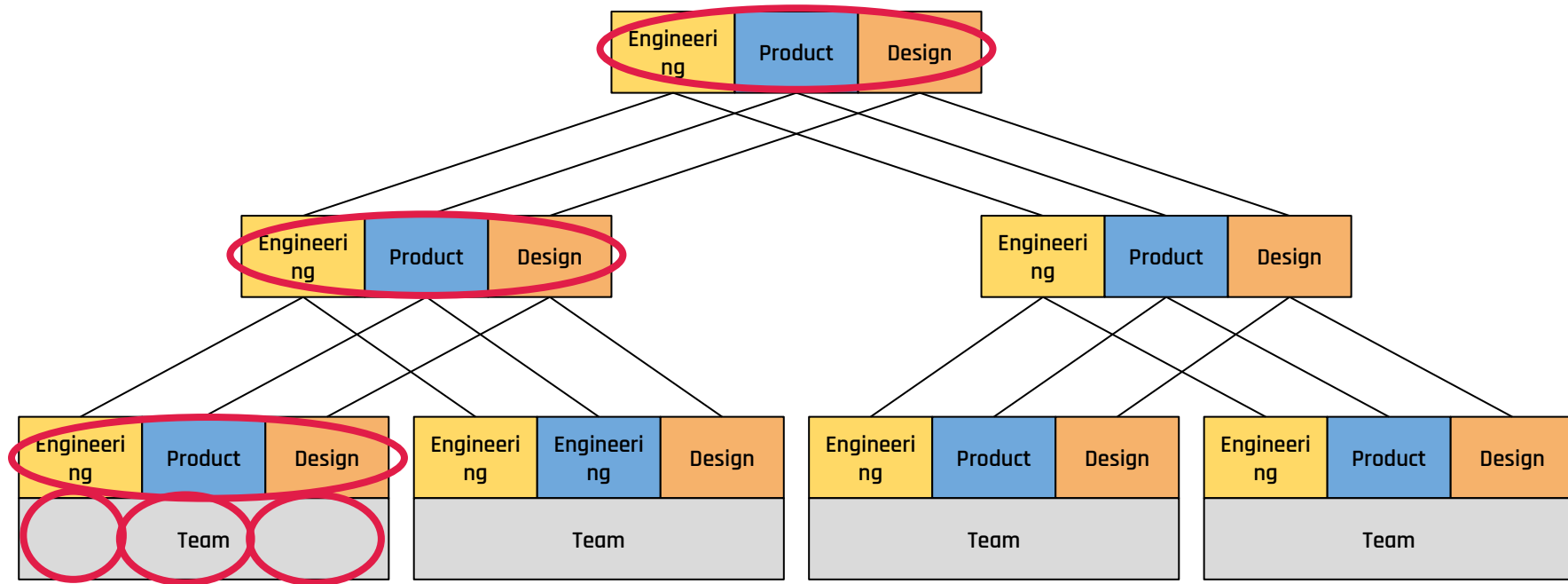
Trifectas All the Way Up



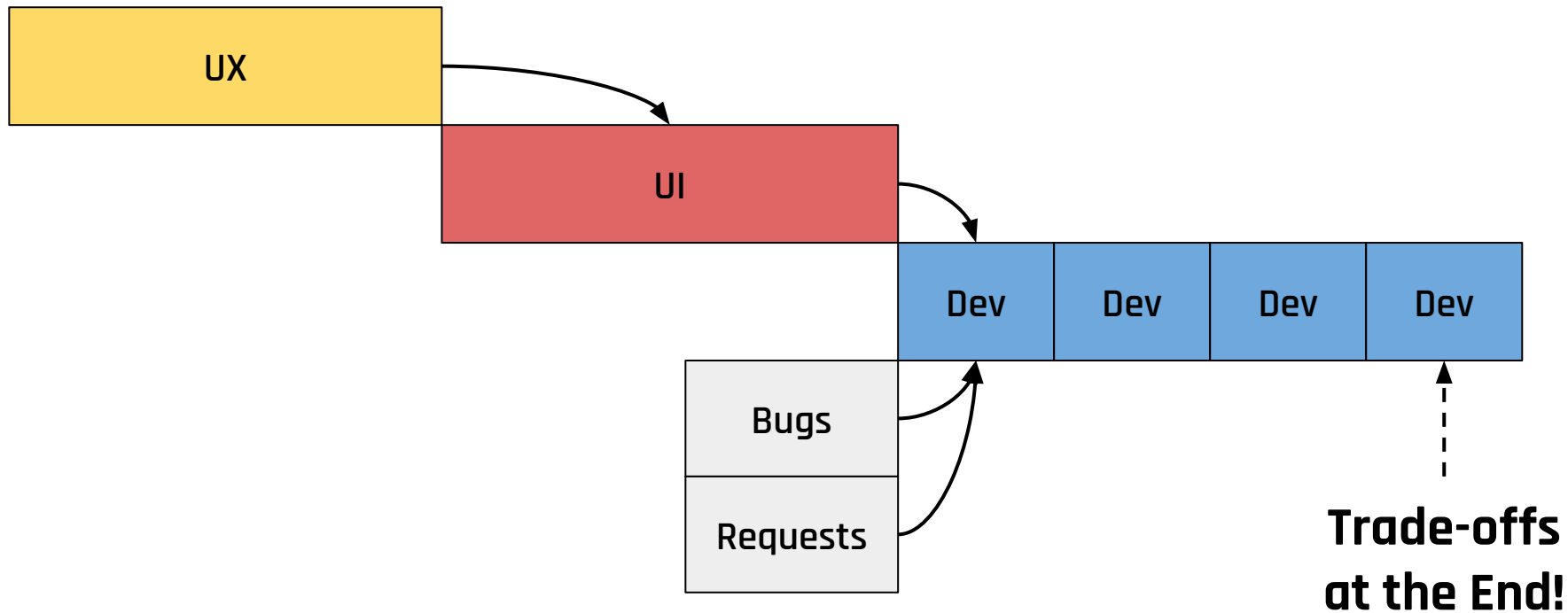
Try: Hire & Foster Interdisciplinary Leaders



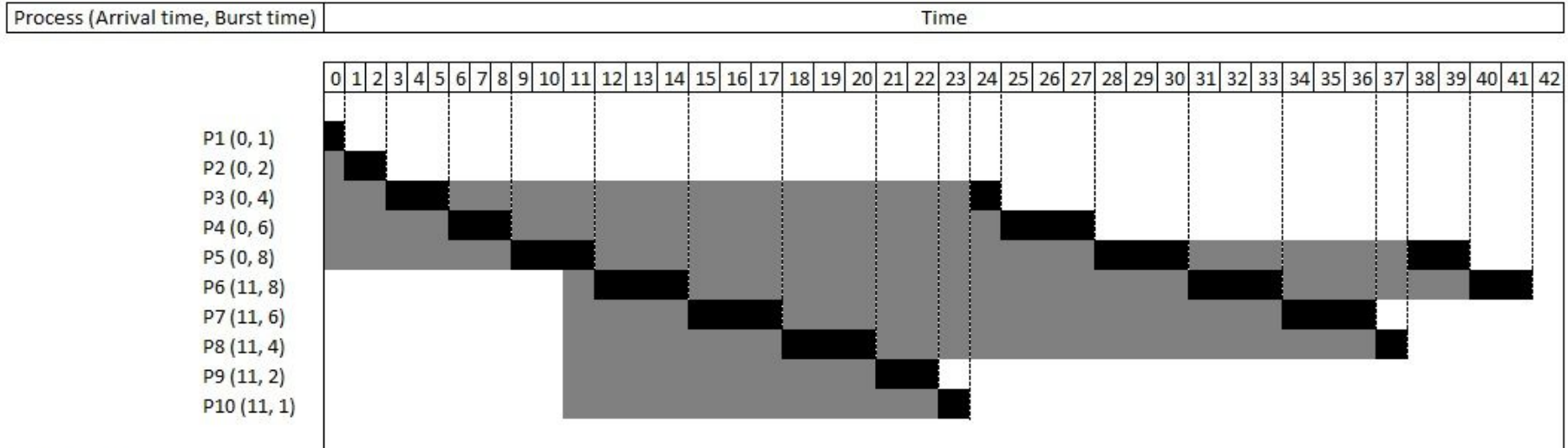
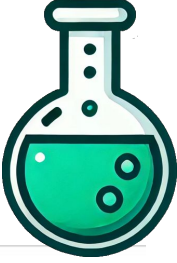
Bringing It Together: Small Teams = Trifectas



Scrum(fall): Small Cycles, Large Team



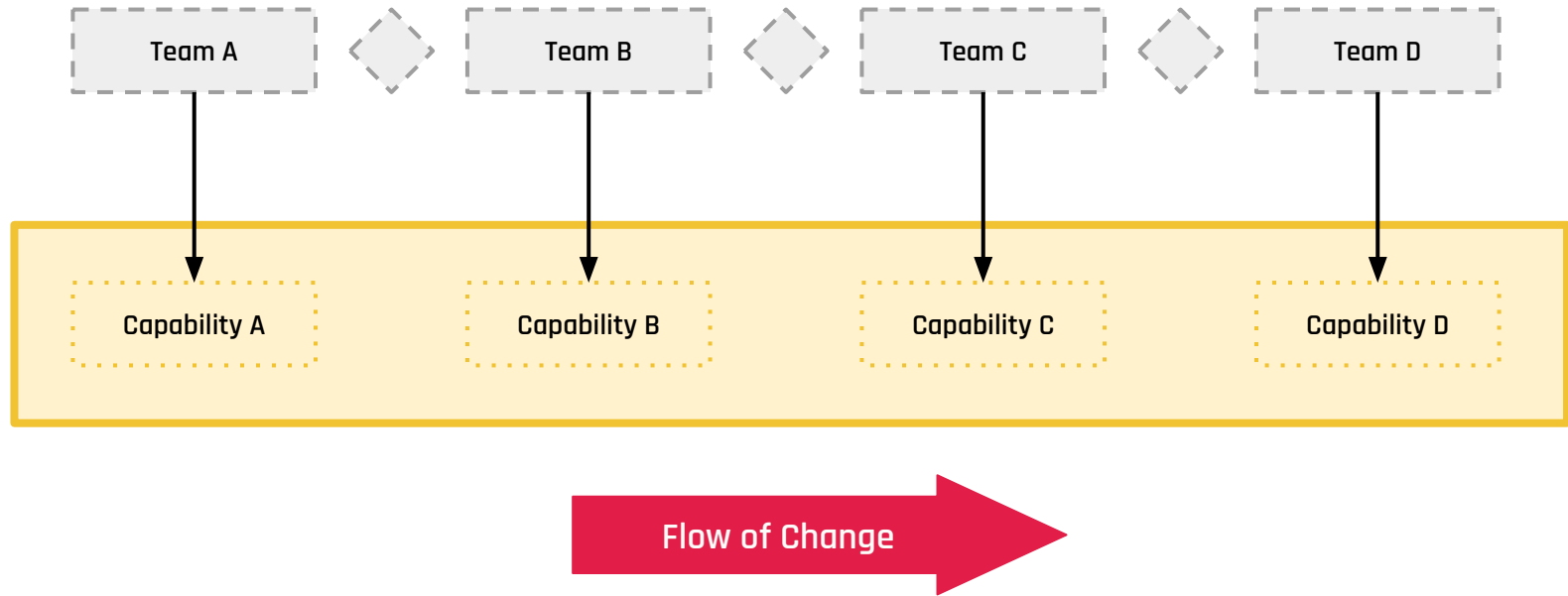
Try: Round Robin to Activate Everybody



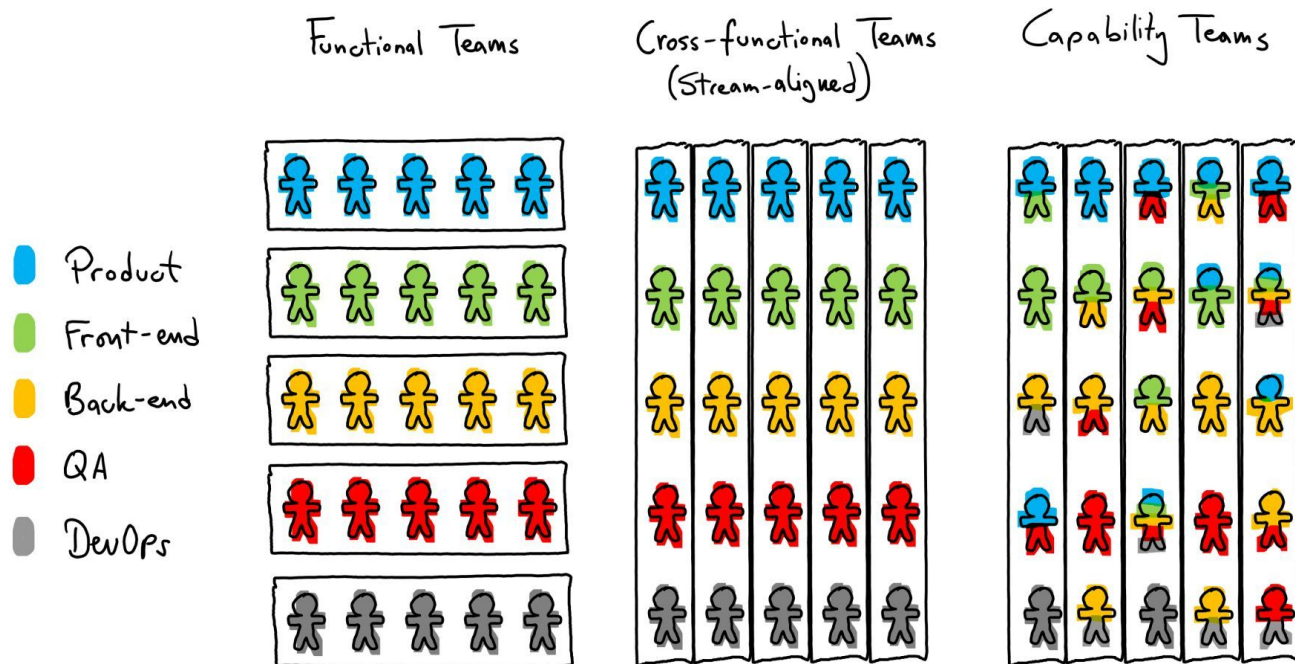
Quantum = 3

Wait time
 Burst time

Optimize Teams for Value Streams - End to End



Teams: Capabilities, Not Specialties



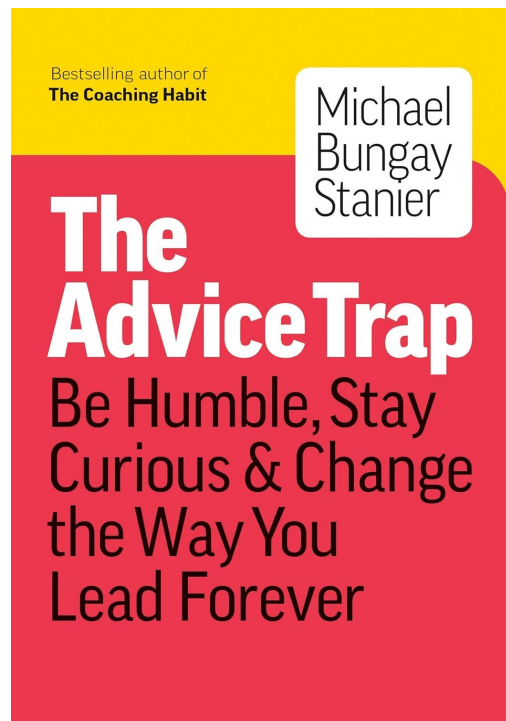
Try: Hire & Foster Interdisciplinary Roles



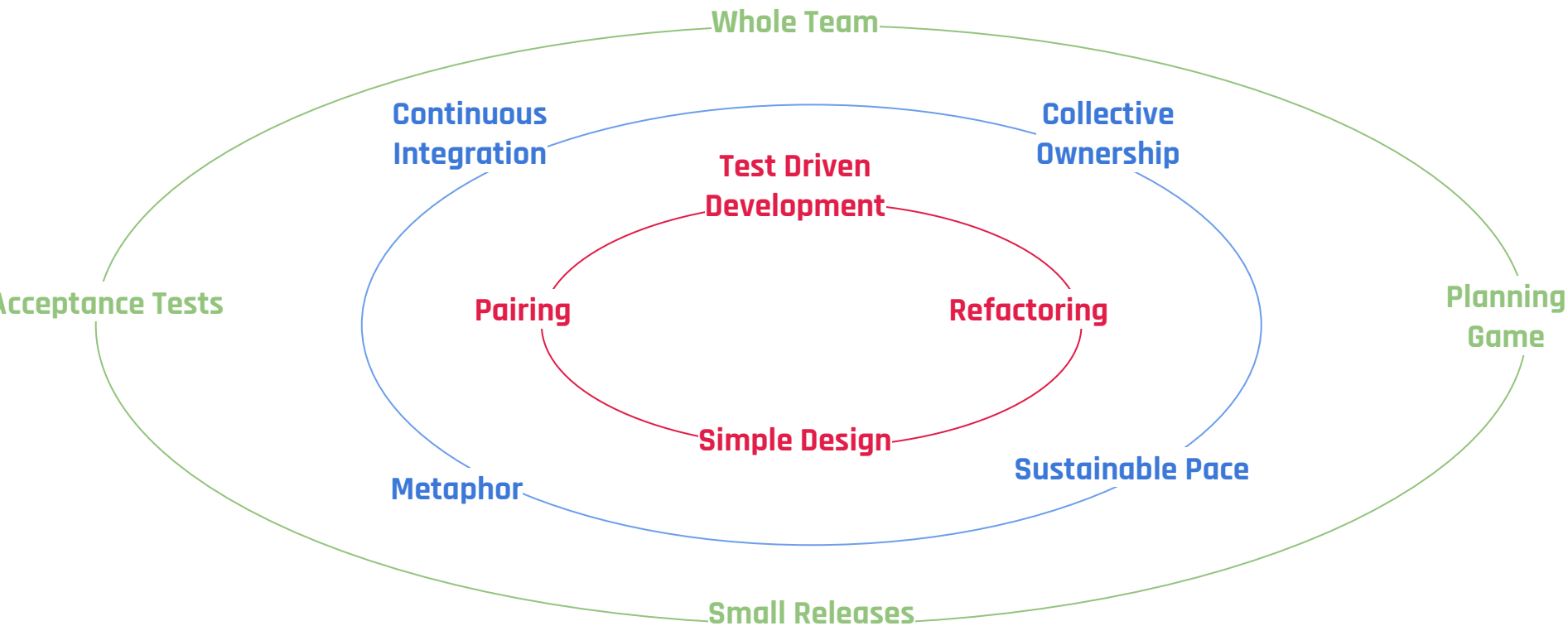
- ✓ Customer obsession
- ✓ Analysts of usage data and the competitive landscape
- ✓ Always prototyping and experimenting



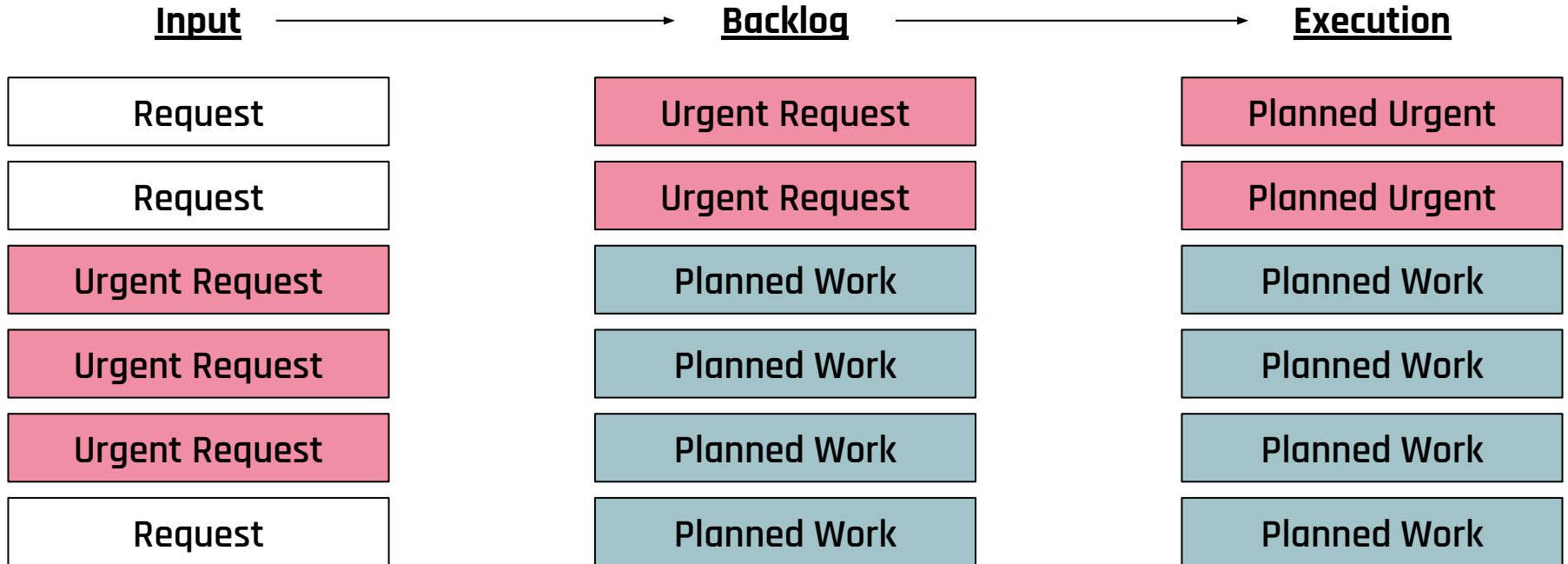
Don't Brief, Coach!



But Business Took Over



Agile Became a Micromanagement Tool



**“Organizations [that
design systems]
produce designs that
mirror their
communication
structures”**

(Melvin Conway, 1967)



**“So much complexity in
software comes from
trying to make one thing
do two things.”**

(Ryan Singer, Shape Up)



Slicing Solutions: Scope by Scope

Shippable in Isolation!



Try: Build Initiatives Without a Backlog



Try: Backlogs Only for Reactive Work



Todo	Work in Progress	Done

**“It's easier to ask
forgiveness than it is to
get permission.”**

(Admiral Grace Hopper)

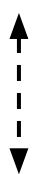


Bringing It Together: Map the Work to the Org

Slice Objectives



Slice Problems

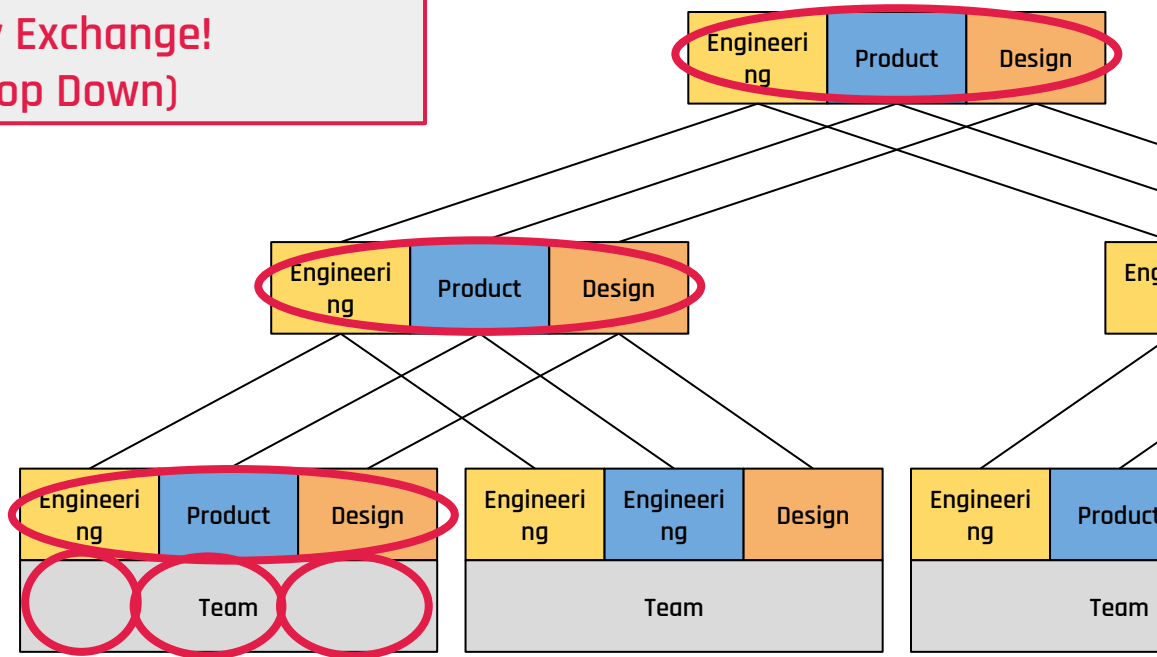


Slice Solutions

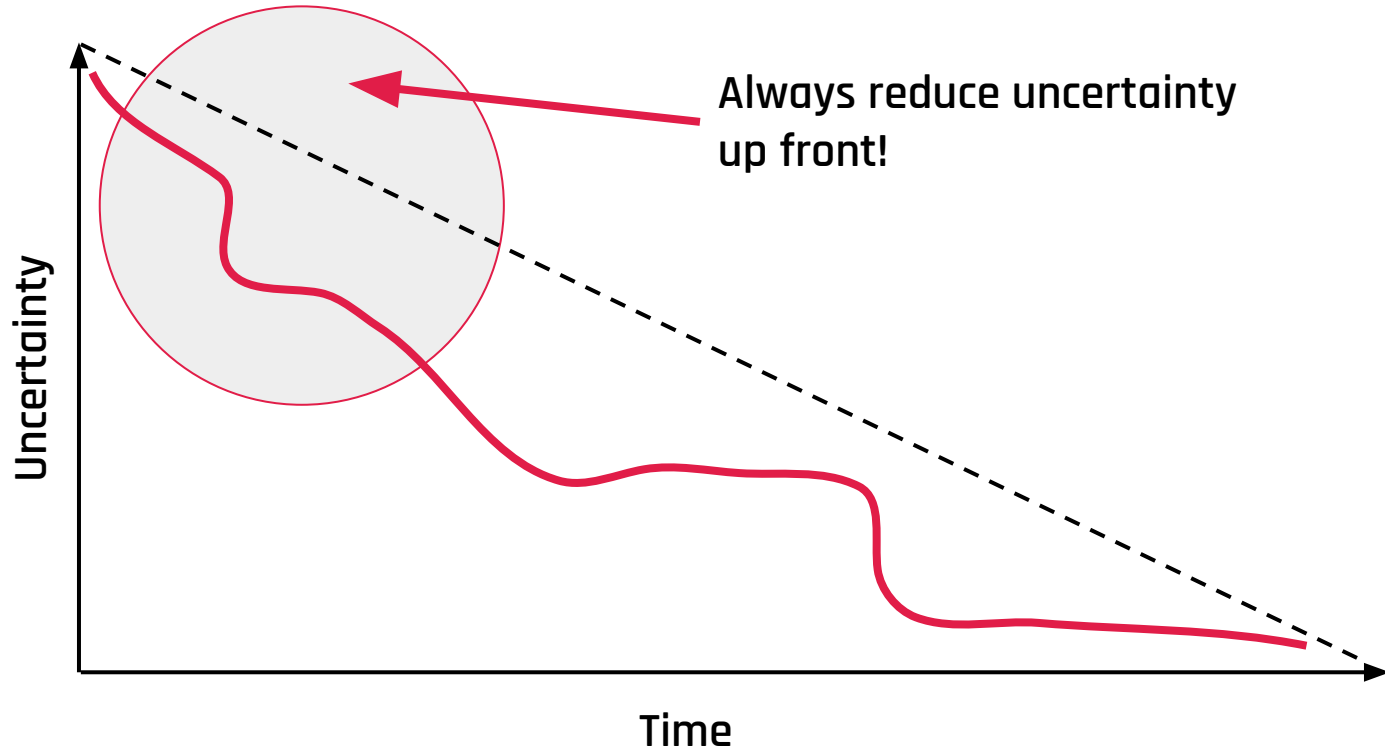


Slice Delivery

2-Way Exchange!
(Not Top Down)



Slicing Solutions: De-Risking





AI HYPE

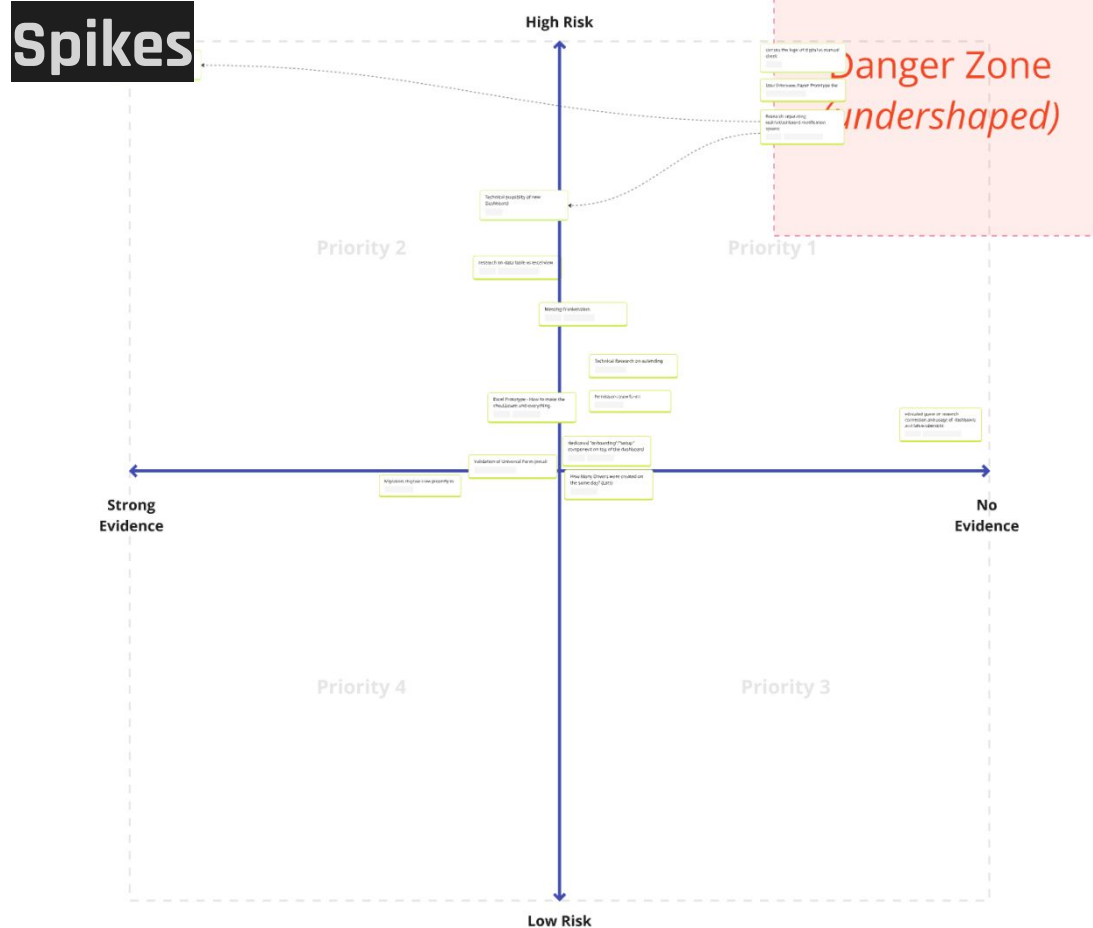
Framework for Decision Making

Facing Reality 

Slicing Work

Aligning Teams

Emancipating People



Open Questions