

# 0-1

The path to product-market fit, a workshop

# BY THE WAY

I have open-sourced this because most 0-1 advice is counterproductive, and it sends you straight to the pain cave. The world needs better 0-1 thinking.

If you find it useful:

- Share with other founders
- Sign up for my (free) 0-1 focused newsletter: [howtogrow.substack.com](https://howtogrow.substack.com)
- Apply to my quarterly B2B 0-1 accelerator: [robsnyder.org](https://robsnyder.org)

# Today's goal

**You don't realize just how much you can suffer in the 0-1 stage.**

**I spent 2 years in pre-PMF hell.**

**After going \$0-\$4M ARR in 2 years & taking 10+ companies from \$0-\$1M ARR, I realize that most of this suffering was unnecessary.**

**It was caused by me having the wrong approach to the 0-1 stage. Yes, despite working at McKinsey, going to HBS, reading all the books, etc.**

**Today's goal is to give you everything you need to attack 0-1 differently and avoid 12-24+ months in the pain cave.**

# What today won't help with

1. **Raising money from investors** (but it will help you build something worth investing in)
2. **How to hire & scale super big, super fast** (that happens *after* this stage)
3. **How to come up with startup ideas when you don't have any** (idk, go find a customer)

# Today's topics:

1. The pain cave & the path out of it
2. Building your case study sales pitch
3. Debugging your sales pitch & figuring out your business
4. Scheduling a bunch sales calls so you can learn fast
5. Delivering & finding PMF

***I believe this is EVERYTHING you need to know.  
Unfortunately, I wasn't taught any of this at HBS.***

# From someone just out of the pain cave:



Jean Marc Goguikian (He/Him) • 1st

10h \*\*\*

CEO @ Konkio AI | Improving Inbound Conversion Rates with AI Agents | e...

Forget about your own perspective, no customer cares about what you think they want or should want

Customers are trapped in what they know they want and expecting them to care about what you think they want or should want is like a ship captain expecting the wind to always blow in their desired direction—it just won't happen. The wind doesn't care about what the ship captain wants.

Get help from people who've been in the pre-PMF pain cave and have clawed their way out. They've exchanged years of their life to learn valuable lessons. Learn from them before attempting to climb out of your own pre-PMF pain cave.

# Topic 1: The Pain Cave

How to waste your life (or not). A practical approach to PMF.

## A friend's first 15 months in the pain cave:

- Ideation exercise -> value prop, product, weighting
- Assumptions mapping
- Validation planning
- Experimentation prioritization
- Run experiments, build MVP, nobody cares
- Pivot
- Do it right this time: 100+ customer interviews
- Force rank pain points, analyze - find #1 pain point
- Build product for 9 months alongside design partners
- Try to sell, pitch the vision & logical case for buying
- Near-zero willingness to pay
- ????

## Every smart founder's path to PMF:

**Step 1:** Come up with startup idea

**Step 2:** Suffer in the pain cave

**Step 3:** If you're lucky, eventually something magically works. "Ok, I guess we're doing that now."

**Reframe**



**what it feels like in the pain cave:**

Why isn't this working?

Am I **just not as smart** as I thought I was?

Am I **not working as hard** as I should be?

There are literally a **million things I could do**, and most sound important & right... **What actually matters?**

**I've stopped using social media**, just can't deal with seeing everyone else but me succeeding.

Am I missing some secret tactic?

How are *those* idiots successful and I'm not?

**Why aren't my marketing and sales working?** Is it tactics or strategy?

**Do I need to hire somebody?** Raise more money?

There are so many things that could be important... **how do I determine what really matters?**

Is it my product that's the problem? **Or am I just bad at sales?**

How am I supposed to run successful experiments when it seems like **even tenured academics can't run replicable social science experiments?**

Every time I'm confident, I'm wrong. **Am I uniquely bad at this?**

Do I just not have a **big enough mission / vision?** Do I have the wrong team?

I am following all the startup advice I was given... **is there something wrong with ME or with the advice?**

I don't know **how far away I am from figuring it out.**

**Can I just get acqui-hired so the pain will stop?** Can I just get a few customers so I feel like a little less of a failure?



**WHY DID I DO THIS?!**

**Reframe**

is the pain cave *avoidable*?

Reframe

# short answer: yes, I think

**Why? Because every story out of the pain cave is the same**

Mine, after 2 years in the pain cave:

- Stopped trying to come up with the perfect answer & do things the “right” way, “validate” hypotheses, etc.
- Just sold whatever someone would buy
- Figured out the business by trying to sell & deliver

What it looked like:

- I *\*was\** the product (in Gsheets) until \$100k ARR
- No website or customer login until \$1M ARR
- Just a 4-slide sales pitch until ~\$3M ARR
- I onboarded first 300 customers 1:1

# so why does the pain cave happen?

Despite (or, because of) all the startup books, frameworks, methodologies...

1. **We build things people don't want** (despite doing a bunch of customer research, interviews, discovery, etc.)
2. **We get overwhelmed trying to “solve” a million startup variables** (e.g., beachhead, niche, persona, value prop, MVP, product, vision, mission, what investors want, etc.)
3. **We work very hard on things that don't matter** (while often feeling like we're doing all the right things)


# we need to understand:

## so why does the pain cave happen?

Despite (or, because of) all the startup books, frameworks, methodologies...

1. **We build things people don't want** (despite doing a bunch of customer research, interviews, discovery, etc.)
2. **We get overwhelmed trying to "solve" a million startup variables** (e.g., beachhead, niche, persona, value prop, MVP, product, vision, mission, etc.)
3. **We work very hard on things that don't matter** (while often feeling like we're doing all the right things)

Reframe

- 
1. How to think about **"what people want"** (to pay for)
  2. A framework for the **mechanics of product-market fit** (and how all the pieces work together)
  3. A **process for navigating 0-1 and finding product-market fit** that doesn't waste our time

Reframe

**1:**  
What people want

**2:**  
The mechanics of PMF

**3:**  
A simple PMF process

# Everyone knows...

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process





why do we fail to build things people want?  
(hint: we are not dumb; we know this is the job)

**the awkward truth:**

We don't have a good way to think about what people want

# Has this ever been said?

1:

What people want

2:

The mechanics of PMF

3:

A simple PMF process

“YES! Today’s the day I get to buy NEW ENTERPRISE SOFTWARE!”

## **What about:**

“WOW! This startup’s value prop is SOOOOO exciting, I’m going to drop all of my goals and projects to buy their product!”

## **Or:**

“WOW! This startup knows my pain points and problems so well, I’m going to drop all my goals and projects to buy their software!”

**Reframe**

# Hard truths:

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

Nobody wants your product.

Nobody cares about how you think the world SHOULD work.

Nobody cares about how much customer research you did.

**And it gets worse...**

**You've found a 10/10 pain point?** Still doesn't mean there's demand.

**You've found a big problem?** Still doesn't mean there's demand.

**People are excited about your product?** Still doesn't mean there's demand.

**The only foolproof way I've found to think about demand:**

**Reframe**

# What is demand?

1:

What people want

2:

The mechanics of PMF

3:

A simple PMF process

## At the atomic level:

- One person who needs to accomplish something.
- Why she needs to accomplish this, vs everything else she could do.
- What options she considers for accomplishing this thing.

## Demand nuances:

- You have no control over demand. You don't create demand. It exists whether or not you exist.
- Demand is not just “pain points” and “problems.” Life is a series of pain points and problems you do nothing about. Demand is about ACTION, which is sometimes caused by pain points.
- Demand is not “value props” or “benefits”. Demand makes your value prop relevant (or not)
- Demand is an invisible force that feels like the customer pulling more than you pushing.
- You can't tell by looking at a product what the demand is (e.g., my AWS product)
- Not understanding demand = why we build products people don't want, why “50% of my marketing budget is wasted”... general waste in companies

**Reframe**

# Demand

What nobody focuses on, most don't realize it's a real thing

You have ZERO control over this

Exists whether or not you exist

What a person is trying to achieve

You don't create this, you FIND it & harness it

CAUSES supply to be relevant or not; is upstream of supply

Demand POV makes business simple & intuitive

PULLS your business into existence

Where you find product-market fit

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

# Supply

Where everyone spends their time (founders, tech advice, etc.)

Everything you control (product, value prop, sales process, etc)

Only exists because of you

How you help them achieve it

Your response to demand

Is only relevant if it responds to demand; downstream

Supply POV makes business complex & impossible

PUSH (uphill if no demand, downhill if demand)

Where you find pain, suffering, infinite ambiguity

# To find *your* demand...

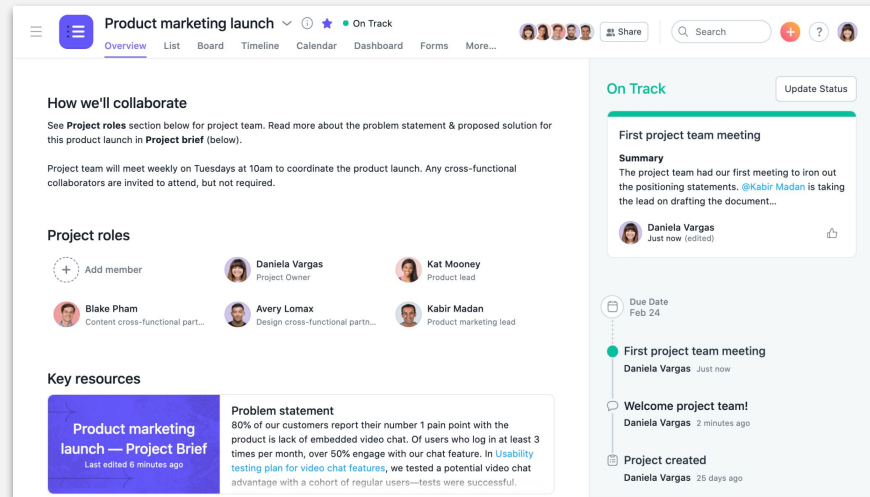
1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

## I would look into your Asana (project management software)

- (Let's pretend that it's perfectly accurate & up to date)
- You have a million projects in backlog in varying states of coherence; a million things you ideally want to do
- BUT...
  - At any one time, you can only prioritize 1 or 2 projects
  - When you prioritize a project, you “get real” about it, add more detail and consider a few approaches for HOW to accomplish it



## Given this, I need to know:

- Which project does my product fit into?
- WHEN do you prioritize this project vs. others? What CAUSES this?
- When you prioritize this project... how do you think about it? What options do you consider? How do you think about the SHAPE of the project, your goals and anti-goals?

Reframe

# my simple demand model

1:  
What people want

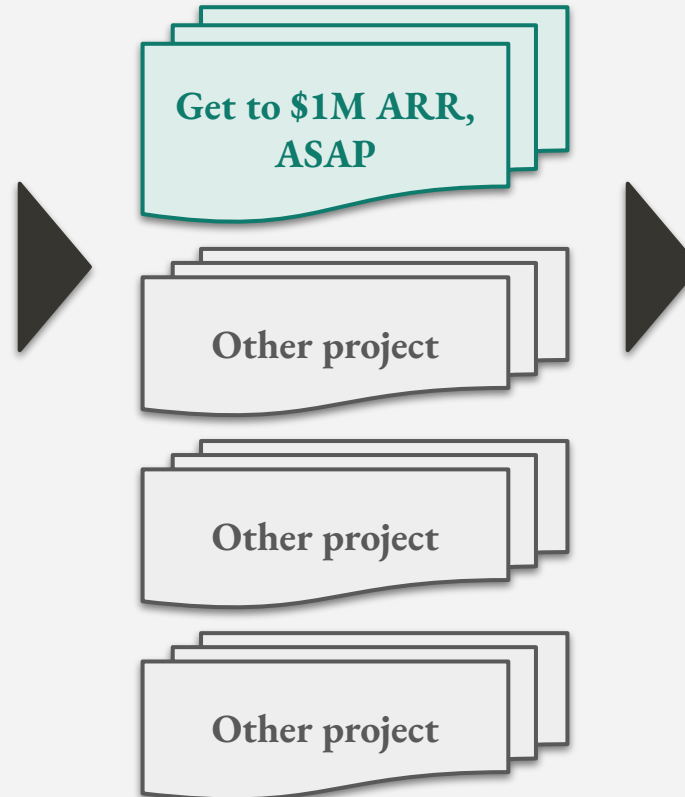
2:  
The mechanics of PMF

3:  
A simple PMF process

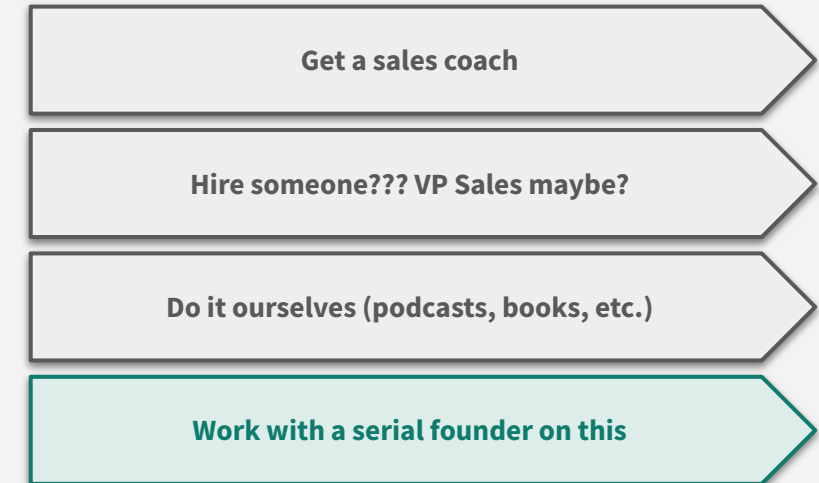
Some context / catalysts create a situation that cause you to...



...say, "screw it, **today's the day** I have to prioritize **this one project** vs. all the others I could prioritize"...



... and as you consider options for accomplishing your project, **my product fits into one of the approaches you consider.**



Reframe



# my job

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

Given ALL of this, that I have NO control over...

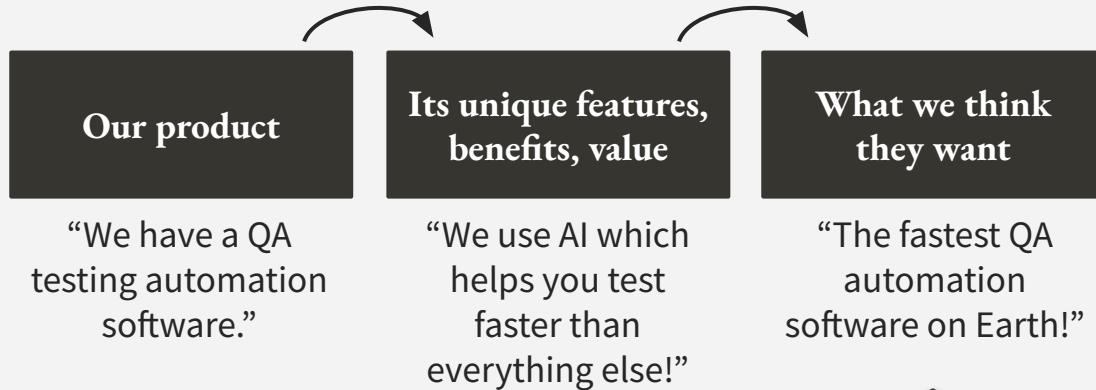
My job:

- **Understand demand** by figuring out, “when you buy, what’s your Asana project”?
- **Describe demand** in such a way that the right person says, “hell yes”
  - E.g., “wow, it’s like you’re listening in to our internal meetings!”
- **“That’s exactly what I’m trying to accomplish, and this is very helpful for me thinking about my options.”**

...so that they **PULL for more information about my supply** (how I help them accomplish this project, and what they’d buy)

Reframe

# v1, “supply-side”



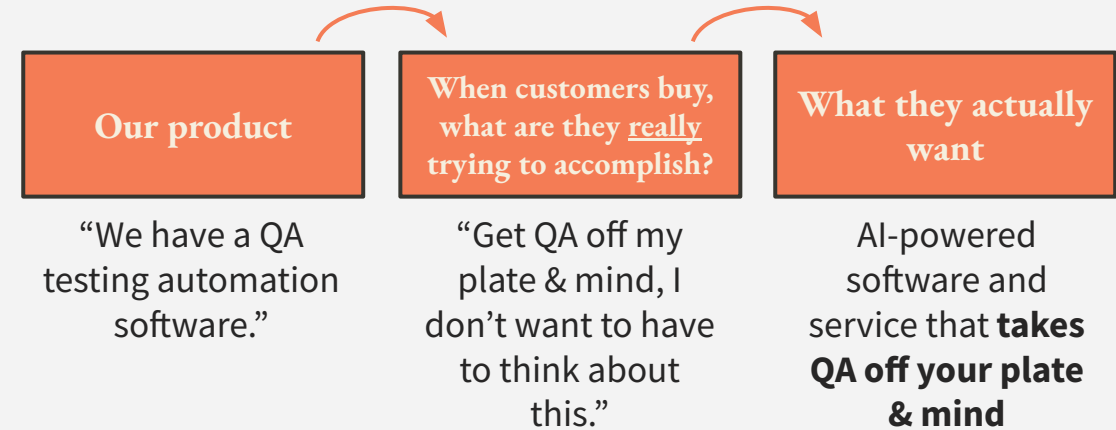
**Doesn't resonate! REALLY hard to sell, low willingness to pay, slow sales cycles, “me too” product roadmap, hard to make customers successful.**

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

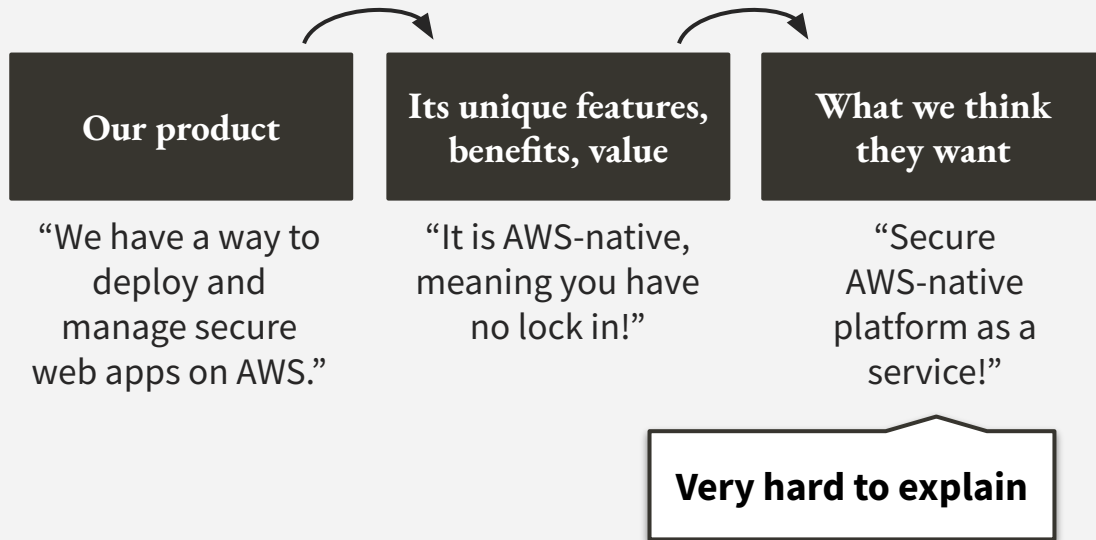
# v2, “demand-side”



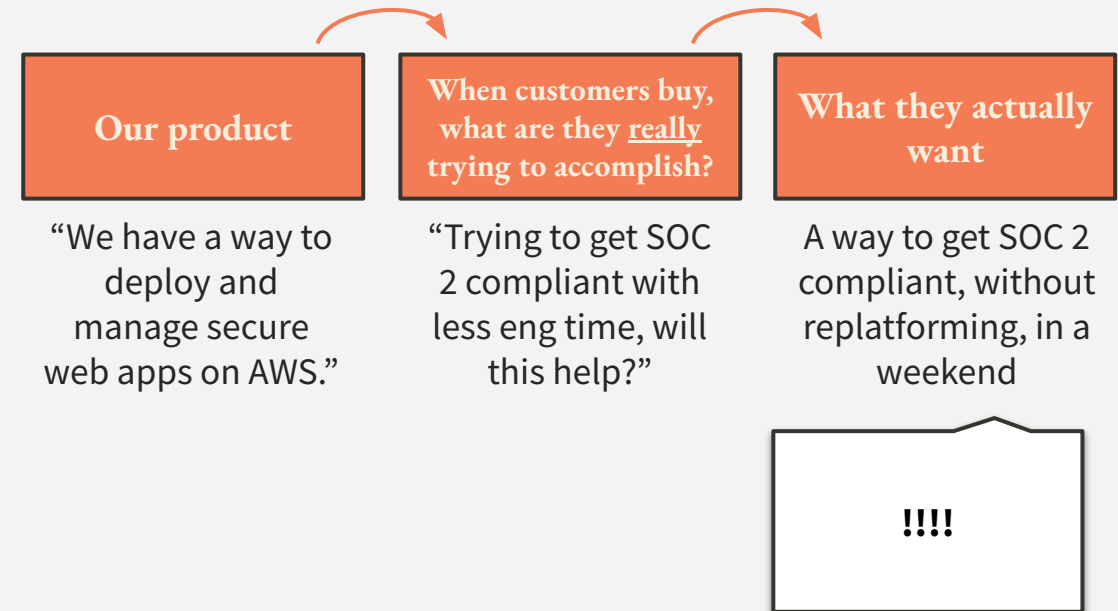
**20x+ price, faster sales cycles, happier customers, differentiated offering.**

**Reframe**

# v1, “supply-side”



# v2, “demand-side”



**Reframe**

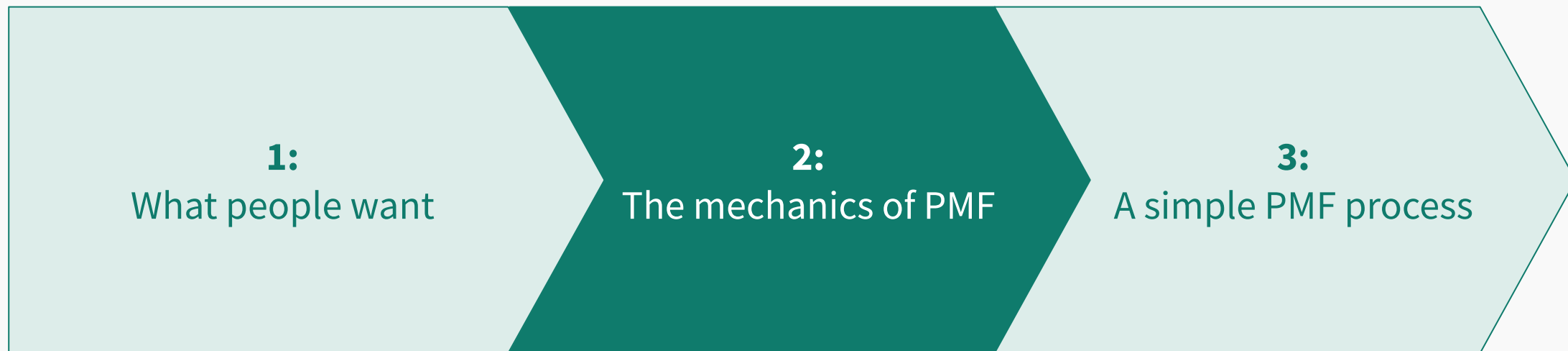
# Questions?

**1:**  
What people want

**2:**  
The mechanics of PMF

**3:**  
A simple PMF process

- What if I'm a social impact entrepreneur?
- Does this work for B2C?
- What's the shape of demand for XYZ product?
- Shouldn't I be building my vision?
- But this VC says...



# Two PMF lenses

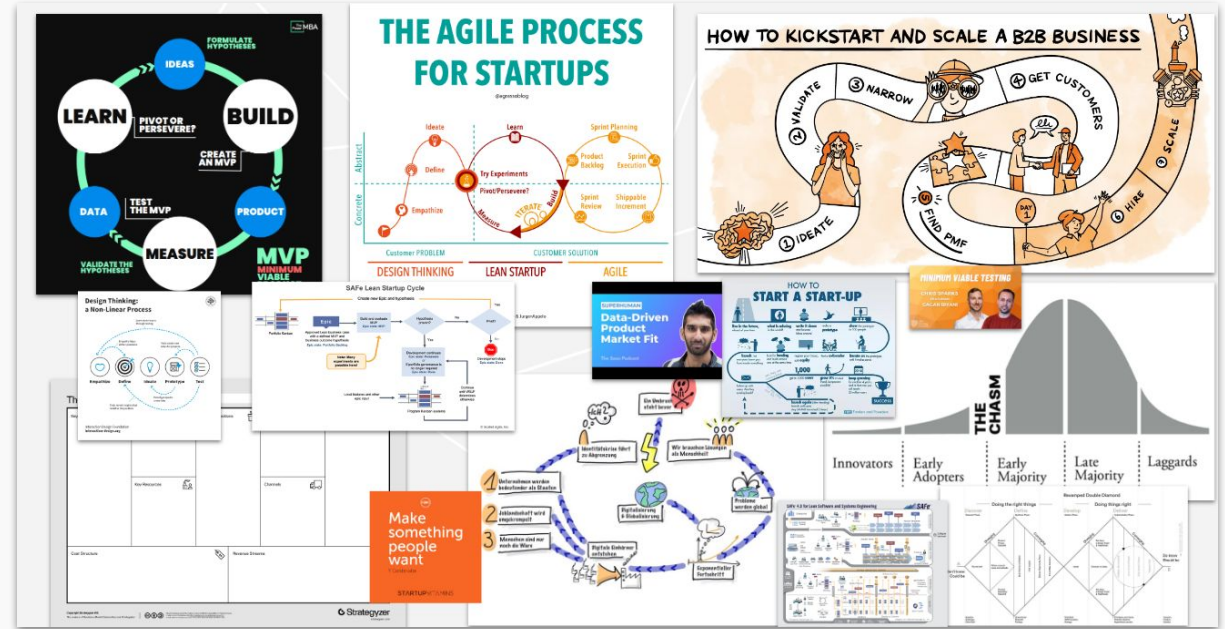
1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

## 1 - Top-down, abstract

- Niches
- Personas
- Market segments
- Hypotheses, experiments
- Metrics
- Processes
- Slogans, hot takes
- Trends



Reframe

# Two PMF lenses

1:  
What people want

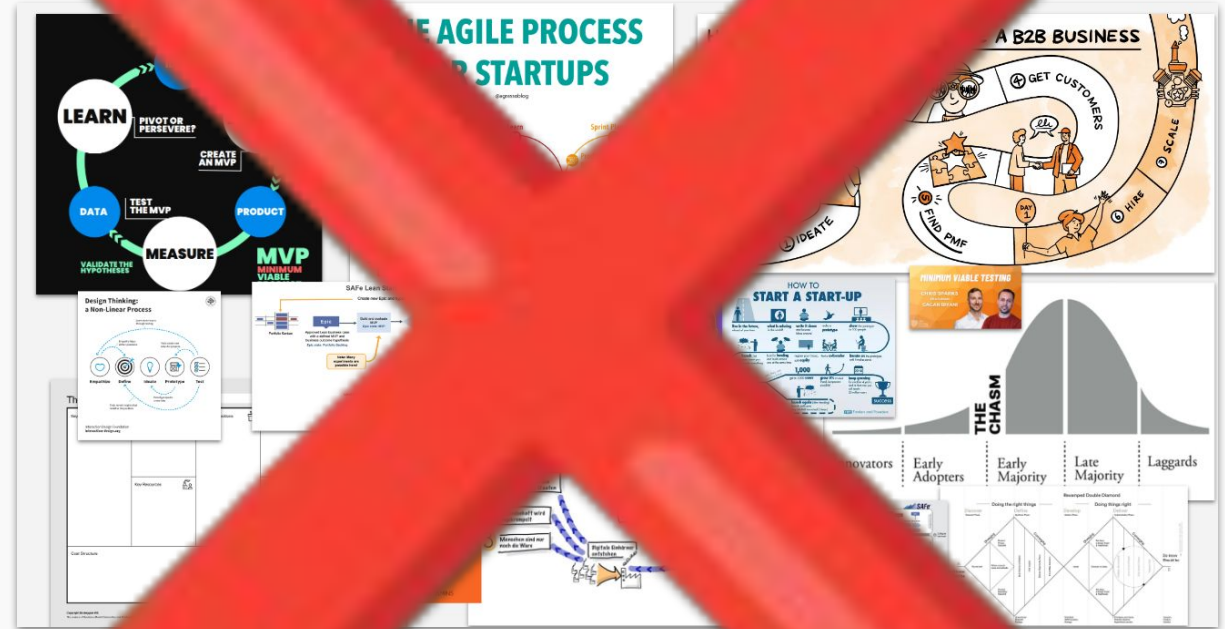
2:  
The mechanics of PMF

3:  
A simple PMF process

## 2 - Bottoms-up, concrete

Let's use the fewest fancy MBA abstract concepts possible...

...and start from ONE REAL CUSTOMER,  
& build a model from there.



Reframe

# What I've found

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

## 1 - Top-down, abstract

- Niches
- Personas
- Market segments
- Hypotheses, experiments
- Metrics
- Processes
- Slogans, hot takes
- Trends

**SOUNDS right & cool!** Works in classroom, pain cave in practice.

## 2 - Bottoms-up, concrete

Let's use the fewest fancy MBA abstract concepts possible...

...and start from ONE REAL CUSTOMER, & build a model from there.

**ACTUALLY WORKS IN PRACTICE!**



# The bottoms-up model

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

**ONE REAL PERSON:** Otherwise you create a fake person you wish existed; plus, there is never a scale transformation (we never sell to 1,000 people at a time, always 1x at a time)



## Case Study

### Demand

What Melissa was trying to accomplish

---

---

---

---

### Supply

How we helped Melissa accomplish it

---

---

---

---

**ONE CASE STUDY** vs “sell 1 product to 1 niche” - why? min(variance, complexity), max(clarity, simplicity)

**WHY ONLY ONE CASE STUDY?**  
N case studies is  $N^2$  as hard (and 1 is hard enough)

**EXACT CLONES** = min(variance) in sales / prod / delivery

**Replicate! Clone! Copy + Paste!**

### Melissa's Journey

*Prospective Customer*

*Actual Customer*

*Successful Customer*  
(Renew / repeat)

Product-market fit is measured by **RETENTION**, not (just) **GROWTH**.

Reframe

# Implications

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process



## Case Study

### Demand

What Melissa was trying to accomplish

---

---

---

---

### Supply

How we helped Melissa accomplish it

---

---

---

---



1. **ONE case study** can take you to \$1M... \$10M... even \$100M+!
2. Your business is really **a system to replicate case studies** (prospective customer -> actual customer -> successful customer)
3. **Demand determines how fast you can replicate case studies & grow.** (Demand means customers PULL for the case study / you push downhill.)
4. Your job is to **figure out your ONE replicable case study** that has intense demand! (But how?)

Reframe

# The 5 levels of PMF

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

**PMF Level 1:** Don't have a customer case study that's worth replicating

**PMF Level 2:** Have a customer case study, but can't consistently replicate it

**PMF Level 3:** Can replicate a customer case study, but it's not a "hell yes" every time.

**PMF Level 4:** Can replicate a "hell yes" customer case study, just need a growth lever

**PMF Level 5:** "Hell yes" customer case study + growth lever, holding on for dear life.

## Where are you?

- Pre-revenue: **1**
- We have a few happy customers = **1-2**
- We consistently add X customers per month the same way with the same pitch & a high conversion rate = **2-3**
- Customers consistently renew / upgrade / refer = **4**
- You'll know when you're at **5**

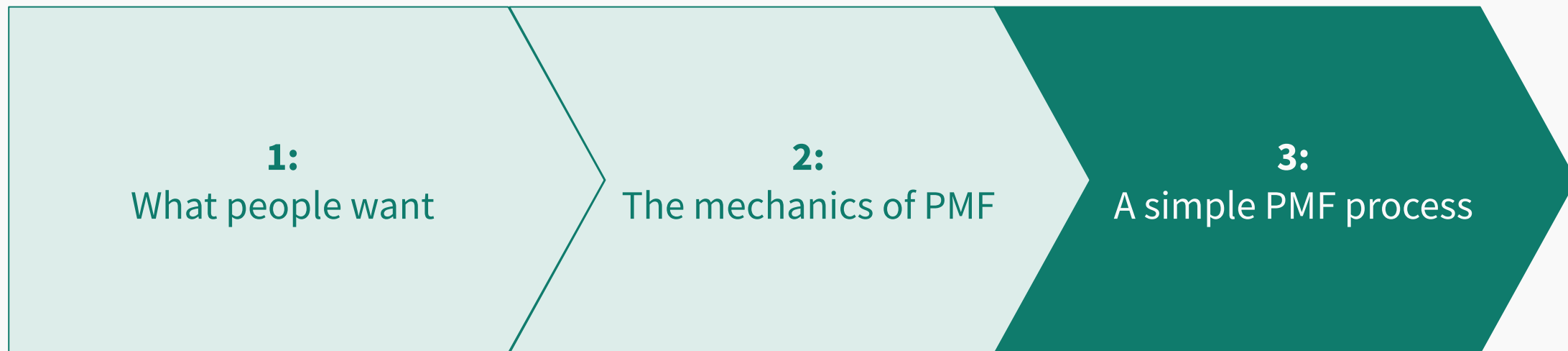
# Questions?

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

- How do I choose the right case study when I have a ton of bunch of possible case studies?
- Does this work for B2C?
- How does the case study determine my total addressable market size?
- Will I have multiple case studies in the future?
- Can I have a generic case study that kinda resonates with a lot of people?



# The big question...

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

**If your job is to figure out ONE replicable case study...**

**How do you do this?**

- Research + interviews?
- Hypotheses + experiments?
- Brainstorming + whiteboards?
- Detailed 27-step methodologies?
- Ayahuasca retreats?

**Reframe**

**My take:**

*\*Selling\* is the only way to figure this out \*for real\**

# Other smart founders agree

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process



**Parker Ence**

Cofounder, Jump

- **15 months** in the pain cave
- Then **\$0-\$100k ARR in 10 weeks**

**If I was starting all over again, after getting to that initial spark of an idea and talking to maybe 20-30 potential customers to get the lay of the land, I'd put together a few slides and start selling something deliverable.**

Here's why I think "sell to learn" makes so much sense. **If you are not asking for real money, then what you are doing is a "social" transaction. Our brain behaves in a very different way when we are in a social transaction.** We are more generous, positive, encouraging, less discerning even. **But when money is on the line, that's an "economic" transaction - the other side of our brain lights up and we start actually evaluating if there's enough value add and urgency to buy something now.** The thinking is completely different.

**For clarity, I'm not talking about "running experiments" where you simulate asking for money, like some landing page where you ask for a credit card as a test and then don't actually charge the buyer or send them a product.** That doesn't go far enough. **We needed to actually deliver something to a real buyer to see if they were still happy.** That's what closes out the feedback loop.

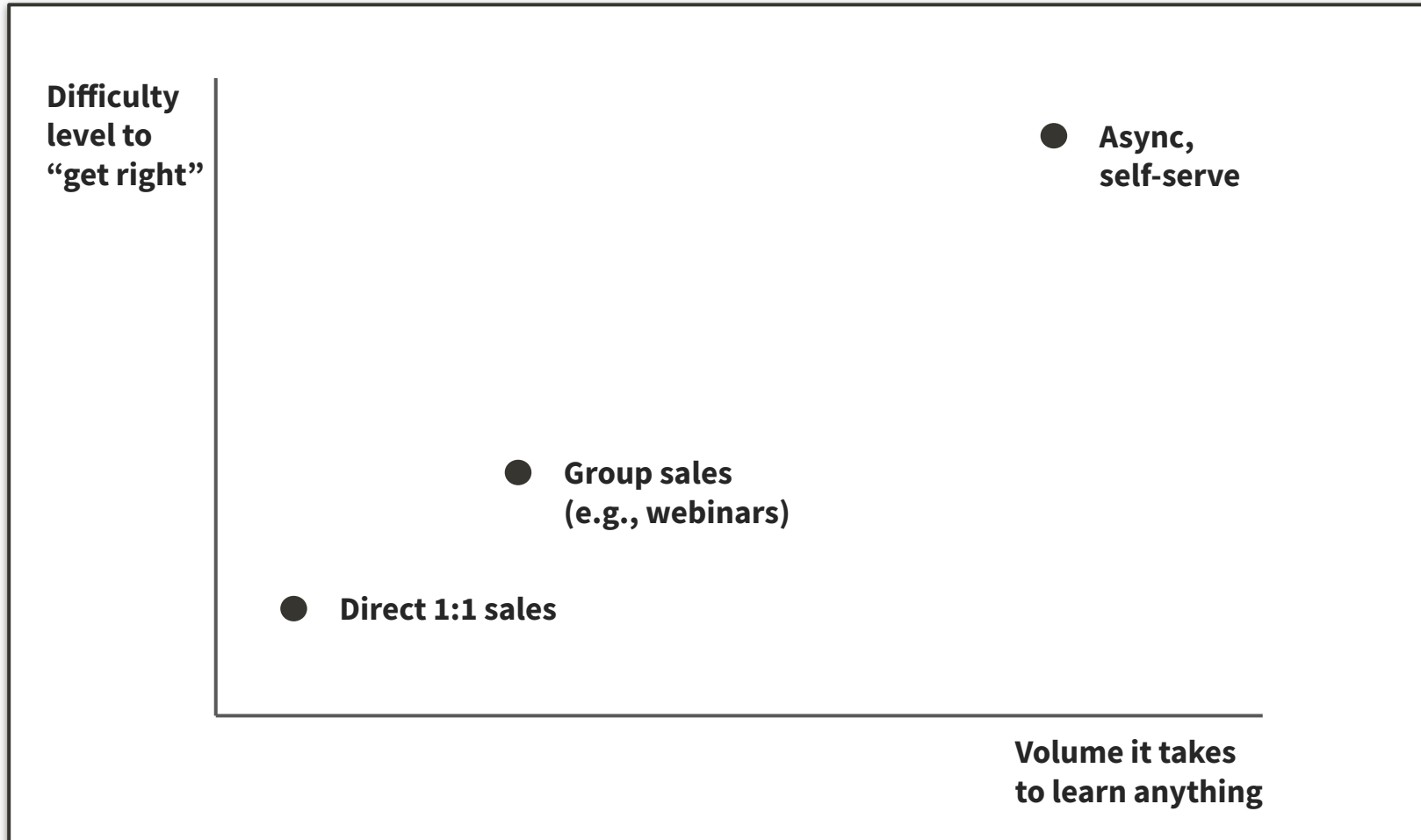


# Why start with 1:1 sales?

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process



# How to “sell to learn”

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process



## Case Study

### Demand

What Melissa was trying to accomplish

---

---

---

---

### Supply

How we helped Melissa accomplish it

---

---

---

---



**You don't have to become a pushy used-car salesperson.**

**Let your** (theoretical) **case study do the selling!**

**It's strangely simple:**

- The best way to figure out your case study is by selling
- The best way to sell is to show your case study and ask, “do you want this to be you?”

...& based on what happens in sales calls, you tweak the case study until it's a “hell yes” pre & post-sale!

**Reframe**

# Pulling it all together

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process



## Case Study

### Demand

What Melissa was trying to accomplish

---

---

---

---

### Supply

How we helped Melissa accomplish it

---

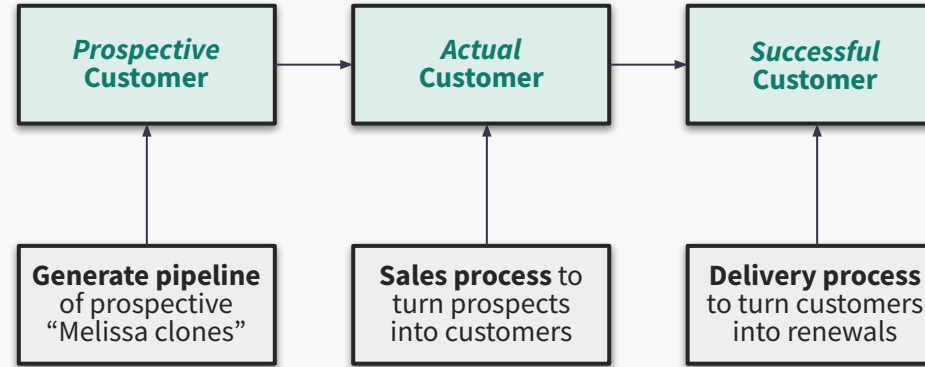
---

---

---



### Melissa's Journey



**Your Job:** Do these things to figure out your case study!

# The Path to PMF

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process


## What matters (by stage)

**Get Real (0→~10 customers)**  
Figure out retention + value

Pipeline

 **Not massively important.** Use “founder magic.”

Sales Process

 **Not massively important.** Use a theoretical case study or beg

Delivery / CS Process



**“Hell yes” (retention) by any means necessary!** (Don’t do things that scale.)

**Case Study Sales (~10→~50)**  
Nail a repeated case study

 **Not massively important.** Use “founder magic.”



**Systematize.** Use a case study; replicate the case study. (Target 60%+ “hell yes” & close on good-fit buyers.)



**Systematize.** Create a bullet train to “hell yes.” Retention drives product development.

**Find GTM Fit (50+)**  
Find a primary scalable channel



**Figure this out!** One primary scalable, cost-efficient channel.

**Debug / enhance when this breaks with scale.**

**Debug / enhance when this breaks with scale.**

**Don’t Get Distracted (100+)**  
Here there be monsters

**Do more of what works.**

**Do more of what works.**

**Do more of what works.**

**Defend the case study!**

Reframe

# Summary

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

## The path to PMF is deeply simple (but not easy):

1. Draft a theoretical case study
2. Use the case study to try to sell
3. Based on what you learn, debug the case study until you get to “hell yes” pre- & post-sale

The rest of today’s workshop is about **HOW TO DO THIS!**

Reframe

# Rest of the day:

**1:**  
What people want

**2:**  
The mechanics of PMF

**3:**  
A simple PMF process

1. Building your theoretical case study
2. Turning it into a sales pitch
3. Debugging sales pitches
4. Generating pipeline

# Questions?

1:  
What people want

2:  
The mechanics of PMF

3:  
A simple PMF process

- What if I don't want to sell? Is there another way?
- How do I sell before I have a product? Isn't that bad?
- How do I sell when I'm not sure what I'm selling?
- What if I think sales is slimy and gross?
- Shouldn't the product sell itself?
- What if I want to do self-serve?
- What if I am B2C, low price?

**Reframe**

## Topic 2: The Case Study

A very, very, very simple framework.



# The case study is super important



## Case Study

### Demand

What Melissa was trying to accomplish

---

---

---

---

### Supply

How we helped Melissa accomplish it

---

---

---

---



## It determines:

- Who you compete with
- What pricing makes sense
- Who your target customer is
- What your product needs to be
- What customers expect when they buy
- How big your addressable market is (for now)
- How fast you can grow (how much demand is there)

*...and basically everything else!*

# How do we write one of these?



## Case Study

### Demand

What Melissa was trying to accomplish

---

---

---

---

### Supply

How we helped Melissa accomplish it

---

---

---

---



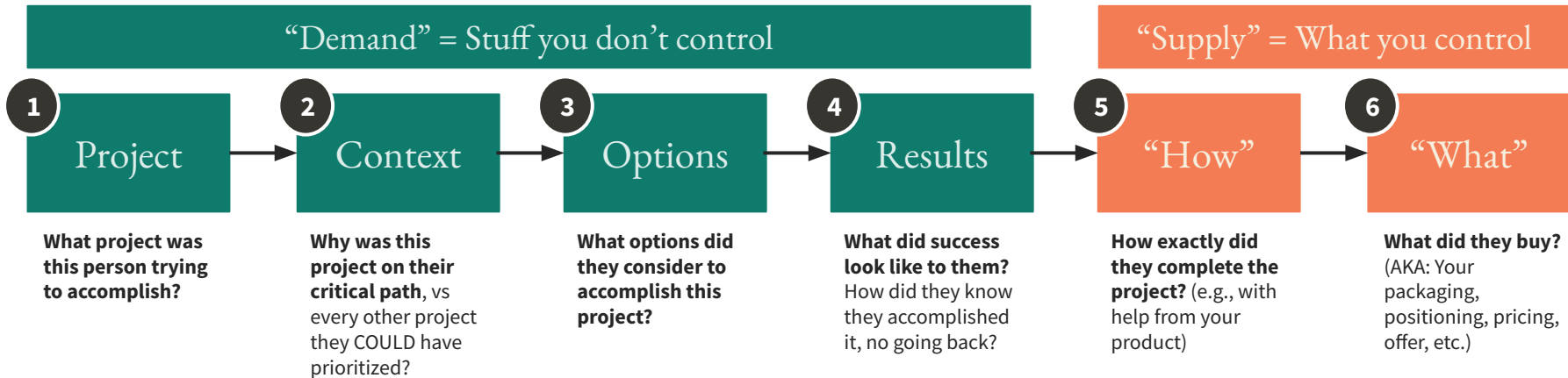
## Requirements:

1. Must make sense to our prospective customers
2. Must get them to want to buy / say “hell yes”!
3. Must be easy for you to change / debug

# A case study is a story in 6 parts



## Case Study



# An example



## Project

CTO wanted to **get QA off their plate & off their mind**, so they can ship faster with confidence

**Project:** Think of what's on their Asana board

## Context

**Are struggling with quality**, have tried simple tools & realize how much work it will take to do this

**Context:** Why is this prioritized, vs. everything else they COULD do?

## Options

Consider hiring a dedicated QA team, outsourcing, or leveraging AI

**Options:** When this is prioritized, what options do they consider to get the project done?

## Results

Using AI, they got QA under control in a few weeks, for ½ the cost of an in-house hire

**Results:** When they choose the option that includes you, what does “success” look like?

## “How”

They did this using QA AI agents, managed by a human QA architect (aka: us)

**How:** What was their path to completing the project, and how did you fit in?

## “What”

We build & run this for \$x / month, with a 1-month money-back guarantee

**What:** What exactly did they buy, and how did they justify the cost?

Reframe

# Exercise: You have 30 mins

Go to [robsnyder.org/harvard](https://robsnyder.org/harvard) -> assignment 1!

Demand (Customer POV)						Supply (Customer POV)			
Project	Who	When/Why	Options	Success	Path to success	How MuukTest helped	Packaging ("Project Plan")	Product positioning	Pricing vs. Value
Help me automate QA fast, without needing to learn to code	QAs at startups	Starting an automation project, or struggling to succeed using other tools	- Learn to code - Use a low-code / no-code tool	I (one QA) was able to automate & maintain our entire automated test suite, in just a few months	- Didn't learn code - Created automated tests while I manually tested (while doing my main job) - Got notified whenever a test broke or failed, and could quickly fix it	- Their tool allowed me to quickly create tests - And maintain tests - Without having to learn to code, or shirk my main duties	- Get trained in how to use a testing copilot - Get support whenever you need it - Get reports on your progress you can use to share with execs	The only no-code/low-code QA copilot, powered by AI	\$200 / user / month  (like making me 10x more productive)

## Fill out your case study in this spreadsheet:

1. Duplicate the [TEMPLATE] tab - rename the tab with your startup's name
2. Write your case study on one line
  - a. Write it with ONE REAL PERSON in mind!
  - b. If you have a buyer vs. a user, choose the buyer
3. If you have multiple different ideas, or multiple different potential people to write case studies about, use multiple rows (don't try to squish them all into one case study)

**We will come back & give feedback...** then turn this into a sales pitch!

## Topic 3: The Sales Pitch

A very, very, very simple approach.

# Remember this?



## Project

CTO wanted to **get QA off their plate & off their mind**, so they can ship faster with confidence

**Project:** Think of what's on their Asana board

## Context

**Are struggling with quality**, have tried simple tools & realize how much work it will take to do this

**Context:** Why is this prioritized, vs. everything else they COULD do?

## Options

Consider hiring a dedicated QA team, outsourcing, or leveraging AI

**Options:** When this is prioritized, what options do they consider to get the project done?

## Results

Using AI, they got QA under control in a few weeks, for ½ the cost of an in-house hire

**Results:** When they choose the option that includes you, what does “success” look like?

## “How”

They did this using QA AI agents, managed by a human QA architect (aka: us)

**How:** What was their path to completing the project, and how did you fit in?

## “What”

We build & run this for \$x / month, with a 1-month money-back guarantee

**What:** What exactly did they buy, and how did they justify the cost?

Reframe

# It really is all you need for early-stage “finding PMF” sales



## Project

Typically, people think about QA automation with 1 of 3 main priorities...

- Testing is the bottleneck for our development process, and we need to fix this to hit our growth goals.
- Reg. are getting to production, but we need to speed up that cycle.
- Our current approach to testing is over-engineering our tests, and not scalable.

Still, everyone's common goal: spend the least amount of money & effort on QA as possible, while ensuring quality, so we can focus our time, money, effort on our roadmap.

Our best customers come to us with 1 of 3 projects... **which is yours?**

## Context

When testing is your bottleneck...

"I need an automated regression test suite that covers 80-90% of our software. It needs to be integrated into our development process and our QA. We still do exploratory manual testing for our current business, but we'll have to do regression testing to have the new features appear all our existing software. This will enable us to deploy without having to wait hours, days, or weeks on testing."

Scott, CTO, Fast Growth Startup

Here's why this project was on someone's critical path... **what's your "why"?**

## Options

Options when testing is your bottleneck

- 01** No code / AI tools  
No need for coding or QA Engineers. Easy to learn and use. Supports all major frameworks.
- 02** QA Engineers  
Need to learn and use. Supports all major frameworks. Easy to learn and use. Supports all major frameworks.
- 03** Automation as a Service  
Pay for the software you need. No need for coding or QA Engineers. Easy to learn and use. Supports all major frameworks.

Here are the options they considered... **what options are you considering?**

## Results

QA-as-a-Service is the fastest way to build an automated test suite that unblocks development

Before MuukTest: 100% manual testing. 100% of test cases were manual. 100% of test cases were manual. 100% of test cases were manual.

After MuukTest: 80% automated testing. 80% of test cases were automated. 80% of test cases were automated. 80% of test cases were automated.

Here's their results, and why they considered it a success... **are these the results you're after?**

## "How"

MuukTest = Software, Service, Methodology

1. You plug into your platform and developer's.
2. QA Expert + AI software design and development.
3. You get full test automation with continuous verification.

Here's HOW we were able to help them achieve these results... **any questions?**

## "What"

Medium-Sized Application: Up to 300 Test Cases

90-95% coverage in 3 months, 100% peace of mind

1-Month POC: 100% coverage, 100% peace of mind. 1-Month POC: 100% coverage, 100% peace of mind. 1-Month POC: 100% coverage, 100% peace of mind.

Here's what they bought... **should we send the contract right now or what?**



# The early-stage minimum viable sales process

For B2B; B2C you can often just sell in convo 1

## Meeting 1: Case Study Pitch

***purpose:***

*identify what they need to accomplish  
& how they think about their options &  
timeline*

Use the “case study” demo (next section)... even if you don’t have case studies yet

## Meeting 2: Deep-Dive

***purpose:***

*help them understand what  
implementation would look like, &  
make the case*

Usually:

- Recap + why this, why now?
- How it actually works / key workflows
- Implementation plan
- Process to get this done internally (e.g., schedule other demos)

## Meeting 3: Decision

***purpose:***

*get it done!*

Usually:

- Recap
- Remaining questions / objections?
- Review contract / order form
- Schedule onboarding, use as forcing function

# The case study sales pitch

The demo **ISN'T** “you showing your product to a prospect & trying to persuade them to buy it.”

The demo: You sitting on the same side of the table as your prospect, exploring \*a similar company's case study\* and letting that story persuade your prospect to buy.

When you understand this, you'll sell without feeling salesy, and you'll figure out how to debug the case study on your path to PMF.

# How it feels to your prospective customer

DEMAND

“What caused you to take the meeting? What’s going on in your world?”

“Not sure if I can help you... Usually I hear from people in your situation they have this one MASSIVE challenge / goal... sound familiar? How are you thinking about it?”

DEMAND

“This is how John (in your exact situation) thought about his options... sound familiar? How are you thinking about YOUR options?”

DEMAND

DEMAND

“John chose X option, and here’s what he accomplished... sound valuable?”

“Here’s how we helped John... worth seeing if we can help you too?”

SUPPLY

Reframe

# Structure

- 1. Intro**
- 2. Case study slide deck**
  - a. Project
  - b. Context
  - c. Options
  - d. Results
  - e. How
  - f. What
- 3. Next steps**

# Goal: “SOC 2 ready” fast, without wasting eng effort

When CTOs talk with us, we typically hear...

“We need to get to our SOC 2 audit, **we already bought Vanta & have our compliance checklist.**”

“Didn’t realize how **much eng work it was going to take just to get Vanta’s lights to turn green** so we can get this SOC 2 report. It’s not even pass/fail!”

“**Need this done yesterday**, going through vendor reviews.”

---

# Options CTOs typically consider at this stage

1

## **Pause on SOC 2**

Good if SOC 2 is not absolutely critical

2

## **Refactor everything**

Good if you have time & \$\$\$  
(2-3 months + consultants +  
all-hands on deck)

3

## **Add AWS Security Layers**

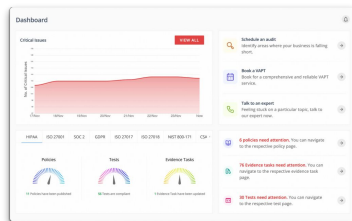
Good if you just need it done,  
have at least partial AWS stack

---

# From zero to secure & “SOC 2 ready” in a weekend

Using AWS security layers

Not SOC 2-ready stack

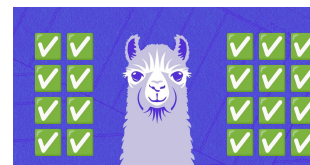


SOC 2-ready stack



## Add security layers to AWS:

- One layer for each deployment (dev/QA/prod)
- Plus one layer for each repository (e.g., front-end, back-end)



“Saved me so much time worrying about turning Vanta’s lights green & keeping them green as we keep building.” - Matt, CTO

# How to get “SOC 2 ready” in a weekend

Using AWS Security Layers, by Waffle

## Step 1:

### Link your AWS + Repos

 30 mins w/ 1:1 support

- Github / Gitlab + AWS account
- Use our setup wizard + get 1:1 support

## Step 2:

### 1-Click Security Layers

 1 min

- Wrap your deployments & services with security layers & push into your AWS account
- **Turns ~all of Vanta’s lights green** (product, not policy stuff)
- All done in 1 click

## Step 3:

### Stay Secure

 Ongoing, forever

- As you add services, build more product, change things, add & update security layers so Vanta’s lights stay green
  - All done in 1 click
-



# “SOC 2 Ready” in a weekend

## What you get:

- **SOC 2 AWS wrappers implemented for you** to turn Vanta’s lights green, in one weekend
- **Plus, all remaining SOC 2 green lights from our application** (including permissioning, infra change mgmt.)
- **All based on open-source, AWS resources** that you can use to build new services, securely, fast in the future (or swap out for different services)

## How CTOs think about price:

**Do this in-house:** 2-3+ months to build (\$50k+), or hire someone (\$150k+)

## What Waffle costs:

- **\$1,000 done-for-you** one-time cost
- **\$250 / month**

(Or, DIY for free using [open-source library](#))

# Exercise: You have 30 mins

Go to [robsnyder.org/harvard](https://robsnyder.org/harvard) -> assignment 2!

## Do your case study slide deck:

1. Pick your best case study
2. Duplicate the slide deck template
3. Create your sales pitch, KEEP IT SIMPLE

**We will come back & give feedback...** then listen to one of these sales demos in action!

Share with **rssnyder13@gmail.com**

## Topic 4: Debugging Sales Calls

How to look dumb today, so you can look smart tomorrow.

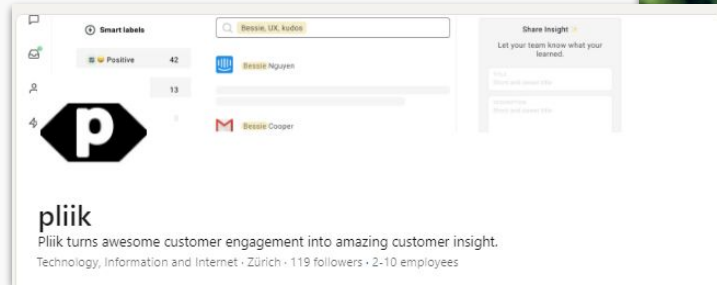
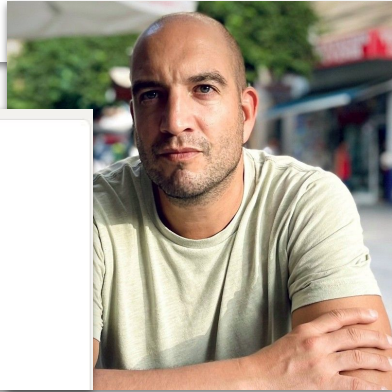
# How you tend to “figure it out”

## The Path to Product-Market Fit

### A simple unfolding story

a strange thing happened on the way to PMF...

MAY 17, 2024



- 3 “hell yes” by sales meeting 27
- These 3 were fundamentally different than the other 24
- Could have just narrowed his targeting to these kinds of people... **but went a step further:**
- Redesigned Pliik’s case study around these 3 people & their situation

# Your case study evolves to “hell yes.” This = unfolding

Why did we change the case study?	Demand (Customer POV)				Supply (Customer POV)					
Unfolding Trigger(s)	Project	Who	When/Why	Options	Success	Path to success	How this company helped me	Packaging ("Project Plan")	Product positioning	Pricing vs. Value
1 [founding direction]  [Problem: No understanding of demand]	Take employee scheduling off my plate, let employees schedule themselves	Any hourly workplace	?	- Do what I'm currently doing - Do this	I don't have to bother with scheduling anymore	- Rip out my current software - Implement this software	- They have awesome software	- Buy this software, good luck using it	Fully automated employee scheduling software	?
2 - Hard to sell (can't tell if we're bad at sales, or if nobody wants the product) - Beta users REALLY use one feature - the last-minute shift-filling tool  [Problem: Unfolding without understanding demand]	Fill 100% of my last-minute open shifts with my own employees	Any hourly workplace	Pain points around last-minute open shifts (10/10 pain point for everyone)	- Do what I'm currently doing - Do this	I press a button and fill every shift, no matter how last-minute my needs are (e.g., last min callouts)	- Don't rip out my current software - Implement this software	- Their tool calls and texts my employees in order, until someone fills the shift	- A very anxious cofounder will try to make this work	On-demand shift-filler for hourly workplaces	?
3 - Nobody fills shifts when the managers try it the first time - Managers don't use it unless we force them to - maybe not big enough of a pain? - Who has bigger pain? Unionized workplaces, where they have to follow certain shift allocation rules or they get sued  [Problem: Unfolding without understanding demand]	Fill 100% of my last-minute open shifts with my own employees	Unionized workplaces with open shift rules in their collective bargaining agreements	Pain points around last-minute open shifts (10/10 pain point for everyone)	- Do what I'm currently doing - Do this	- I press a button and fill every shift, no matter how last-minute my needs are (e.g., last min callouts) - in line with our collective bargaining agreement  - I get a report so if the union gets mad, I can prove I followed the rules	- Don't rip out my current software - Implement this software	- Their tool calls and texts my employees in order prescribed by the CBA, until someone fills the shift	- They write an algorithm based on our union contract - Maybe they integrate with our current scheduling system / HR system? - They give us a tool to contact employees for last-minute shifts	On-demand shift-filler for unionized workplaces	\$500 / workplace / month
4 - Unionized workplaces move so fucking slow - Where do they have the most call-outs? - How can we add "supply" to the marketplace and actually fucking fill a shift now and again? (Insight from one successful janitorial customer - used former employees / "on-demand bench")  [Problem: Unfolding without understanding demand]	Fill 100% of my last-minute open shifts with my on-demand bench of current and former employees	Janitorial firms that have a lot of absences	Pain points around last-minute open shifts (10/10 pain point for everyone)	- Do what I'm currently doing - Do this	- I press a button and fill every shift, no matter how last-minute my needs are (e.g., last min callouts)  - I keep a bench of on-demand workers who previously worked here who are trained	- Don't rip out my current software - Implement this software	- Their tool calls and texts my employees in order, until someone fills the shift	- They upload our current employees - They contact our past employees to add them to the bench - They give us a tool to contact them	On-demand shift filler, with your past employees as a "bench"	\$5 / shift filled
5 - Pandemic, all ~5 janitorial firm customers churn - Still fucking hard to fill last-minute shifts, even contacting former employees - One fast-food customer calls and says, "hey this shift filling thing sucks, but can you help us hire employees? we're short staffed."	Hire employees from your former employee network	Any hourly workplace, maybe fast food companies?	Really fucking hard to hire right now, everyone is short staffed	- Advertise on Indeed - Buy yard signs - Try something new?	- They send me candidates from my former employees  - I hire them and am a little less short-staffed	- Text and email former employees - When someone is interested, we follow our normal hiring process	- They downloaded people from my HR system - They texted & emailed these former employees - When someone was interested, they sent us a notification to reach out to them	- Text and email all my former employees, every month, to drive additional applicant flow - I can customize the messaging every month	Solution to re-engage your former employees	\$100 / month / restaurant
6 - Fear that nobody is going to respond in month 2 - District manager at customer 1 saying, "wait, why don't you text our past applicants too? There's like 10x the number of them!"	Hire employees from your former employees and past applicants - the people you already know	Any hourly workplace, maybe fast food companies?	Really fucking hard to hire right now, everyone is short staffed	- Advertise on Indeed - Buy yard signs - Try something new?	- They send me candidates from my former employees and past applicants  - I hire them and am a little less short-staffed	- Text and email former employees and past applicants - When someone is interested, we follow our normal hiring process	- They downloaded people from my HR system and applicant tracking system - They texted & emailed these former employees and past applicants - When someone was interested, they sent us a notification to reach out to them	- Text and email all my former employees and past applicants, every month, to drive additional applicant flow - I can customize the messaging every month	Solution to re-engage your former employees and past applicants	\$100 / month / restaurant

Reframe

# Unfolding vs. pivoting

**Pivoting is a mechanical direction-change.**

- Implication: Go back to zero, to first principles
- Bad idea, leads to “pivot hell” & extra time in the pain cave

**Unfolding is an evolution that represents *compounding* knowledge.**

- GIVEN our knowledge & product, what is the EVOLUTION that builds on ALL OUR ADVANTAGES & EVERYTHING WE’VE LEARNED and makes us stronger, simpler, more coherent, more in-line with demand?

# Unfolding implications

1. **The case study determines your current market size & shape**
2. As you change your case study to find demand, your market size & shape change (as does your product)
3. You can only figure out what the “right” case study is by selling & debugging the case study to get to “pull”; this happens forever
4. **So, you really can't predict much.** Embrace this, and ONLY focus on getting the case study right. In time, it will change, and what might seem like a \$10M business now will unfold into a \$100M+ business.

# Five case study unfolding patterns

1

## Reframe to true demand

### **Example:**

From “Simplify AWS” to “Build SOC 2 Compliant Products”

### **Do this when:**

*You find out why people REALLY buy - what demand REALLY is. Or, WHO has intense demand.*

2

## Intensify demand

### **Example:**

From “Build SOC 2 Compliant Products” to “SOC 2 in 1 Weekend”

### **Do this when:**

*You feel slight pull, and know there’s something deeper.*

3

## Simplify / clarify the story

### **Example:**

From “Perceived as an alternative to Vanta” to “partnering with Vanta”

### **Do this when:**

*You feel pull earlier in the case study, but they ask questions indicating they don’t understand later on.*

4

## Remove friction from supply

### **Example:**

From “replatform / migrate your stack” to “wrap your stack” (+ open-source, AWS-native, in your AWS)

### **Do this when:**

*You get objections around “how it works” - or lack of action despite clear demand.*

5

## Remove friction for next step

### **Example:**

From “let’s catch up next week” to “we can walk you through the open-source version”

### **Do this when:**

*Good meetings + clear demand, but drop-off to next step.*

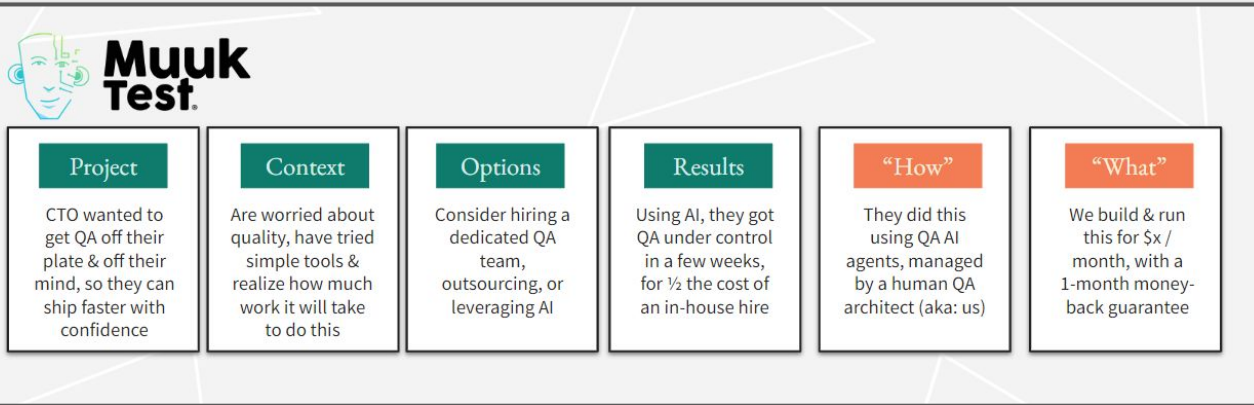
Business impact (higher)

What founders lean towards (supply-side)

Reframe



# Another way to think about this



**Earn “hell yes” on every slide / part of the case study!**

Like a funnel - if people “drop off” on the demand side, you don’t get the opportunity on the supply side!

(So - start from the left!)

# The practical approach

**Every call is an opportunity!**

## **During the call:**

1. Make the call a conversation, not a presentation
2. Thank them for their objections & questions
3. Ask clarifying questions to really understand what they're saying -> "help me understand..."

## **After calls:**








1. Review every call for 15-30+ mins
2. Have a reflection every week to debug towards "hell yes" (from specific customer examples, not from theory)

**Reframe**

## Topic 5: Scheduling Sales Calls

The metric that matters: Scheduling 5-10 per week

# Remember: Founder magic!

What matters (by stage)	The “Case Study”		
	Pipeline	Sales Process	Delivery / CS Process
<b>Get Real (0→~10 customers)</b> Figure out retention + value	<b>Not massively important.</b> Use “founder magic.” 	<b>Not massively important.</b> Use a theoretical case study or beg 	 <b>“Hell yes” (retention) by any means necessary!</b> (Don’t do things that scale.)
<b>Case Study Sales (~10→~50)</b> Nail a repeated case study	<b>Not massively important.</b> Use “founder magic.” 	 <b>Systematize.</b> Use a case study; replicate the case study. (Target 60%+ “hell yes” & close on good-fit buyers.)	 <b>Systematize.</b> Create a bullet train to “hell yes.” Retention drives product development.
<b>Find GTM Fit (50+)</b> Find a primary scalable channel	<b>Figure this out!</b> One primary scalable, cost-efficient channel. 	<b>Debug / enhance when this breaks with scale.</b>	<b>Debug / enhance when this breaks with scale.</b>
<b>Don’t Get Distracted (100+)</b> Here there be monsters	Do more of what works.	Do more of what works.	Do more of what works.

**Defend the case study!**

# Internet advice is, as usual, stupid

“You HAVE to build a repeatable, scalable cold outbound sales process from DAY 1!”

“Focus on their PAIN POINTS!”

“VOLUME!!!!!!”

“No links!”

“Here’s how I use ChatGPT at scale!”

“Write super short, punchy messages. Basically Haikus.”

“Nobody opens emails today.”

“Use these PsYcHoLoGy hAcKs!”

**Reframe**

**Most outreach is awful. The bar is low.**

- I saw you went to Harvard too! Want a demo of our product?
- Just bumping this.
- Breakup message! I'll assume you DON'T want to get our life-changing product, then!

Why? You're focused on what YOU want. They don't care.

**Reframe**

## 2 cheat codes:

1. what's the message only you can send to only them?
2. what's in it for them?

# find your cheat code



I'm just getting back from filming **Sesame Street**, and would love to learn more about your company's marketing plan, plus share what I'm hearing from other brands...

Here's my calendar link: [link]



# Early on

1. LinkedIn + email is usually more than enough to get to 5-10 per week
2. Can use automation tools (e.g., Sales Nav + Dripify; Apollo), but don't sound like you're using automation tools
3. Get to 5-10 per week, then debug your case study, sales + delivery process
4. Eventually your \*scalable\* pipeline source will emerge based on your case study

# Exercise

Draft an outreach message to someone you want as a customer!

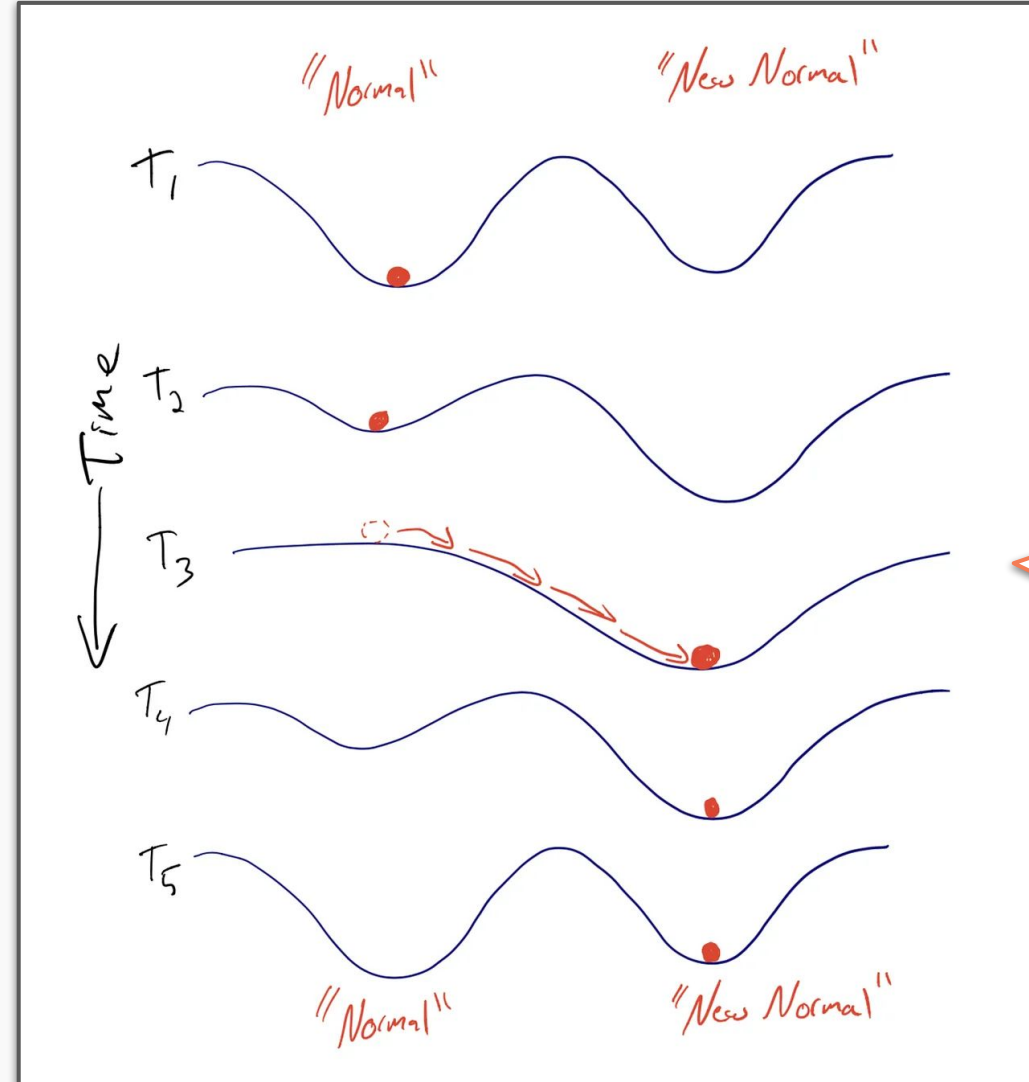
Send it to [rob@reframeb2b.com](mailto:rob@reframeb2b.com)

We will review :)

# Delivery & Recap

Have I earned my Felipe's yet, Phil?

# Sales gives you the *chance* to figure out delivery & PMF



**Retention:** They are at equilibrium 2 and don't want to / can't go back!

**Retention is the marker of PMF!**

**Reframe**

# Sales gives you the chance to figure out delivery & PMF

**Deliver manually to figure out what *causes* retention!**

**Pitch:**

“We can double your applicant flow by texting & emailing your past applicants & former employees!”

**V1: \$0 - \$125k**

Me using Google Sheets & Text-Em- All for 10 hours a day (eng watched over my shoulder)

**V2: \$125k - \$500k**

Eng fully automated the process from Google sheets

**V3+: \$500k - \$5M+**

Back-end built, no customer login (just email reports), then customer login from Bubble, then React

(Then 2nd product based on customer feedback, etc.)

# how v1 actually worked, when we started selling



## Step 1:

I logged into customers' existing applicant software & downloaded all their past applicants into a CSV



## Step 2:

I copied a Gsheets template and added the past applicants from the CSV



## Step 3:

I uploaded 50 contacts per store per day, sent a mass recruiting message, and chatted w/ candidates



## Step 4:

When a candidate was interested, I'd click a button in the Gsheet to send an email to the recruiter



## Step 5:

Every week, I'd manually create & email a report for the customer based on Gsheet data



## Step 6:

Every 2 weeks, we'd have a Zoom check-in with the recruiters to see what they thought



**8am-7pm for 3 months**  
w/ eng on Zoom watching me

- ✗ No customer login
- ✗ No website
- 💰 3-slide sales deck
- ⚙️ Eng shipped things to keep me alive

# what the customer experienced



**1:1 Personalized  
Onboarding with a  
Founder**



**Simple, easy-to-  
understand email  
reports** (“The best  
part? You don’t have  
to learn some new  
system!”)



**My direct phone  
number to yell at me  
when things went  
wrong**

# Principles

**Customers are buying a project, your product is nested within that.**

- Does not matter to them what % is product vs. elbow-grease on the back-end (for now)
- FORCE them to project success by ANY MEANS NECESSARY.  
They don't have an option, they are going to get the results (!!)
  - Eventually, nail self-serve / systematized onboarding
  - BUT AFTER YOU'VE FIGURED OUT WHAT "HELL YES"  
POST-IMPLEMENTATION IS ("leading indicator of retention / upsell")
- **You need far less "product" than you think**

**Reframe**



# The goal

Know EXACTLY WHAT happens to get them to “hell yes” so you can systematize it for EVERY new customer

→ This is the “leading indicator of retention / upsell”

	Learn		Scale
	#1 Product-Market Fit	#2 Go-to-Market Fit	#3 Growth and Moat
Goal of Phase	Customer Retention	Scalable Unit Economics	Revenue Growth Rate
Target Market	Early Adopter	Early Majority	Scale vs. Experiment vs. Ignore
GTM Playbook	Win At All Cost	Codified, Scalable	Reinforced
Sales Hire	PM + AE	Process Builder	Process Executor
Demand Gen.	Personal Network + Referrals	1 Scalable, Measurable Medium	Multiple Mediums. Tightly Aligned with Sales.
Pricing	Solve for Customer Commitment	Solve for Unit Economics	Assess Disruption Risk
Compensation	Based on Customer Retention	Customer Retention + Unit Economics	Add Promotion Path

Reframe

# “What advice would you give your 0-1 self?”



**Chris LaFleur** • 1st

4m ...

Bringing the next generation of smart, zero-emission machinery to life.

Not worrying about how big the opportunity is. Founders are pushed to find the next huge thing and show how they're disrupting a massive market. I would get stuck in this painful loop of how to connect my small (but real) idea to some BS industry shift. Just find a problem, big or small, that is solid and that you have passion for.



**AJ Richichi** (He/Him) • 1st

6h ...

CEO/Sprockets, Author/Hourly

PMF isn't just about selling. A customers' appetite for risk on buying a new product is proportional to their level of pain. So, if you find a problem grand enough - selling will be the easy part.

Don't sell sell sell. Instead, sell... then study study study. Ensure that the product solves the problem.

If it doesn't, change solution or change customer.  
If it does, scale.



**Dan Willoughby** • 2nd

7h ...

Creator of tellspin.app | L2 Rotations in Slack | Principal Software Engineer at Storj

Don't write code...

No seriously, do not build anything. Go pre-sell the software.

You can spend months building it and then try to find people to buy it.

OR

Find people to buy it then build it.

Either way, you have to find customers, so might as well solve that problem first



**Alex McLeod, MBA** • 1st

7h ...

Founder @ Parlay | Financial Inclusion | Capital Readiness for SMBs | TEDx Speaker & Coach

Change your success metrics from sales to learning until you can start to sense early indications of pmf. Saves you from being distracted by things that you're not ready for.



**Masha Krol** (She/Her) • 1st

11m ...

CEO @ Glowstick 🚀 DM me to share rock climbing beta 🧗‍♀️

- just cause there's a gap in the market, doesn't mean there's a market in the gap (saw this somewhere, resonated hard)
- don't build for the world as you think it should be, find out how it actually is and build for that (demand-first, of course)
- if YOU don't have conviction, you're ngmi
- follow this Rob guy

# Some last thoughts

1. Demand rules everything. Your job is to find it, you don't create it or control it.
2. Business is just a system to replicate case studies. It's not more complicated than that.
3. Sell to learn, using a case study. You'll look dumb today to look smart tomorrow.
4. The whole point is to serve customers to achieve what matters to them. It's not about you or how you think the world should work.