

# Agile Dev Better Software DevOps **WEST**

A TECHWELL EVENT

## **AW6**

Agile Product Development  
Wednesday, June 6th, 2018, 1:30 PM

## **Beating the Feature Factory Mindset**

Presented by:

**John Cutler**

Independent Consultant

Brought to you by:



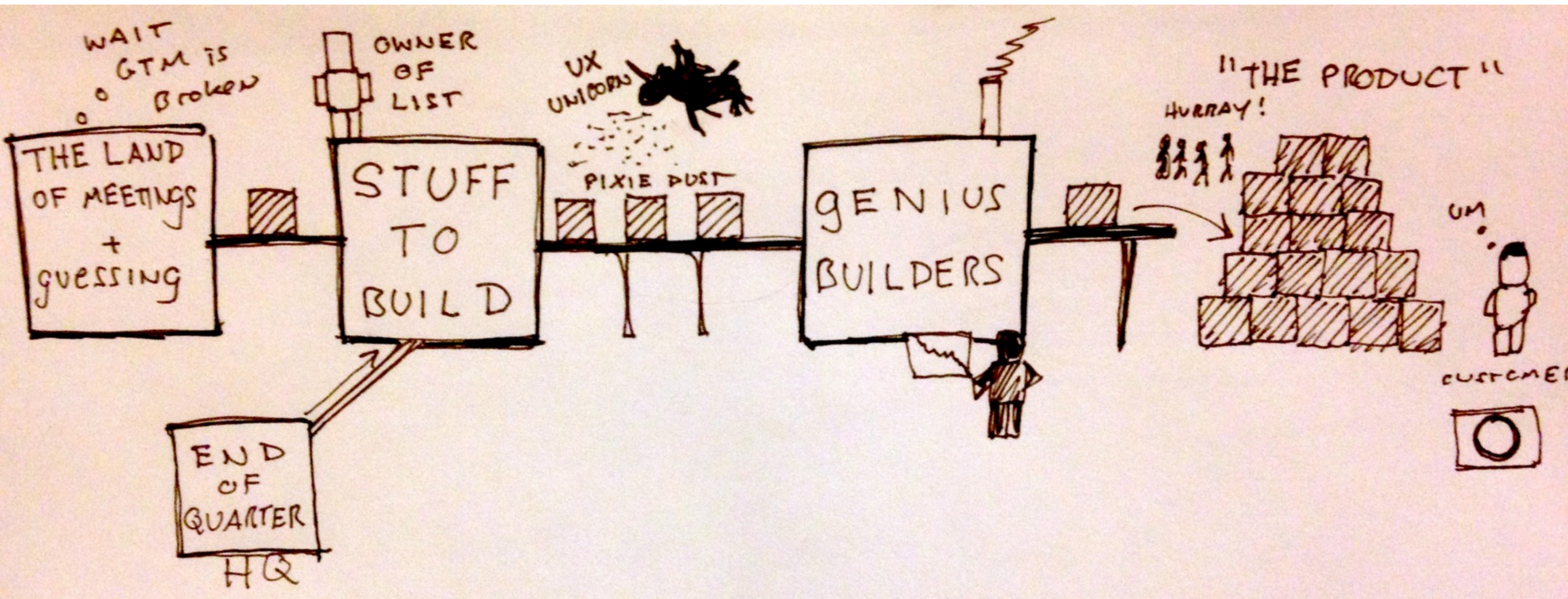
350 Corporate Way, Suite 400, Orange Park, FL 32073  
888-268-8770 · 904-278-0524 - [info@techwell.com](mailto:info@techwell.com) - <https://www.techwell.com/>

# John Cutler

## Independent Consultant

John Cutler is a multiple-hat-wearer and product development nut. He loves wrangling complex problems and answering the "why" with qualitative and quantitative data. John blogs on Medium in Hacker Noon. He has a perspective that spans individual roles, domains, and products. John knows agile product management and frequently shares his knowledge on Twitter.

# Beat the Feature Factory



**@johncutlefish**



baking a birthday cake

touchscreen

be a good listener

go fishing and catch a fish

user empathy Siri university degree

a world class IT conference

get married

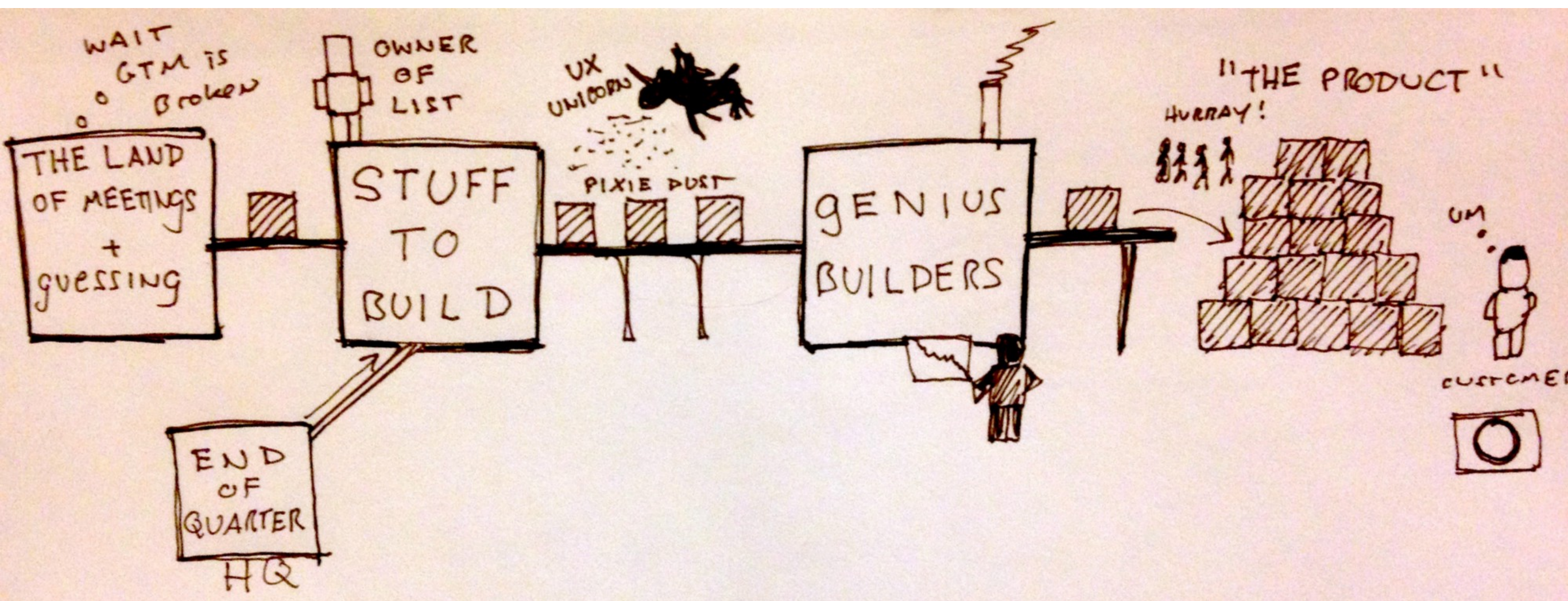
marriage

project, feature, or mission?



**The Post**

**@johncutlefish**



# 12 Signs You Are Working in a Feature Factory

**No measurement**

**Rapid shuffling of teams/projects**

**Success theater**

**Infrequent acknowledged failures**

**No connection to core metrics**

**No product manager retrospectives**

**Prioritization over validation**

**No tweaking/iteration**

**Culture of hand-offs**

**Large batches**

**Chasing upfront revenue**

**Shiny objects**



We crave impact.

We appreciate rigor.

We love solving problems.

We sense new opportunities.

...and we need to deal  
with the mess.

**(Or leave the company before it gets too bad)**

**“I hop between companies, and look for who is using interesting technologies. That is usually worth a couple good years. Of course, I had always dreamed of helping people with technology. But it looks like our businesses aren’t designed to do that. They all go to shit.”**

*–Developer*



**“We never really get to iterate. So of course I am going to freak out about the design. Design is my craft. I was trained to use design to help people. I thought the whole point of software was the ability to keep improving. But that just doesn’t happen.”**

*–Designer*

**“I fucking hate MVPs.  
It is shorthand for cut corners and ship.”**

*–Seems Like Everyone*



**The Post**



**Software, Oh Software**

**@johncutlefish**

Everything as a Service (EaaS)

Customers rent products

Customers rent access to a stream of innovation

Customers buy on features, renew on value

Customers expect to be able to walk away

Product is not “shipped” (and built repeatedly)

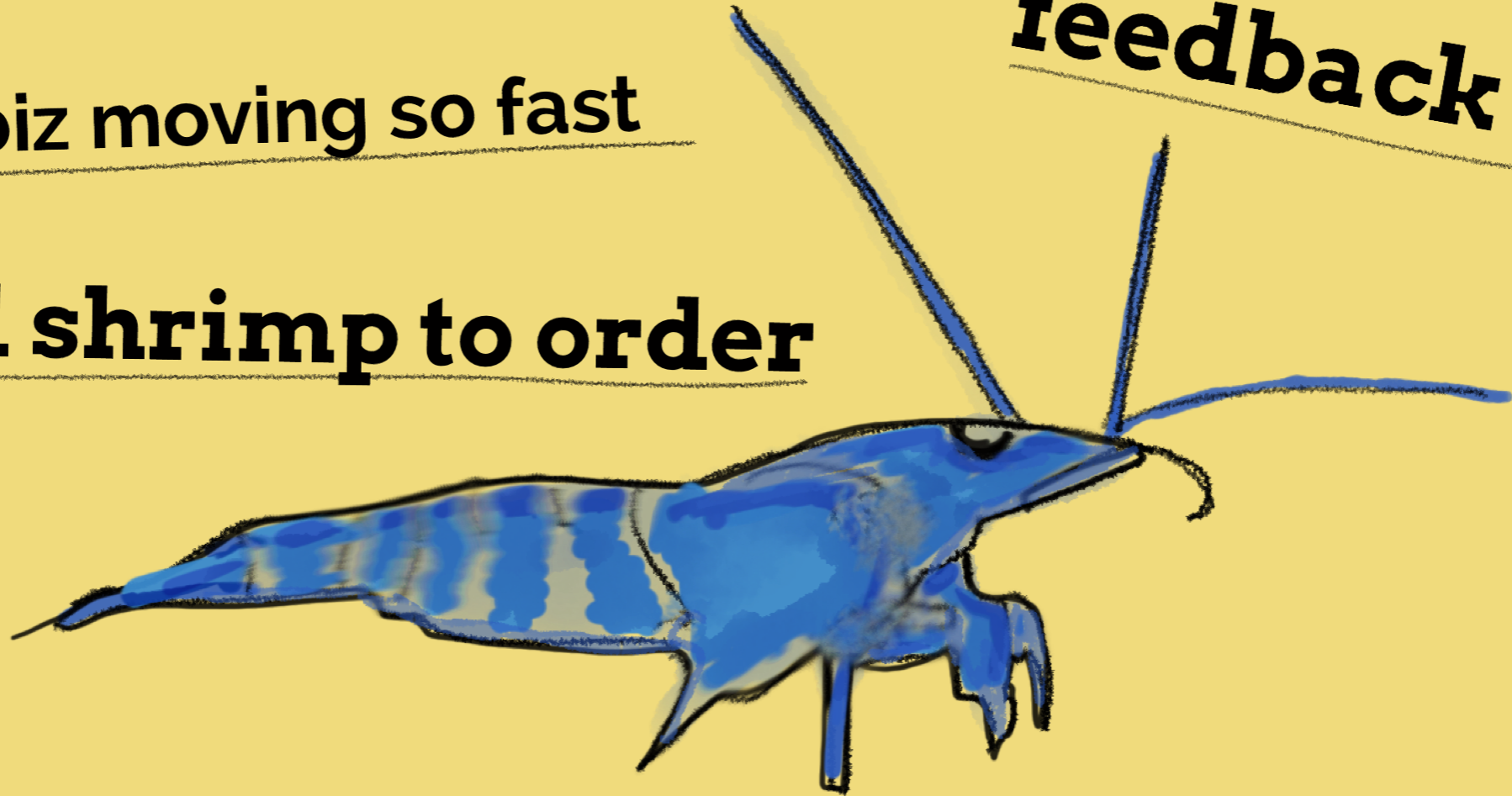
**Software**

people want their shrimp now

shrimp biz moving so fast

feedback 24-7

i breed shrimp to order



amazon is setting the standard, man

everything moves so damn fast

shrimp as a service

**Software**



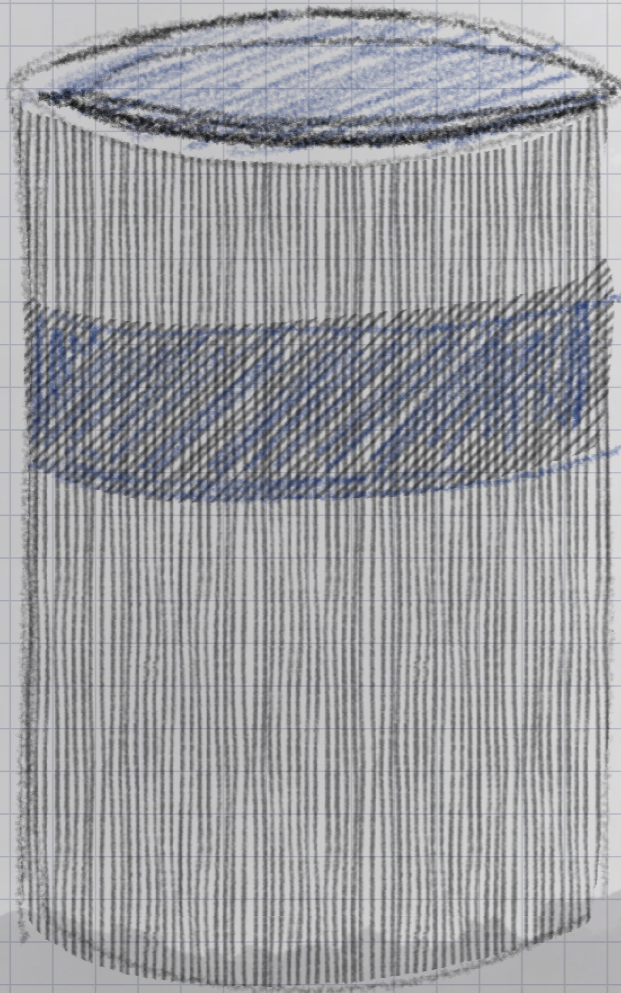


**Complexity**

**“It is like building a hotel iteratively, but only adding stuff. Creepy hallways to nowhere. 1000s of rooms, most unused. Layers of paint (no sanding). And then trying to build a boutique hotel on top of that. And then an amusement park.”**

**Software**

... OK ✓



- MAKE BREAKFAST
- MANAGE MY PORTFOLIO
- HELP ME GET MARRIED
- DIAGNOSE THAT ISSUE IN PRODUCTION

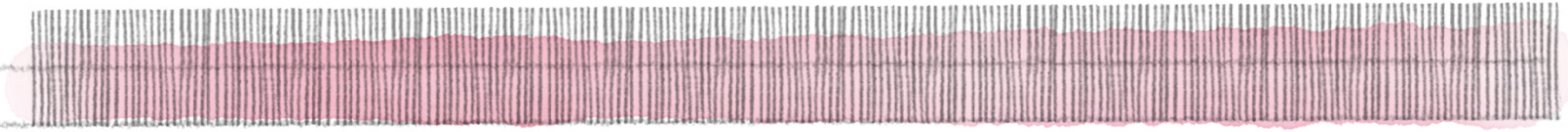
Software

**Features are temporary.**  
**Products are temporary.**  
**Technologies are temporary.**

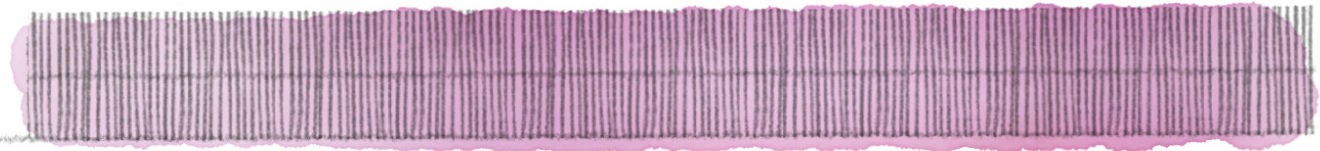
**Software**



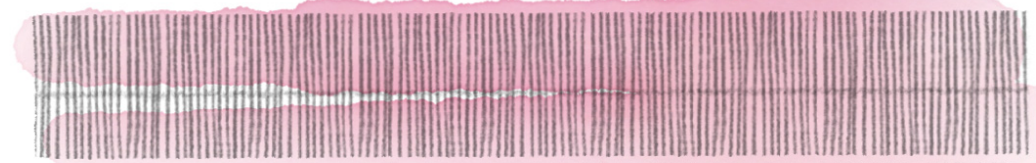
Feature



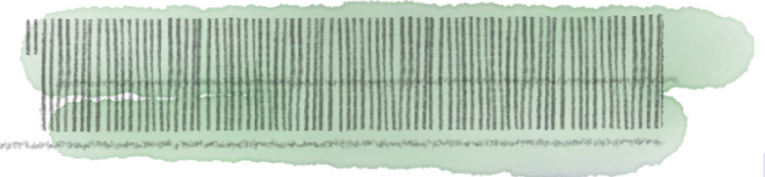
Feature



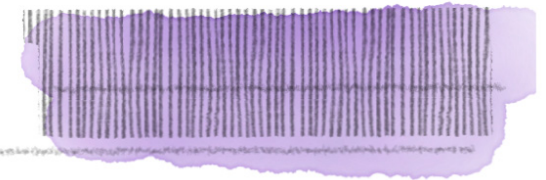
Feature



Feature



Feature



**Software**



"The Product"

Job

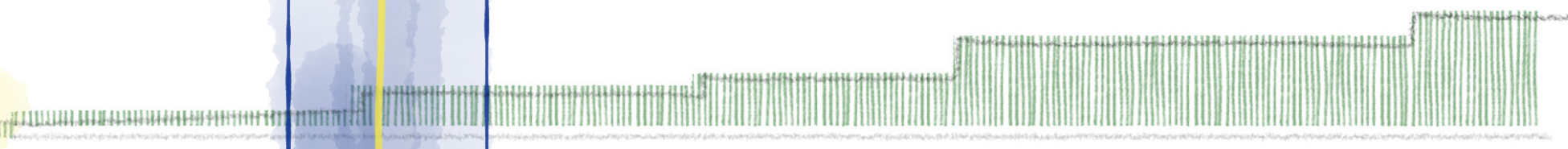
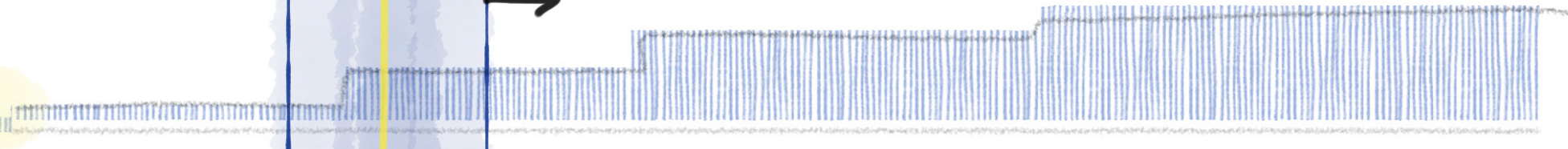
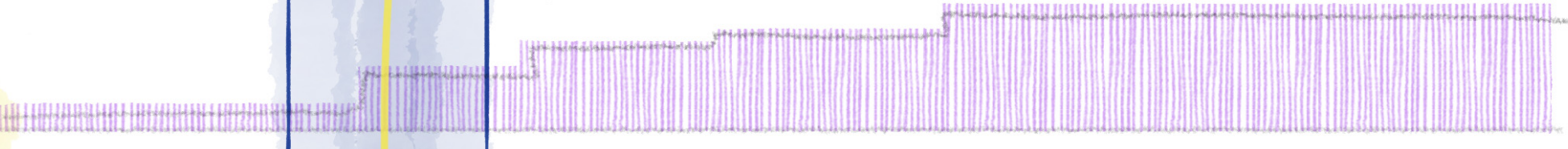
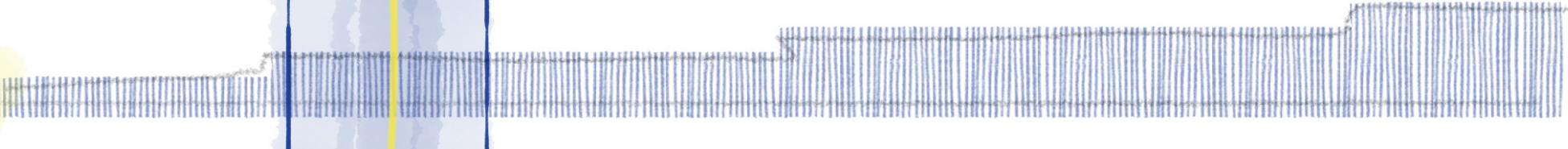
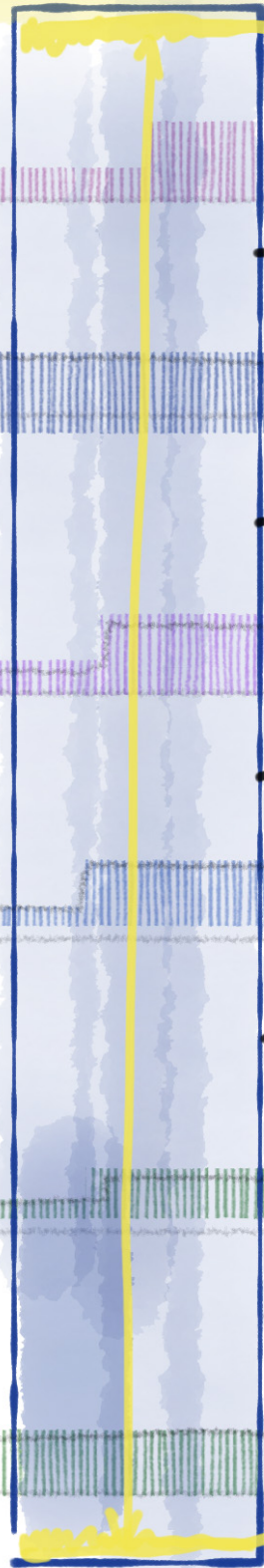
Job

Job

Job

Job

Job



Features will come and go...

Software



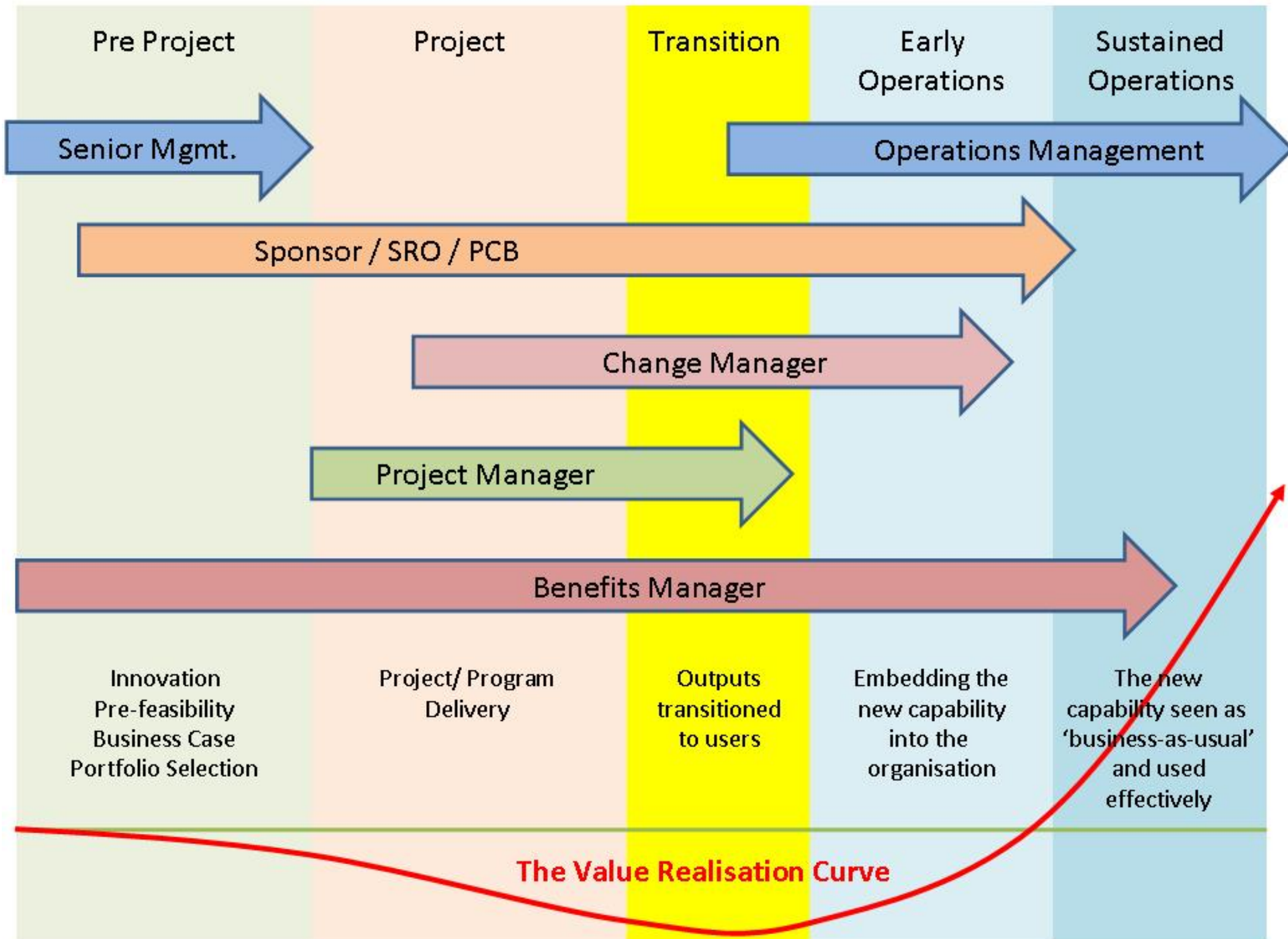
# The Problem

**@johncutlefish**

**“This is all well and good, but we can’t get anything out the door let alone figure out if it works. I mean things used to be better. We shipped a ton of stuff. But things have just slowed down since then. I’m not sure why.”**

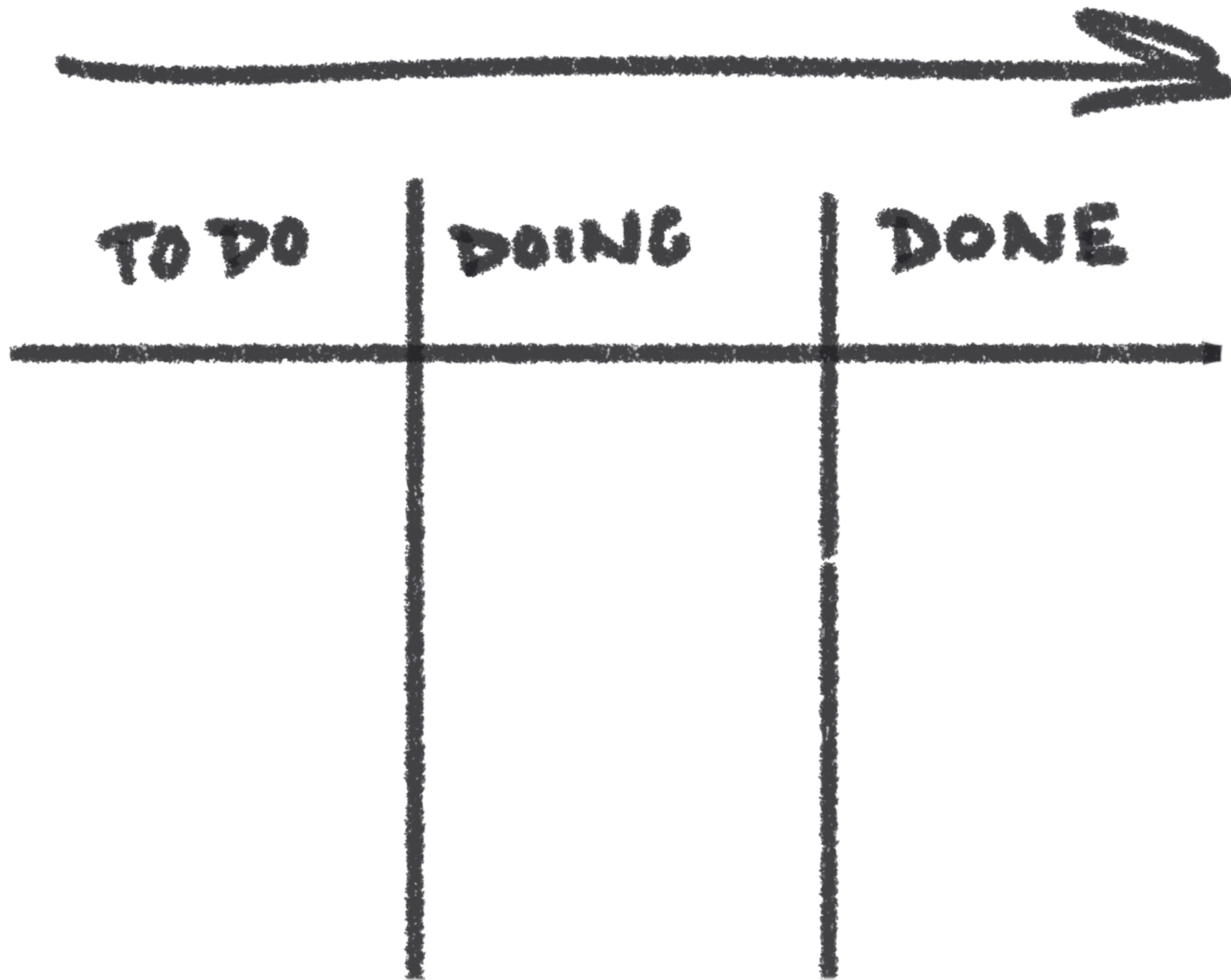
*–Manager*

**The Problem**



<https://mosaicprojects.wordpress.com/2012/11/03/who-manages-benefits/>

# The Problem



**The Problem**

# The Agile Scrum Framework at a Glance

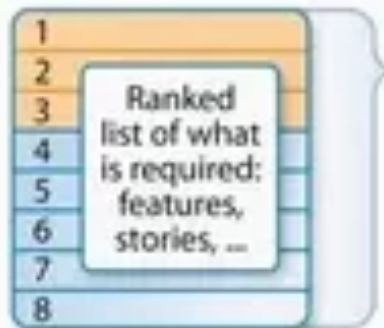
Inputs from Executives, Team, Stakeholders, Customers, Users



Product Owner



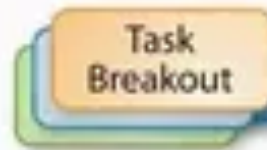
The Team



Product Backlog



Sprint Planning Meeting



Sprint Backlog

Sprint end date and team deliverable do not change



1-4 Week Sprint



Scrum Master



Burndown/up Charts

Every 24 Hours



Daily Scrum Meeting



Sprint Review



Finished Work



Sprint Retrospective

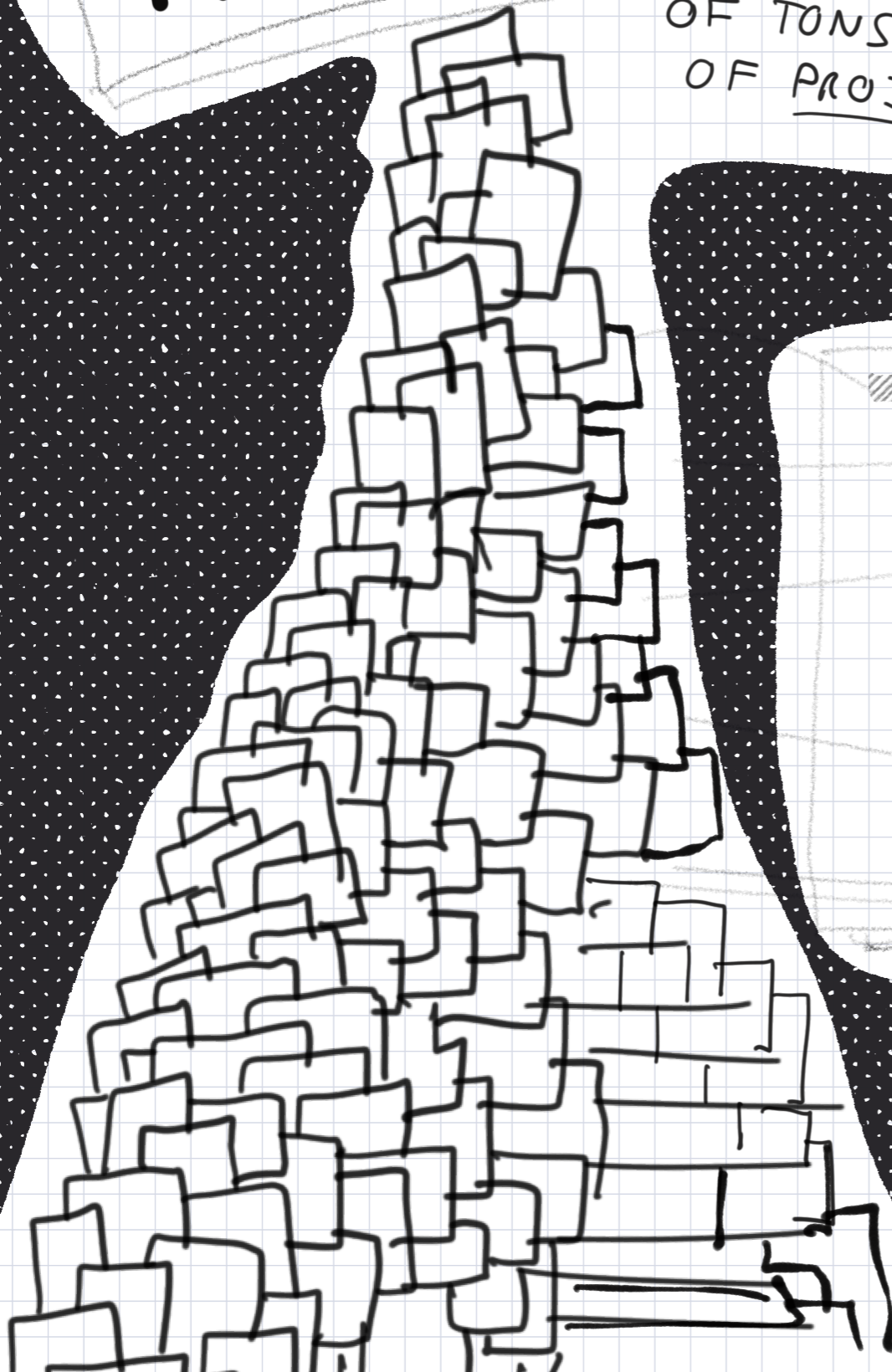


The Problem



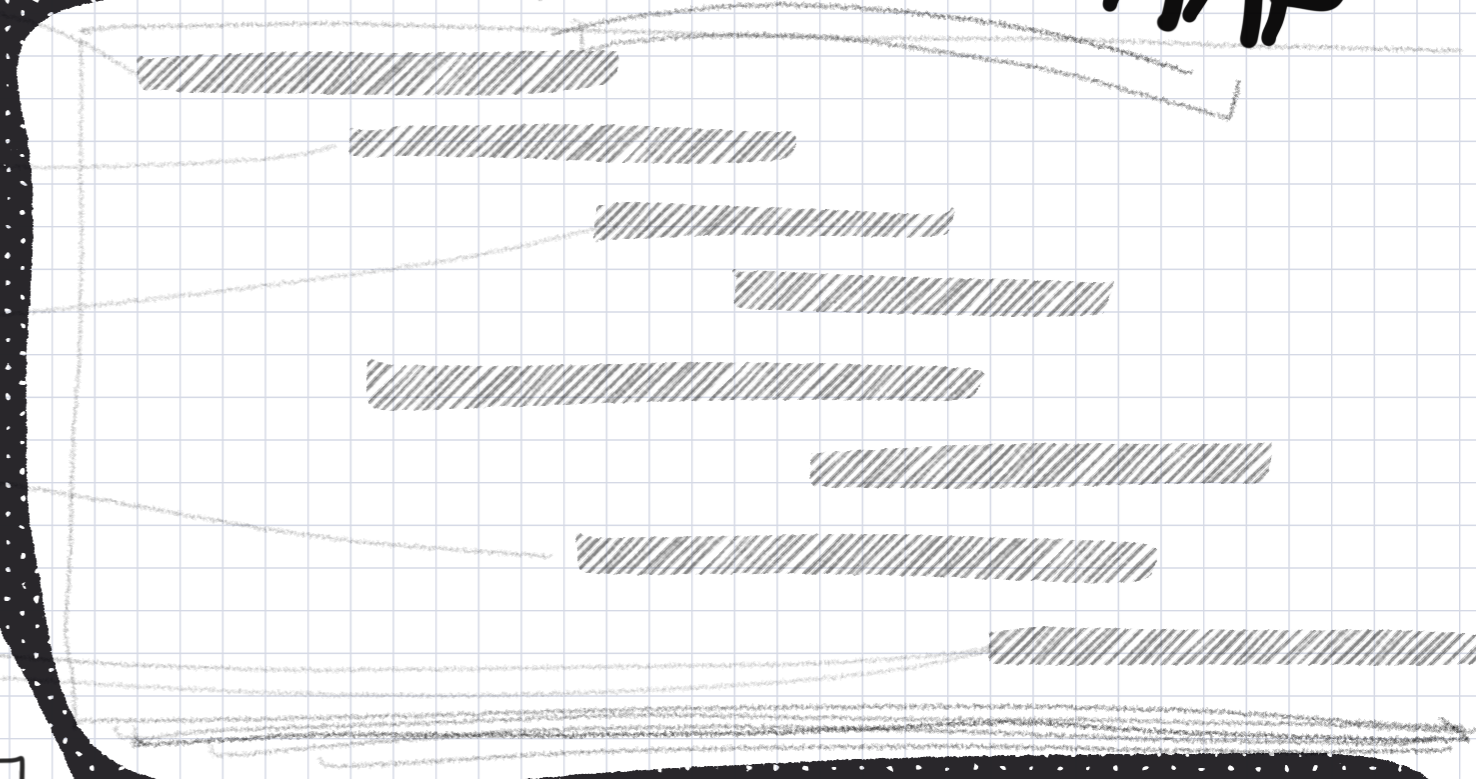
# THE BACKLOG

OF TONS  
OF PROJECTS

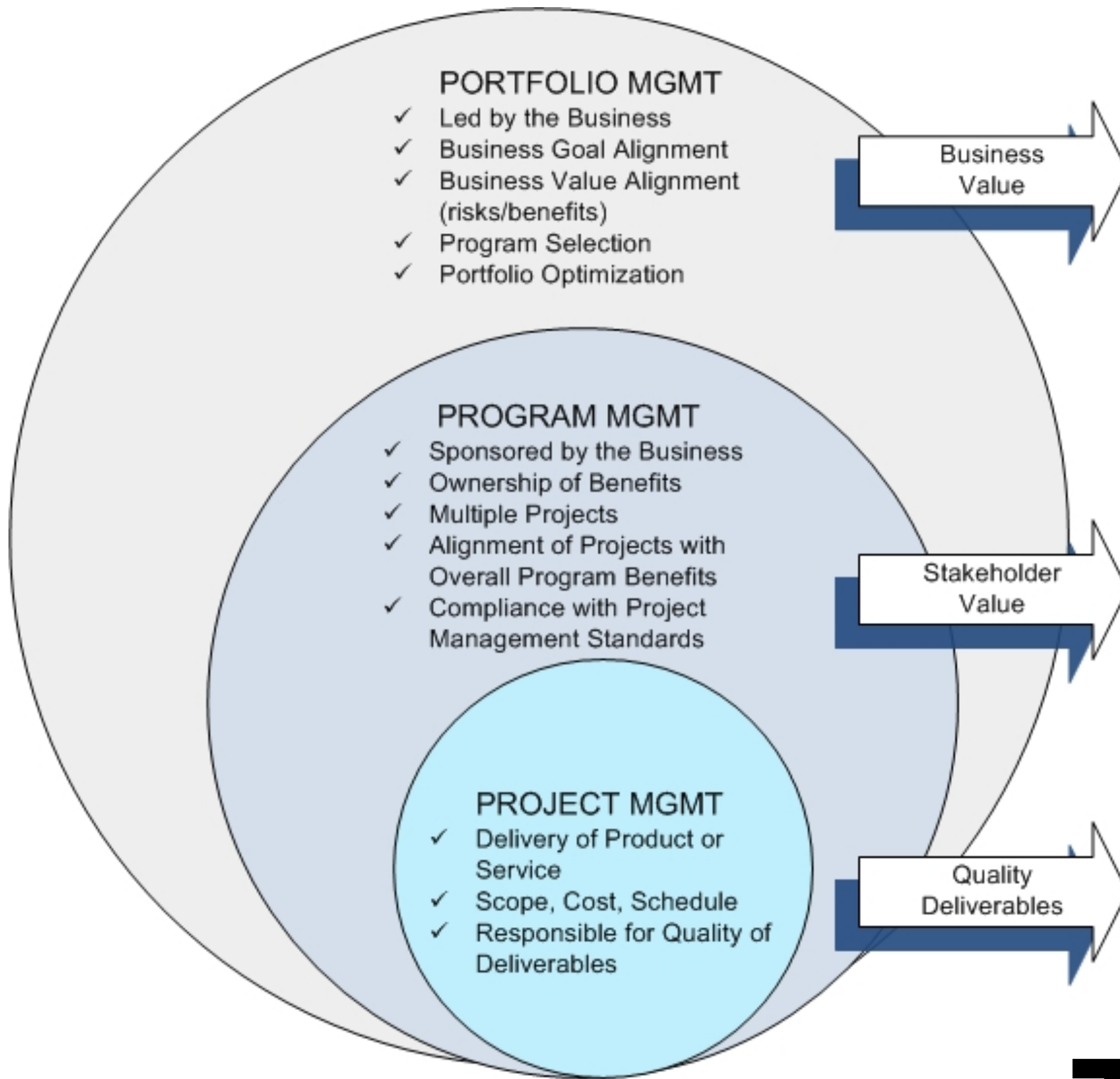


We equate  
VISION TO  
A PLAN

# THE ROADMAP



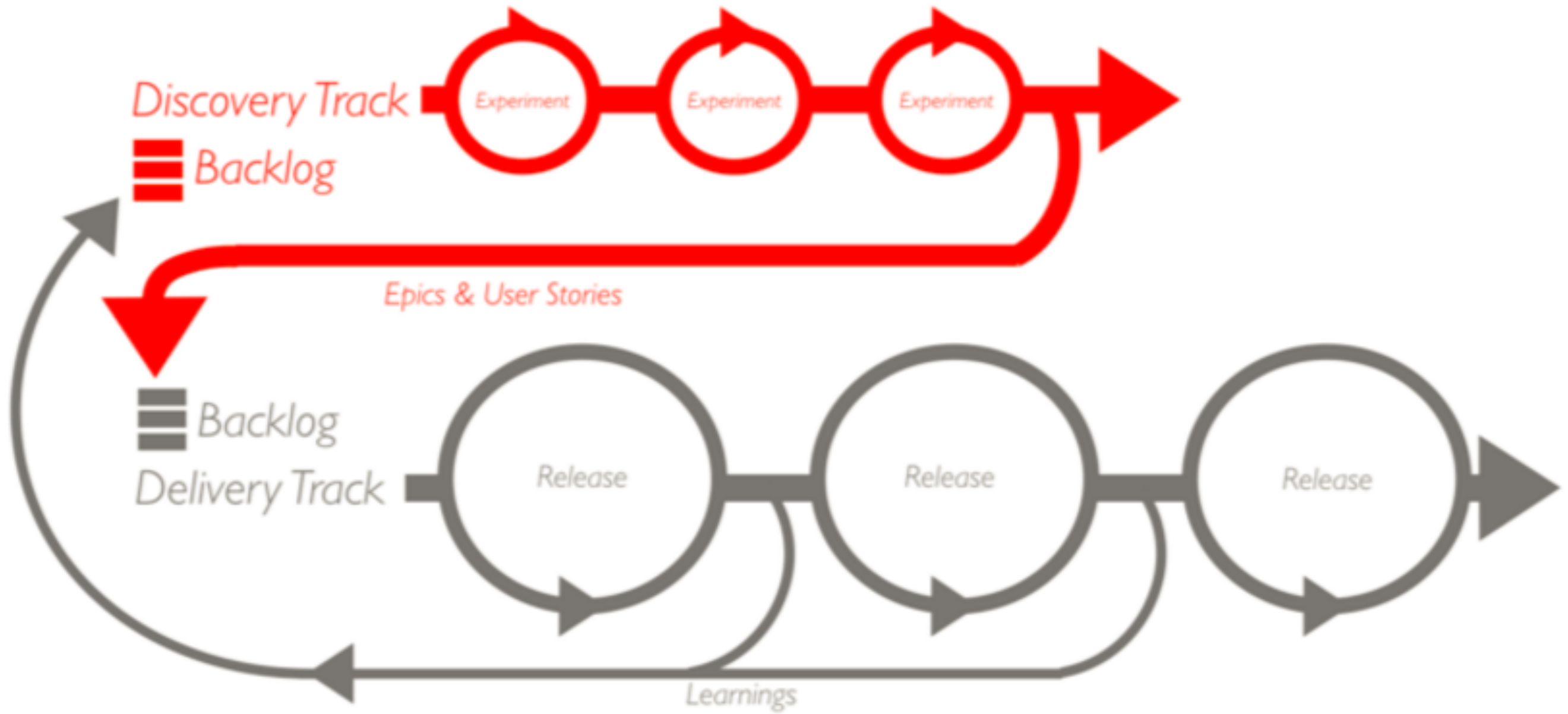
all "inventory"



© 2009, Jerry Bucknoff, PM Best Practices.

# Team?

## The Problem



## The Problem

Software is eating the world

Mutability is so powerful, but so dangerous

We are continuously adding complexity

Our understanding of risk has not caught up

Management, finance, governance ... not caught up

We eat so fast sometimes ...

**The Problem**





**The Problem**



**Summary:**

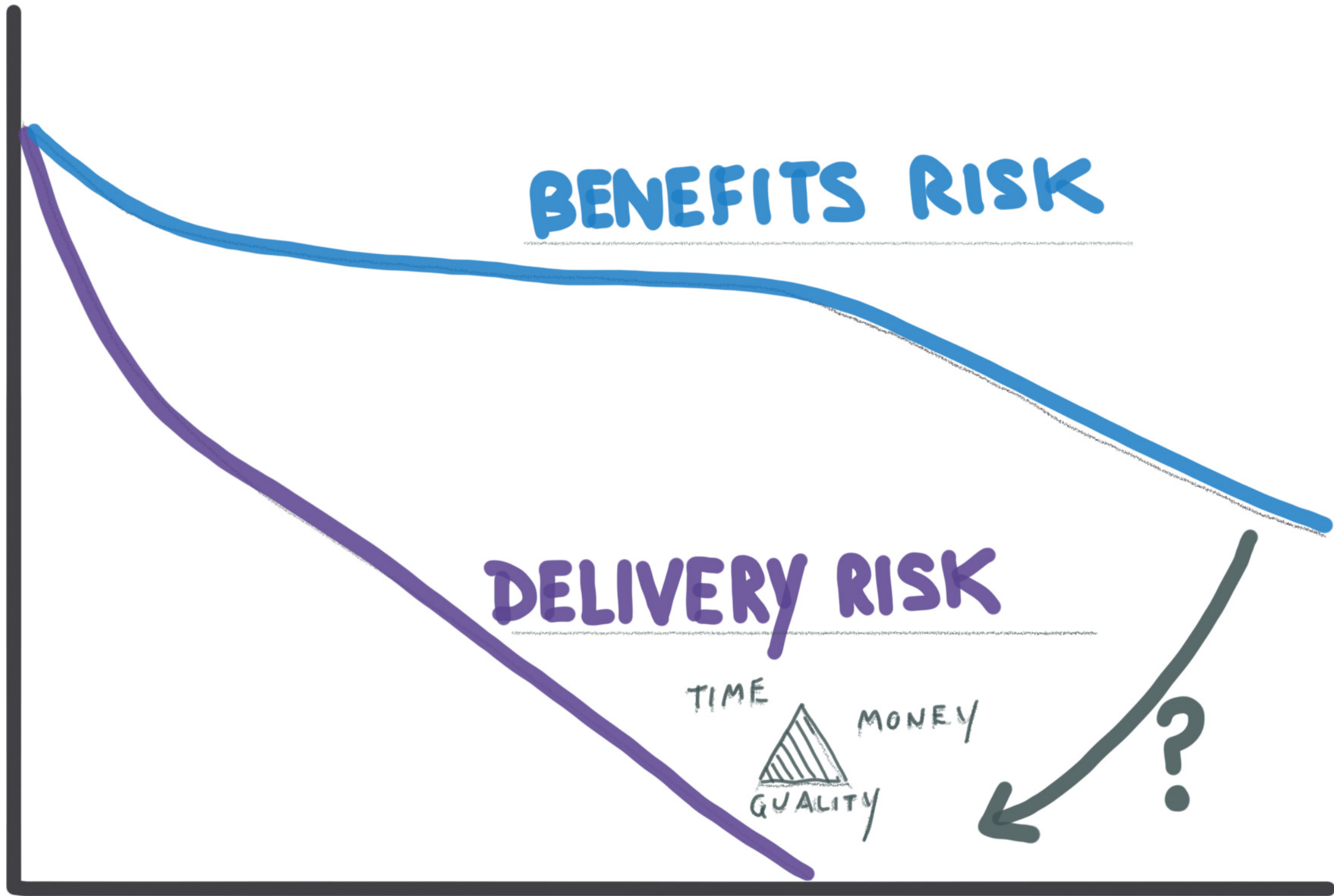
**Humans care**

**Software is powerful**

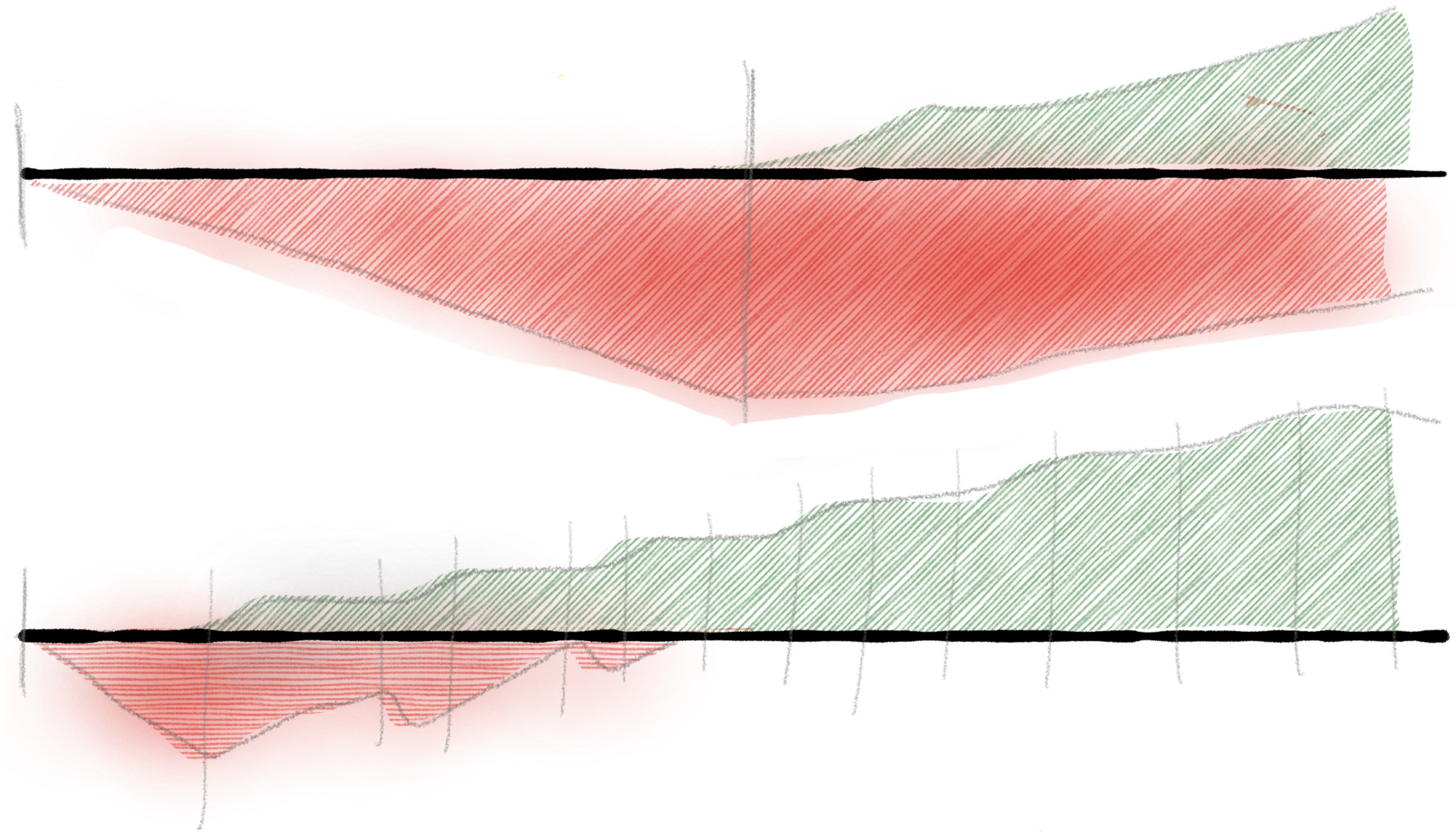
**This is \*\*\*\*ing hard**

**Opportunities**

**@johncutlefish**



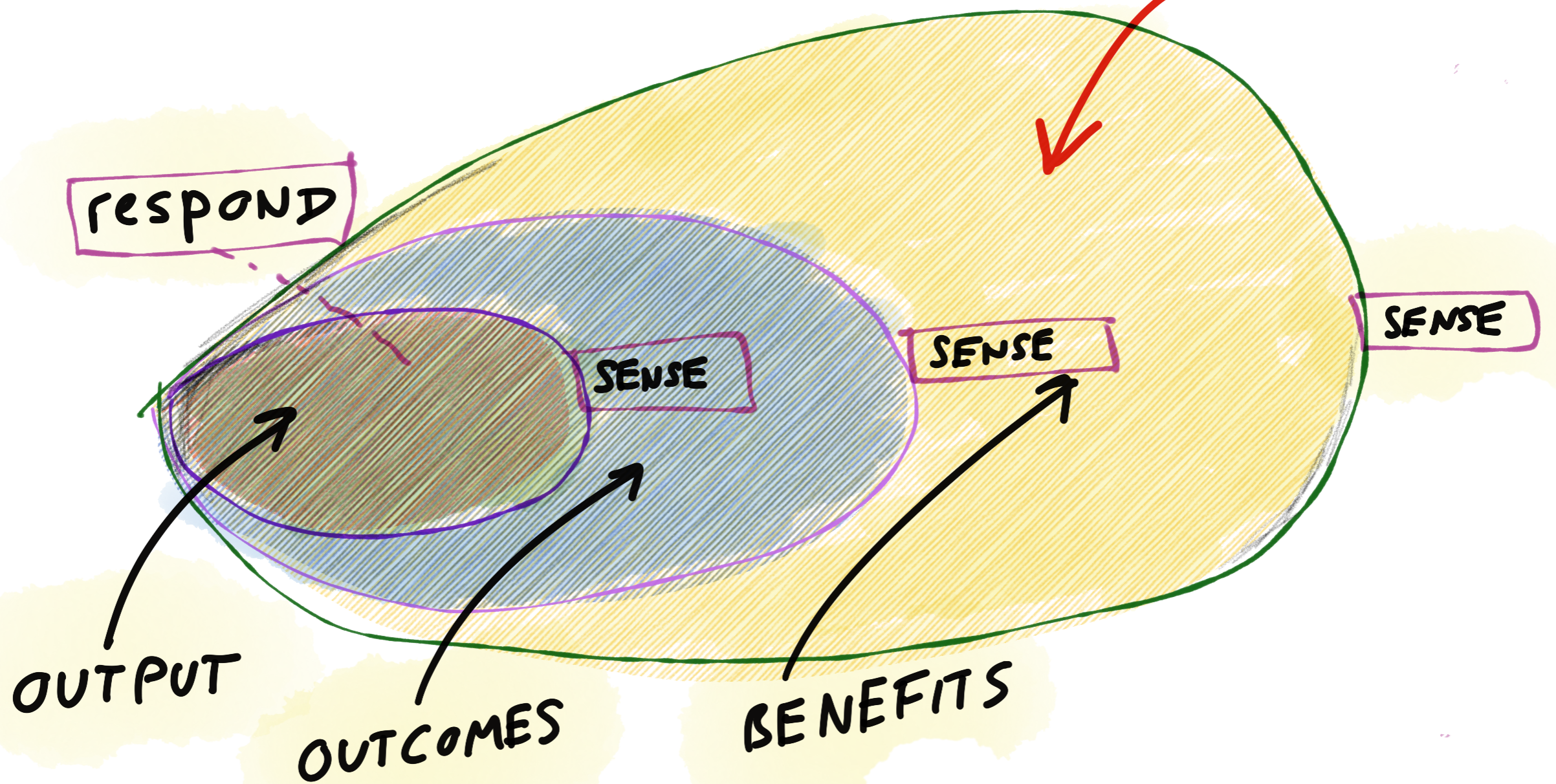
**Opportunities**



**Opportunities**



CAN WE SHORTEN THIS CYCLE?



Opportunities



**Achieve equal (or greater)  
outcomes while adding  
as little complexity as  
possible**

**Human Angle:**  
**Meaningfulness of work**  
**Retention**  
**Engagement**  
**>1.7 years**

**Show don't tell.**

**Most people haven't seen it work.**

**We believe something is possible when we see it work.**

**This is possible in your context.**

**We're the problem solvers. We understand what is possible. It is up to us.**

Shuffle teams

\$\$\$

We need ESTIMATES

40 WEEKS



MISSION



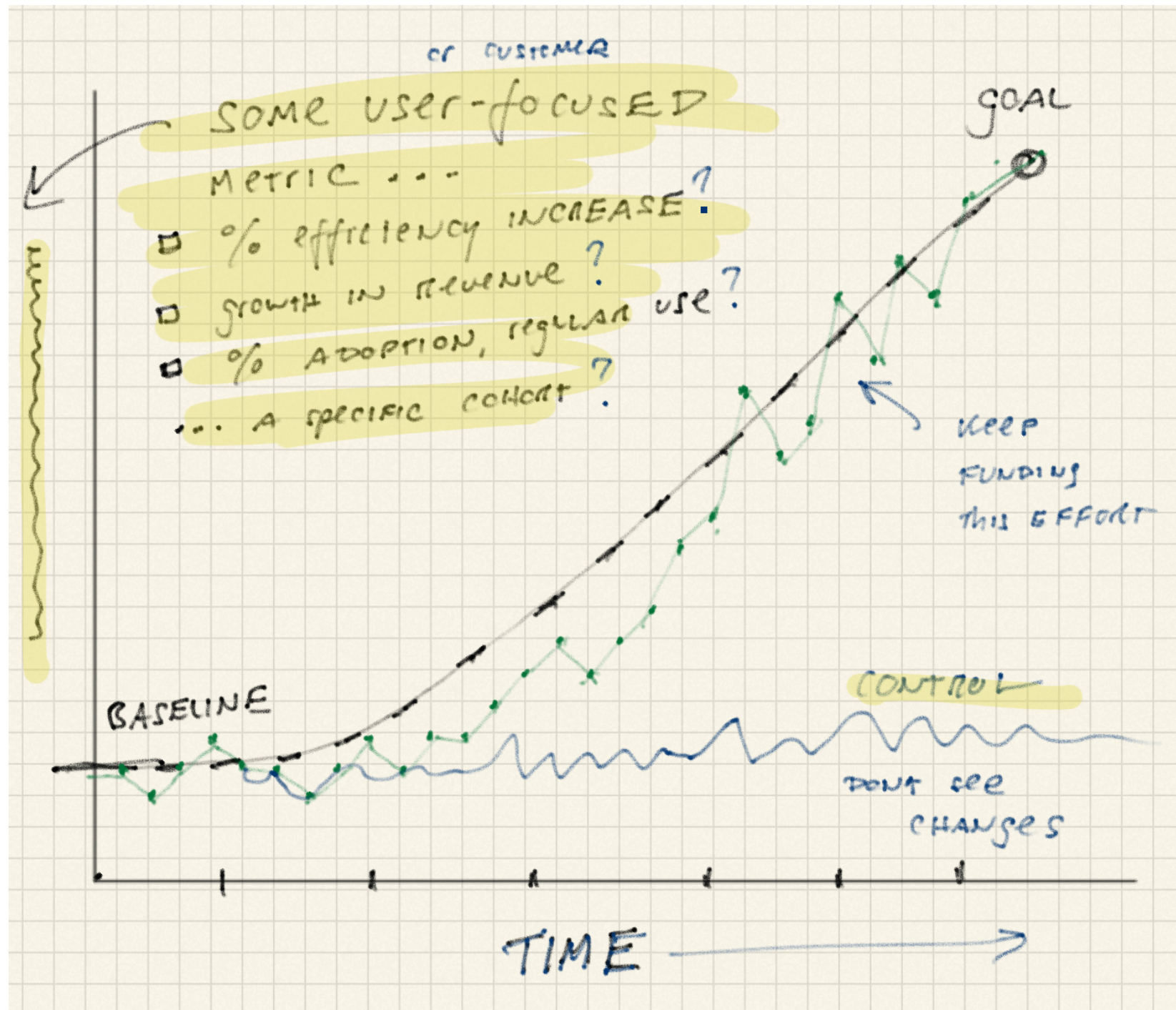
3 DAYS

(TO COLLECT MONEY)

# Letter to the Future

- Commit to present to organization in 3-6 months
- Write a brief/presentation with baseline metrics, assumptions, risks, etc.
- Add projections (dotted lines)
- Start a regular initiative “check-in” activity across the company. Present the letters to the future.

# Mission Based Teams



**Incrementally funded  
Outcome/benefit based  
Dedicated team  
Continue as needed  
Regular validation  
Vs. Control**

**Opportunities**



**Feature flags**  
**Experiment frameworks**  
**Continuous Delivery**  
**Plan for personalization**

**Opportunities**

**Cheap measurement  
Record everything...  
Pendo / Heap / MixPanel**

# De-demonize Rework

**Opportunities**

# Lunch With Customers



## Customer Research Panel



# Destroy your MVPs

**Opportunities**

# Feature Kill Quotas

**Opportunities**

**Measure for increases in  
complexity (and impacts)**

**Opportunities**

# Outcome-Centric Boards

MISSION	EXPERIMENTS	IN PROGRESS	MEASURING	REVIEW OUTCOMES	
				<u>LEARNINGS</u>  <u>DECISIONS</u> 	
				<u>LEARNINGS</u>  <u>DECISIONS</u> 	
				<u>LEARNINGS</u>  <u>DECISIONS</u> 	

Opportunities



**Outcome based roadmaps**

**Jobs-to-be-done**

**Capabilities**

**Belief networks**

**Opportunities**

**“John, you have to remember that what seems intuitive to you, may not be intuitive to other people. This isn’t an us-vs-them thing. This is an issue of awareness, trust, and change.”**

*–Friend and mentor*

**Summary:**  
**Humans care**  
**Software is powerful**  
**This is \*\*\*\*ing hard**  
**And that's why it is valuable**  
**Show don't tell**

**Make people awesome**

**Deliver value continuously**

**Make safety a prerequisite**

**Experiment & learn rapidly**