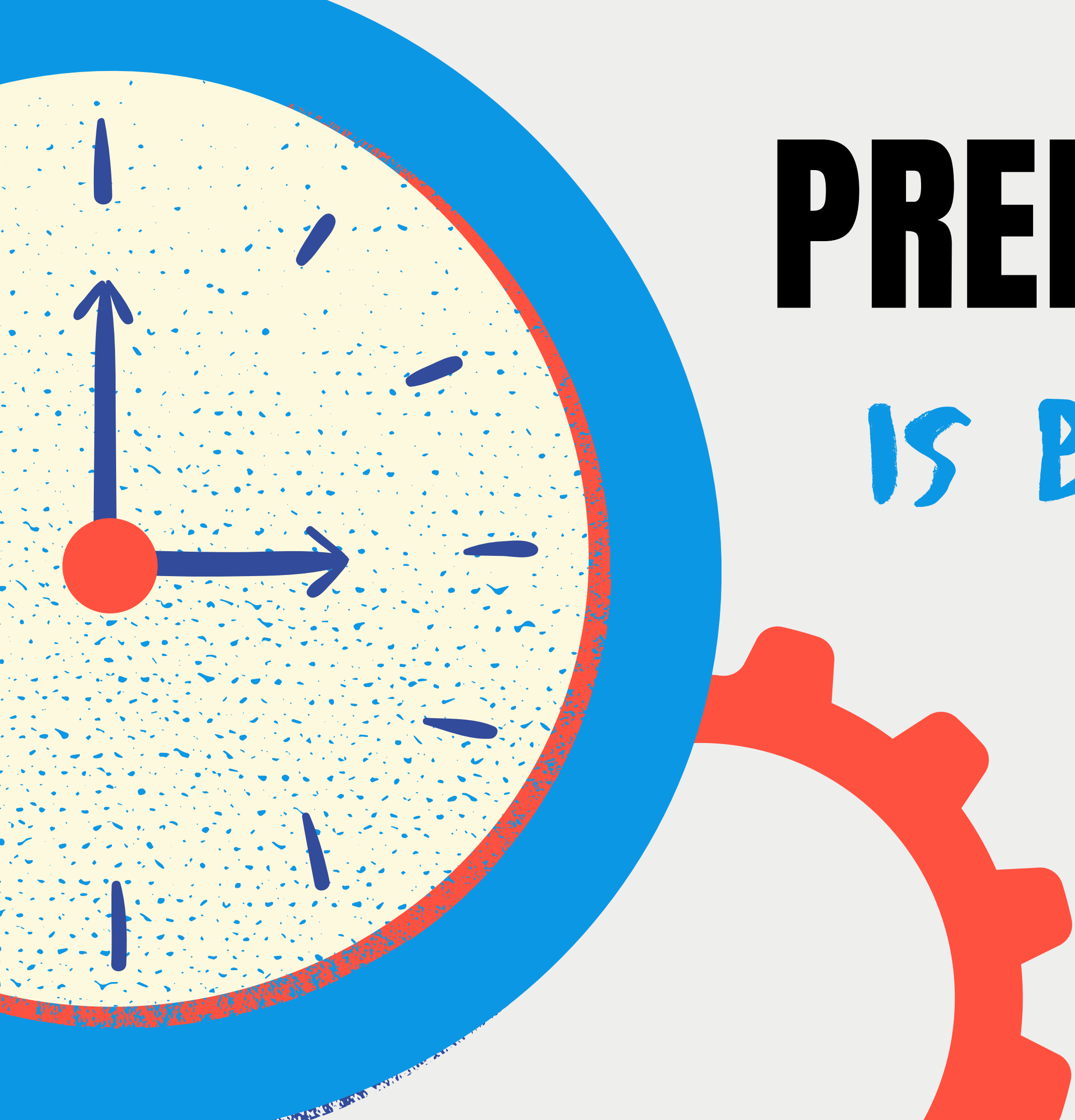




PREDICTABLE

IS BETTER THAN

FAST

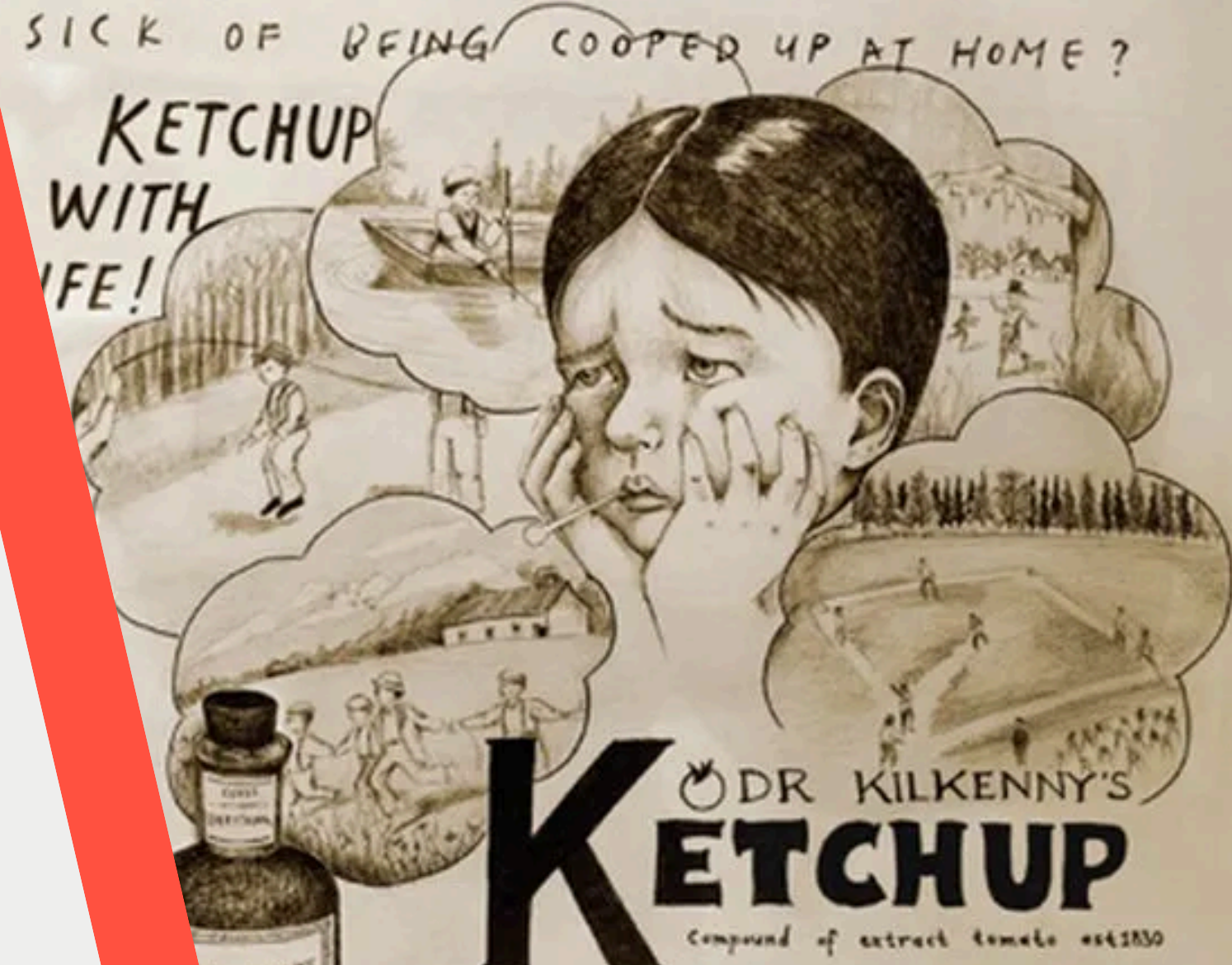
Chris Bigelow, LeadDev NYC 2025

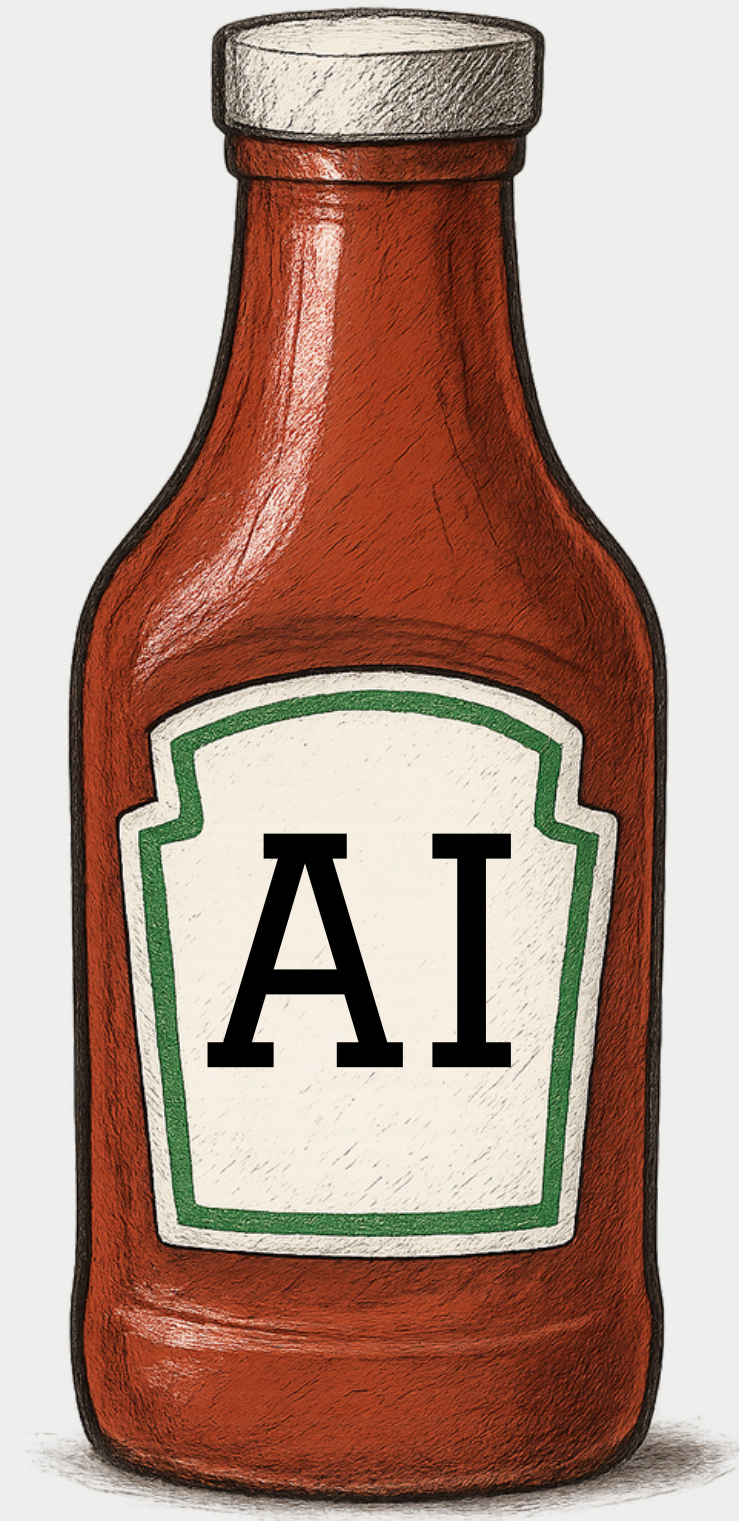
FUN FACT #1:

KETCHUP WAS ONCE SOLD AS MEDICINE.

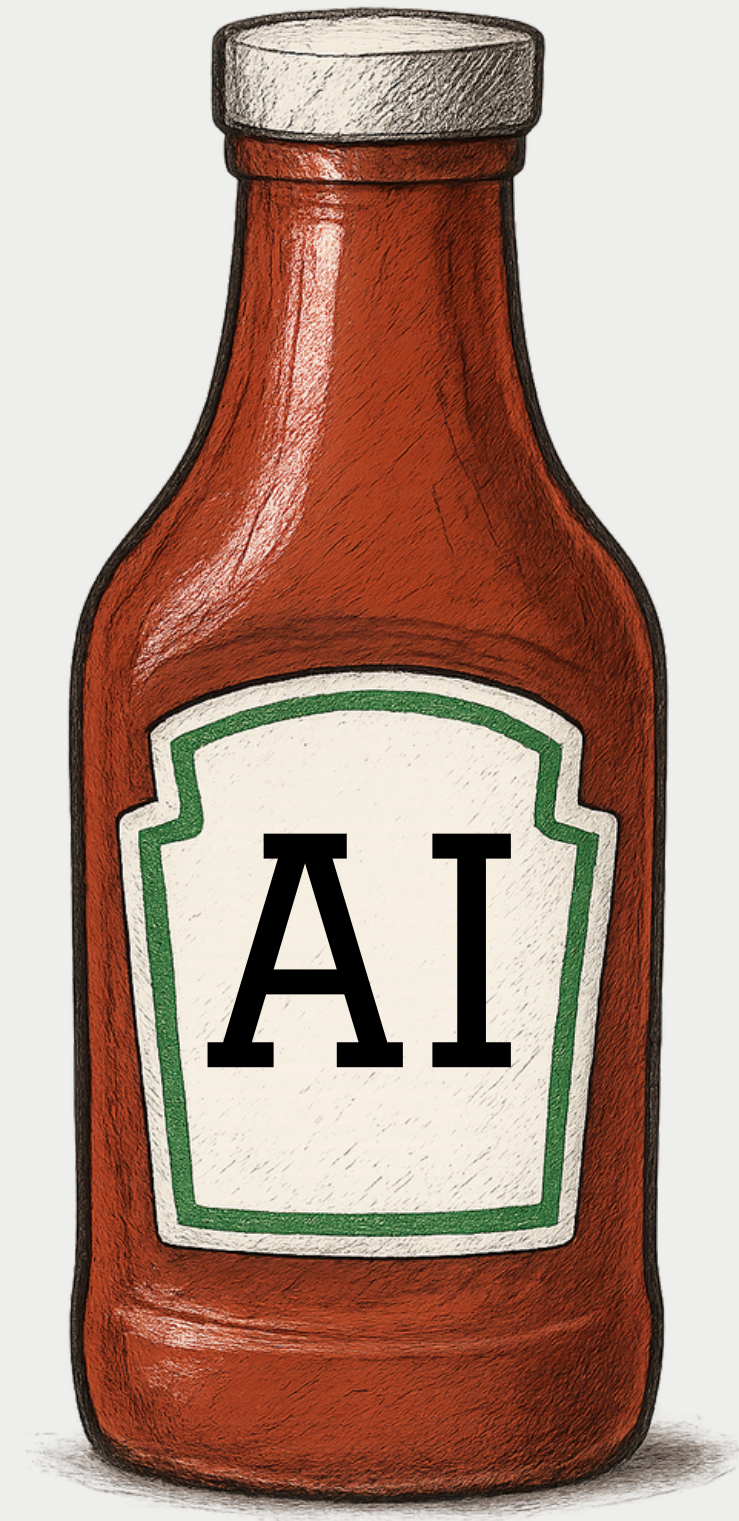


<https://brucewilsonauthor.medium.com/was-ketchup-used-as-medicine-920f0a1ba1b4>





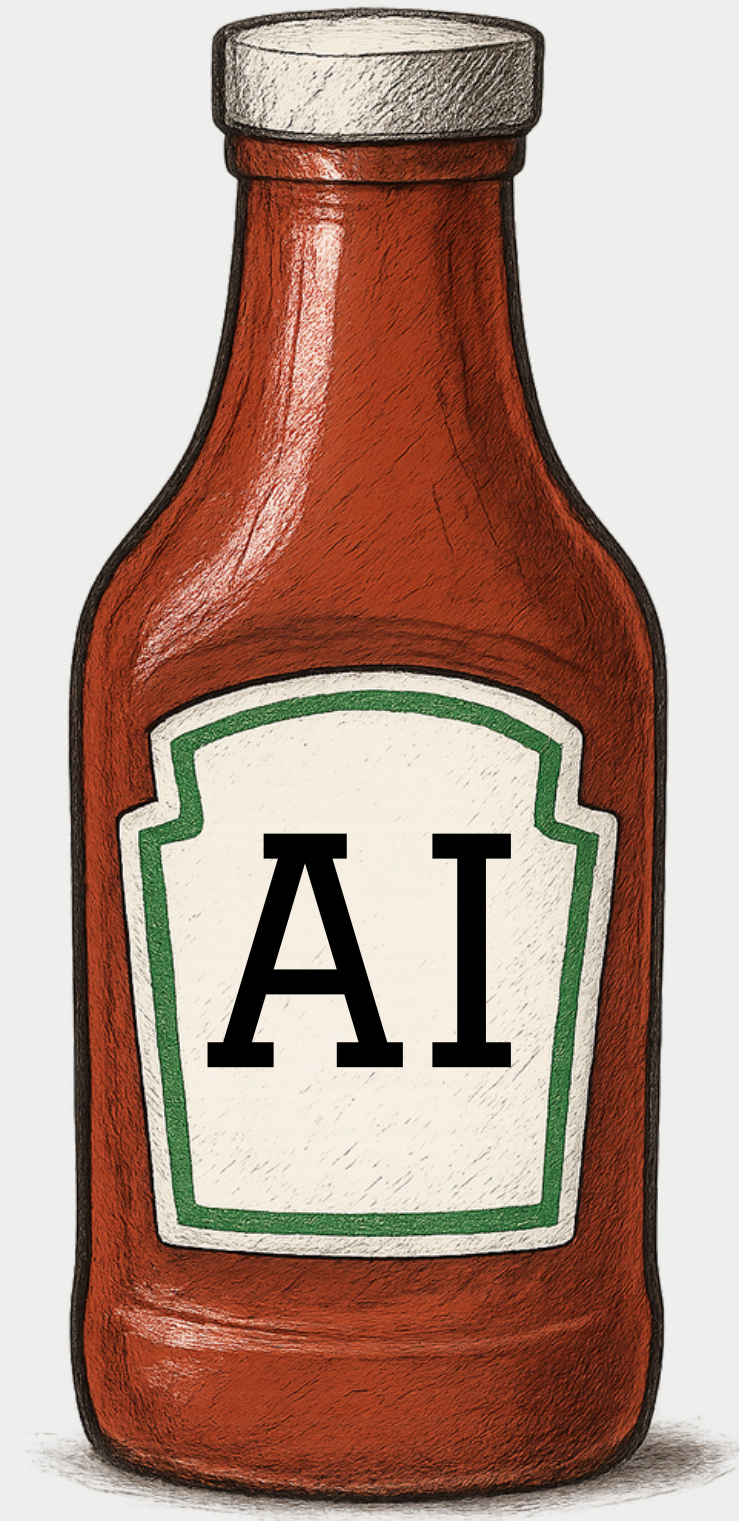
AI will alleviate our business woes !



GOING FASTER

AI will alleviate our business woes !

AI = Speed



GOING FASTER

AI will alleviate our business woes !

AI = Speed



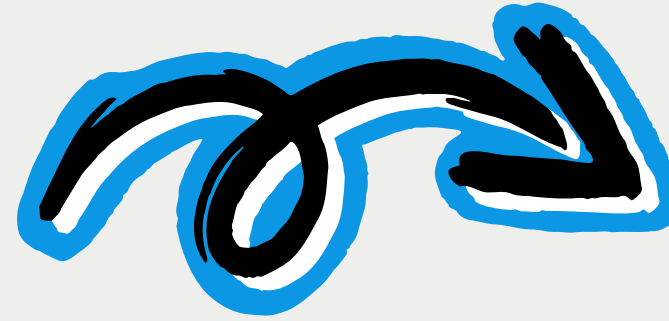
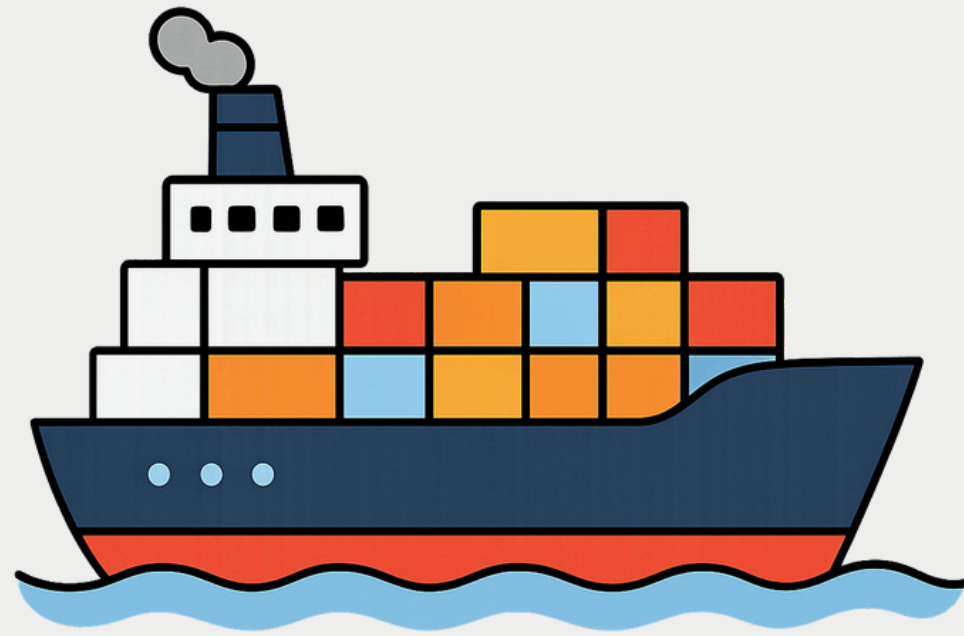
Intro Time 🙌

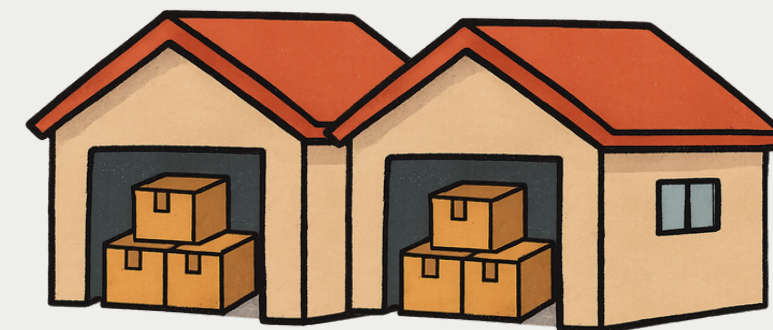
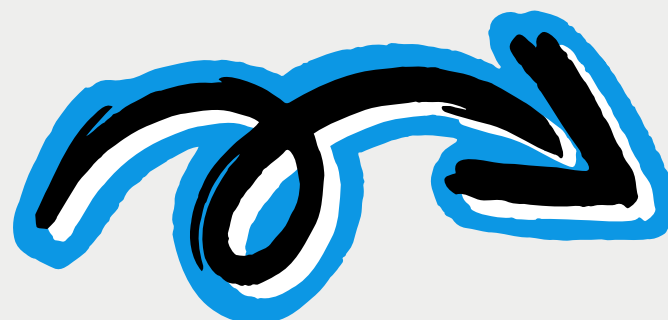
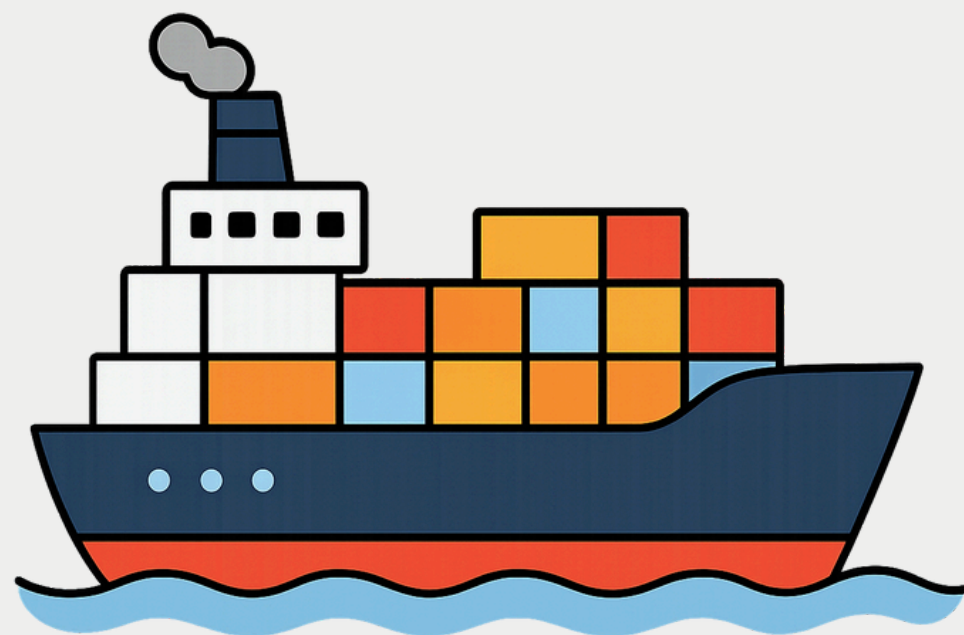
LOVEVERY®



9-month-old

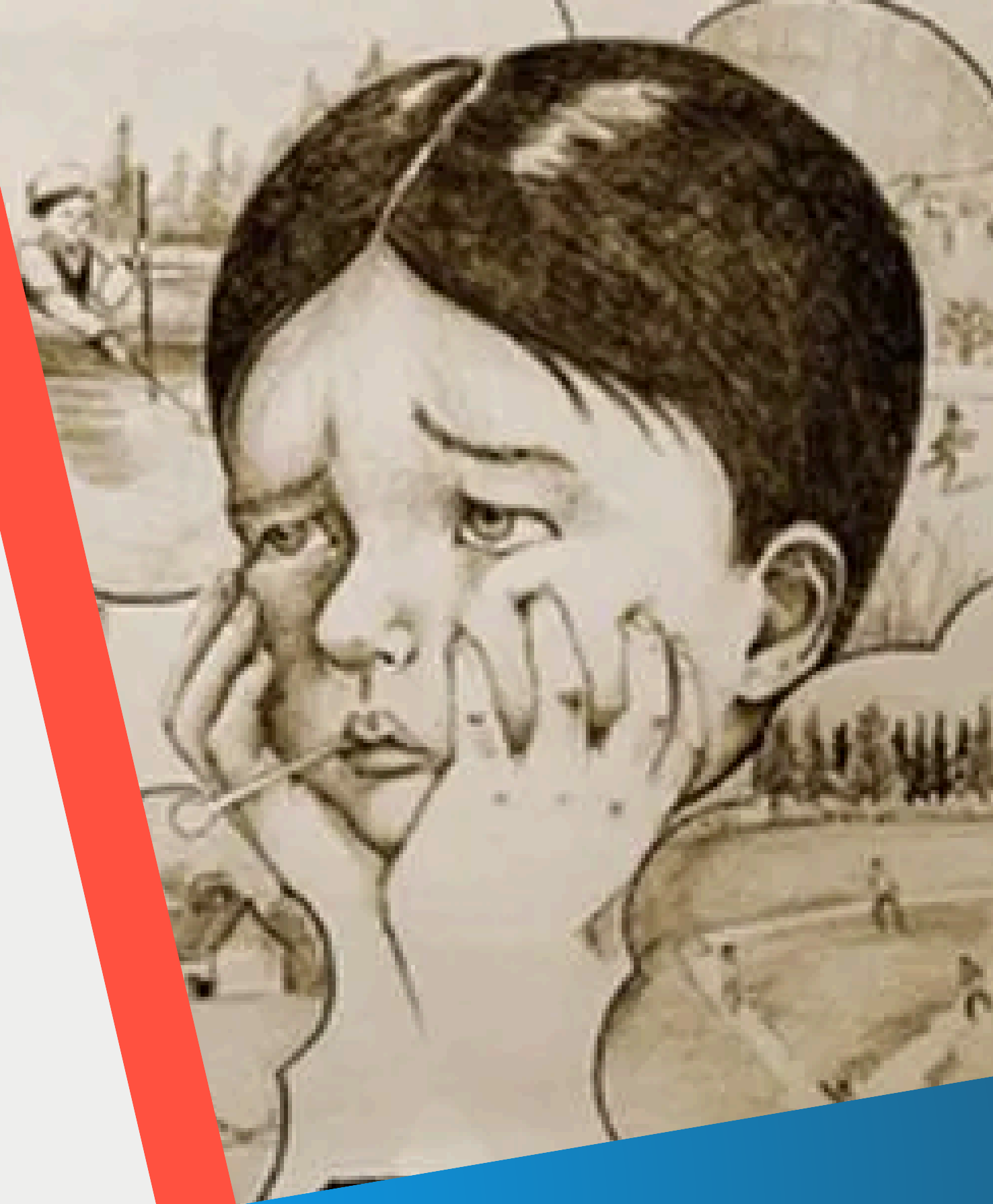
2-year-old





FUN FACT #2:

I'VE ALWAYS HAD THIS PROBLEM!!

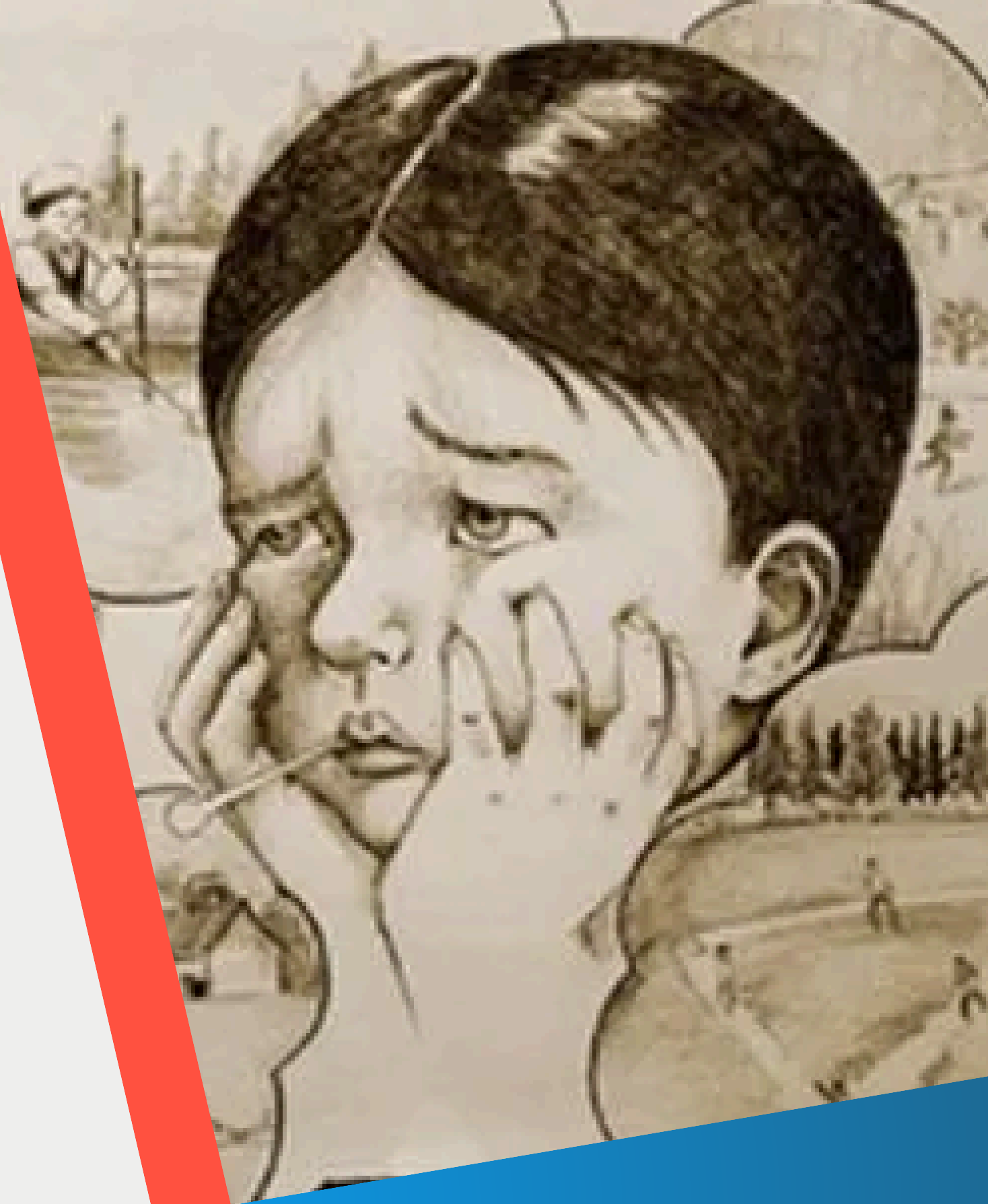


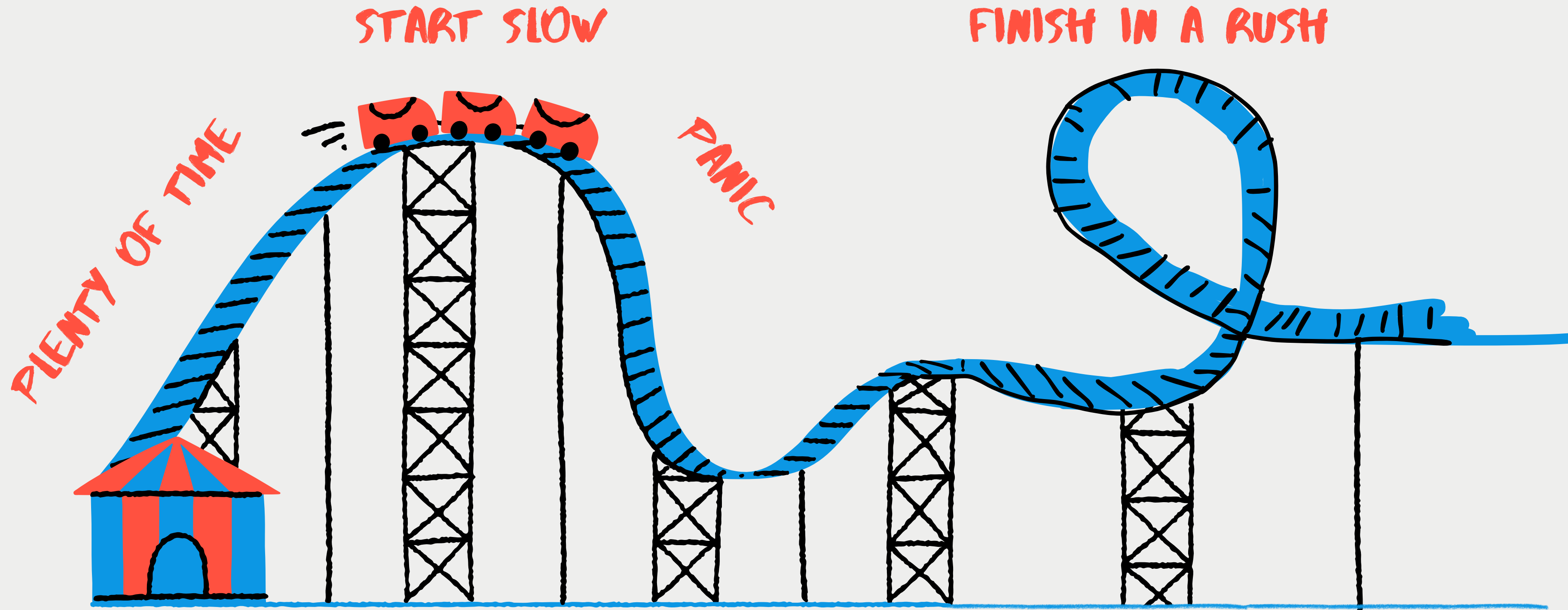
FUN FACT #2:

I'VE ALWAYS HAD THIS PROBLEM!!

You all have this
same problem too.

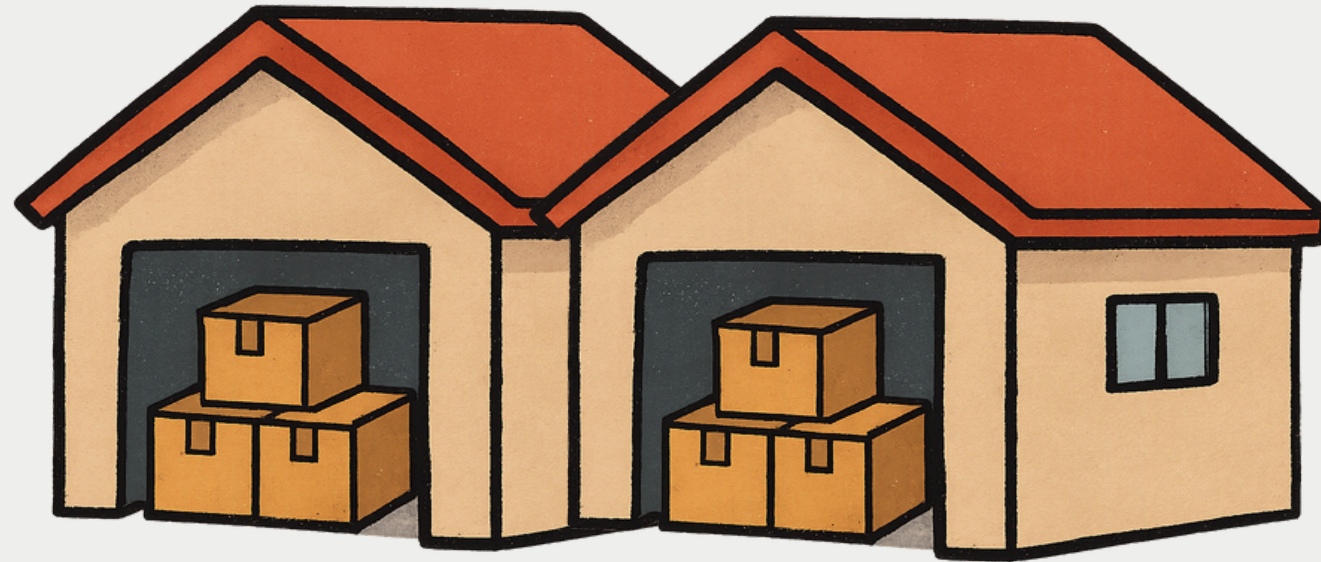
Muahahaha!





THE COST OF MISSING A DELIVERY DATE

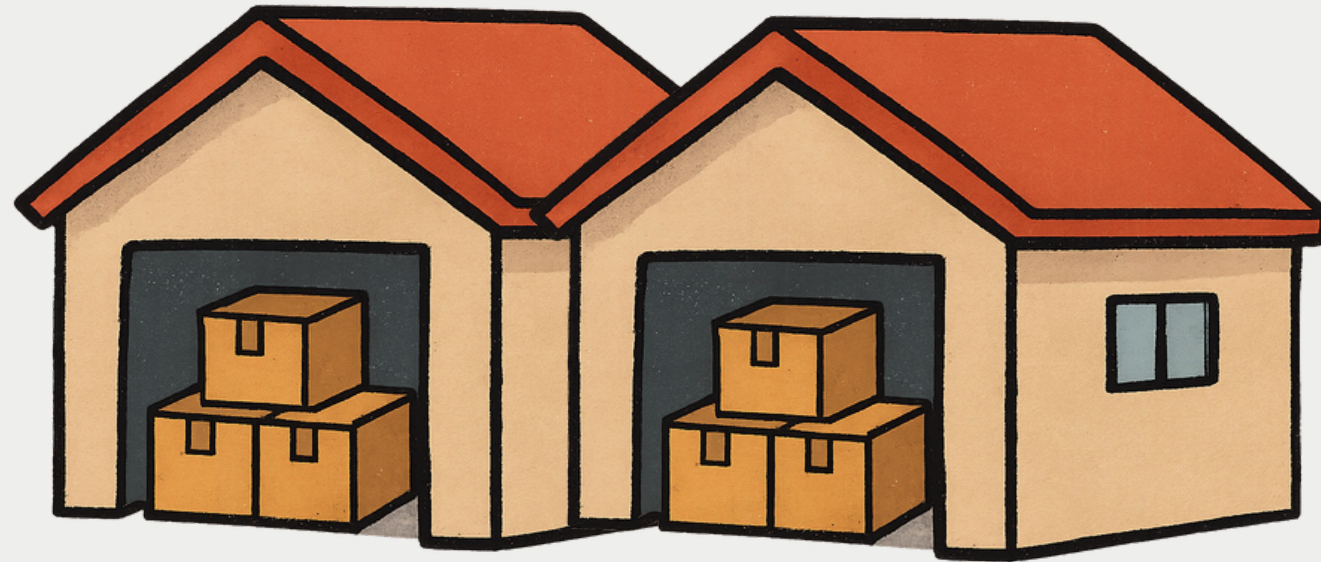
COST OF FINISHING LATE



=

-\$£€

COST OF FINISHING LATE



=



COST OF FINISHING **LATE**

- **Marketing:** How to hit the emergency brakes on the campaign?
- **Sales:** How to manage reputation from not delivering on our promises?
- **Product:** How to shift timelines for cascading delays on other projects?
- **Engineers:** How to launch faster with missing or untested features?





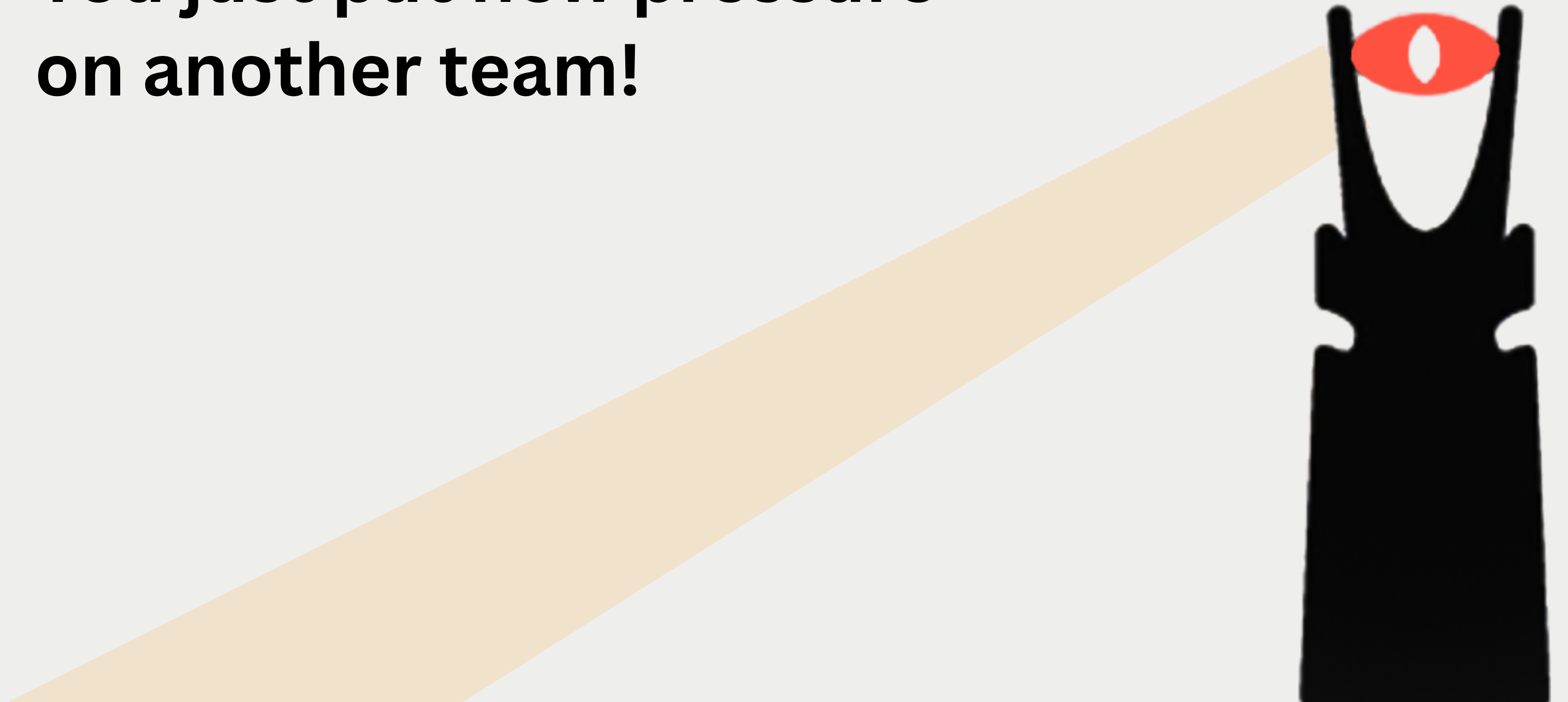
MORE DECISIONS

COST OF FINISHING **EARLY**



COST OF FINISHING **EARLY**

- You just put new pressure on another team!





MORE DECISIONS

COST OF FINISHING **EARLY**

- **Go live early?**
- **Hold it unmerged?**
- **Merge it, behind a feature flag?**

COST OF FINISHING **EARLY**

Go live early?

- How to measure cause and effect?
- A/B test collisions
- Another team was not expecting to merge on top of your changes

COST OF FINISHING **EARLY**

Hold it unmerged?

- **Continuous rebasing**
- **Consuming brain space**
- **Team can't fully move on**

COST OF FINISHING **EARLY**

Merge it, behind a feature flag?

- **Best Practice, but still need to implement a flag**
- **Testing for all possible code paths**
- **Watch out for misconfiguration**

COST OF FINISHING **EARLY**

In any case – your PM needs to be ready **WAY** in advance!

- **More decisions**

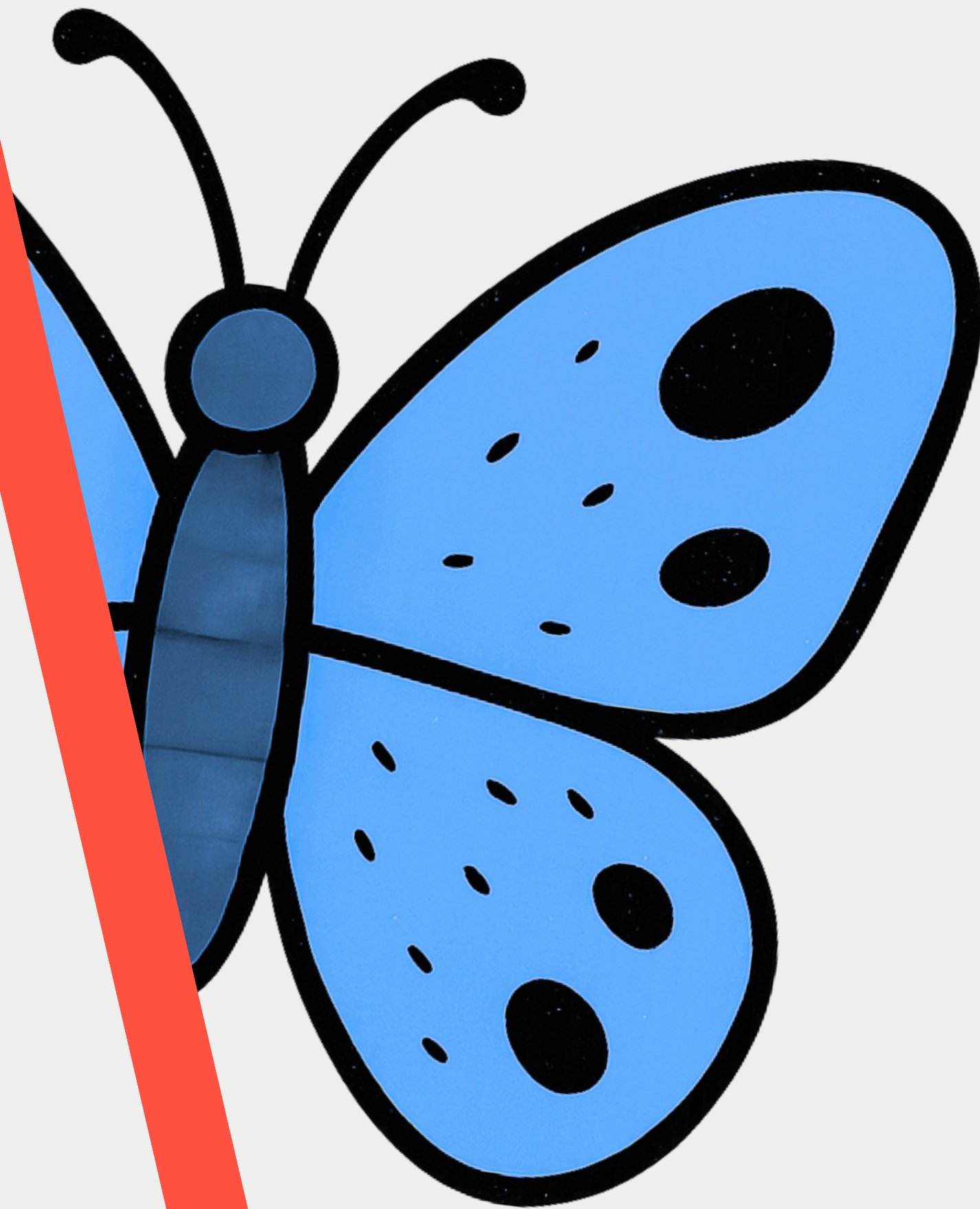
- Worth it to try to launch early?
- What should team work on next?
- What's the ROI of upcoming projects?
- Will the design mocks be ready?
- etc.



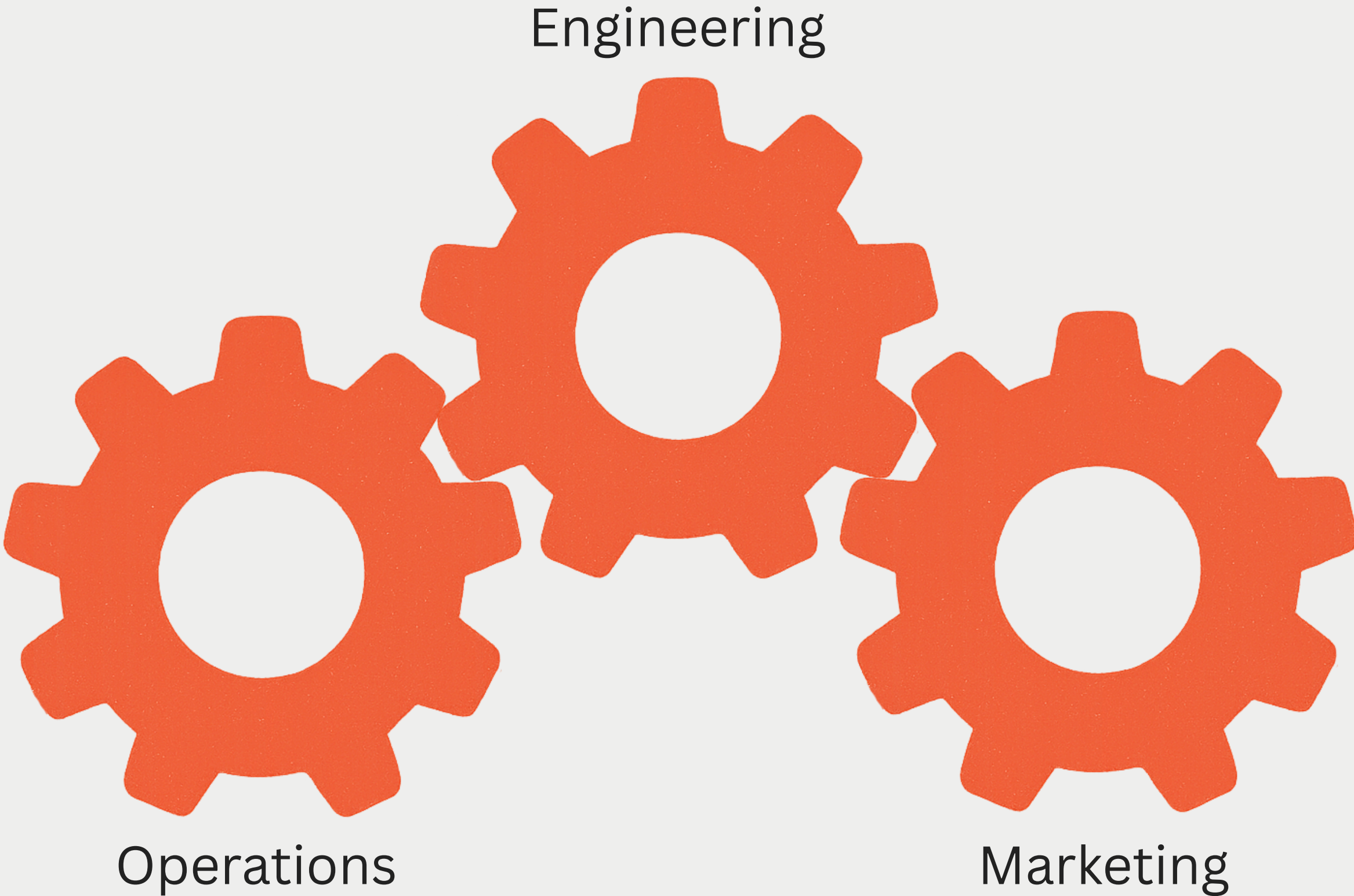
FUN FACT #3:

ORGANIZATIONAL BUTTERFLY EFFECT

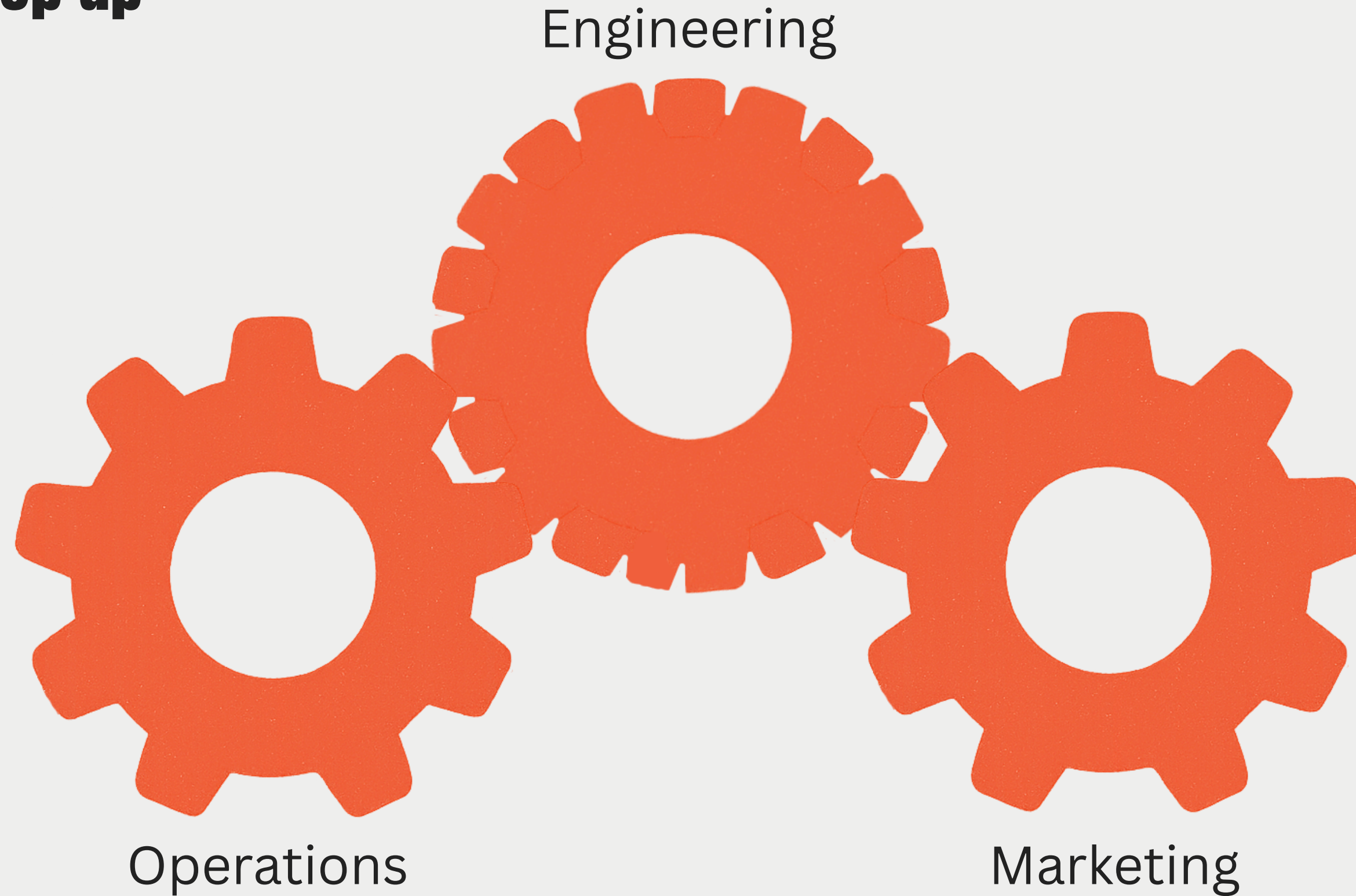
You impact your **entire company** by either hitting or missing **your** target date!



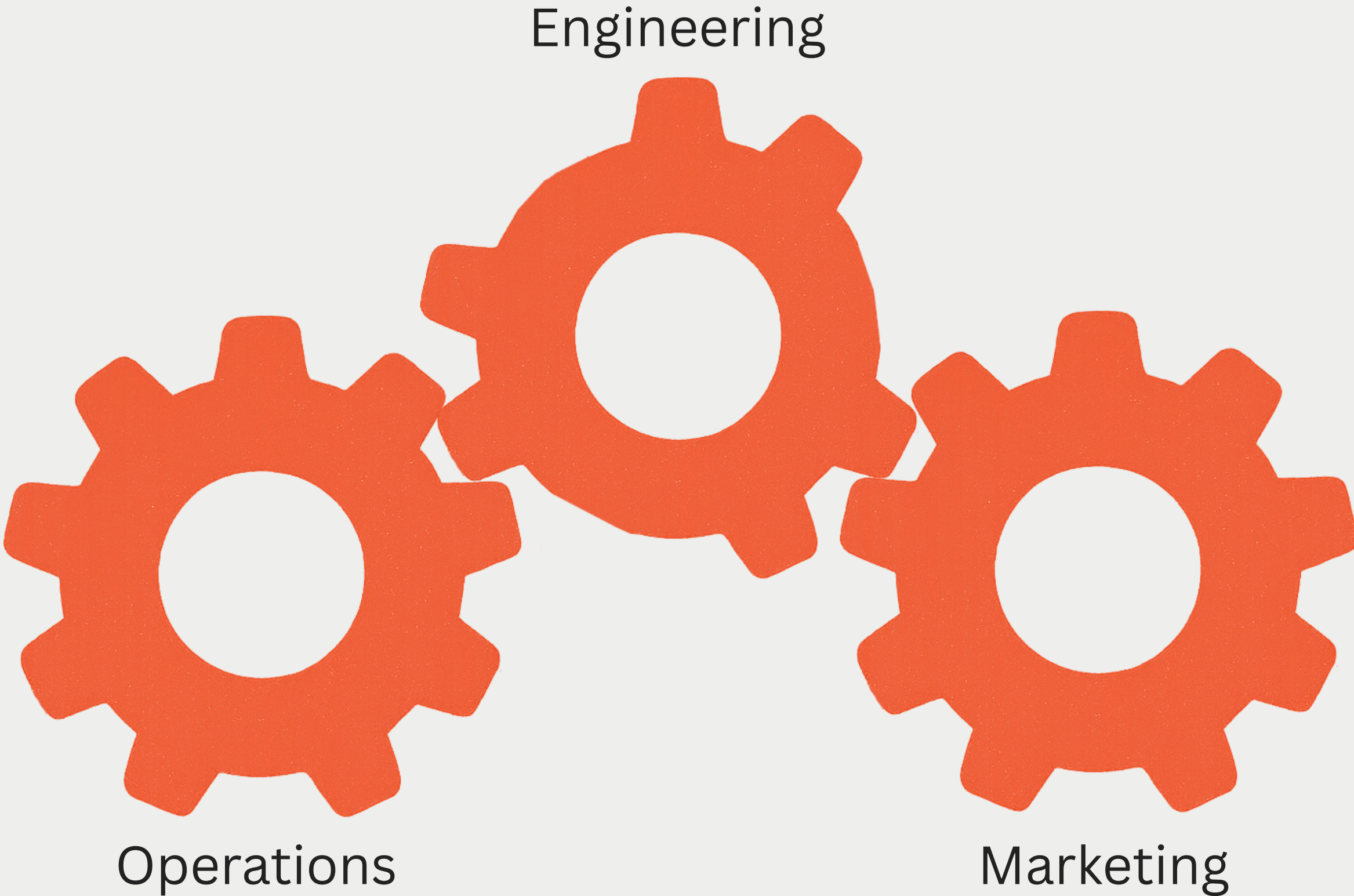
Organization running smoothly



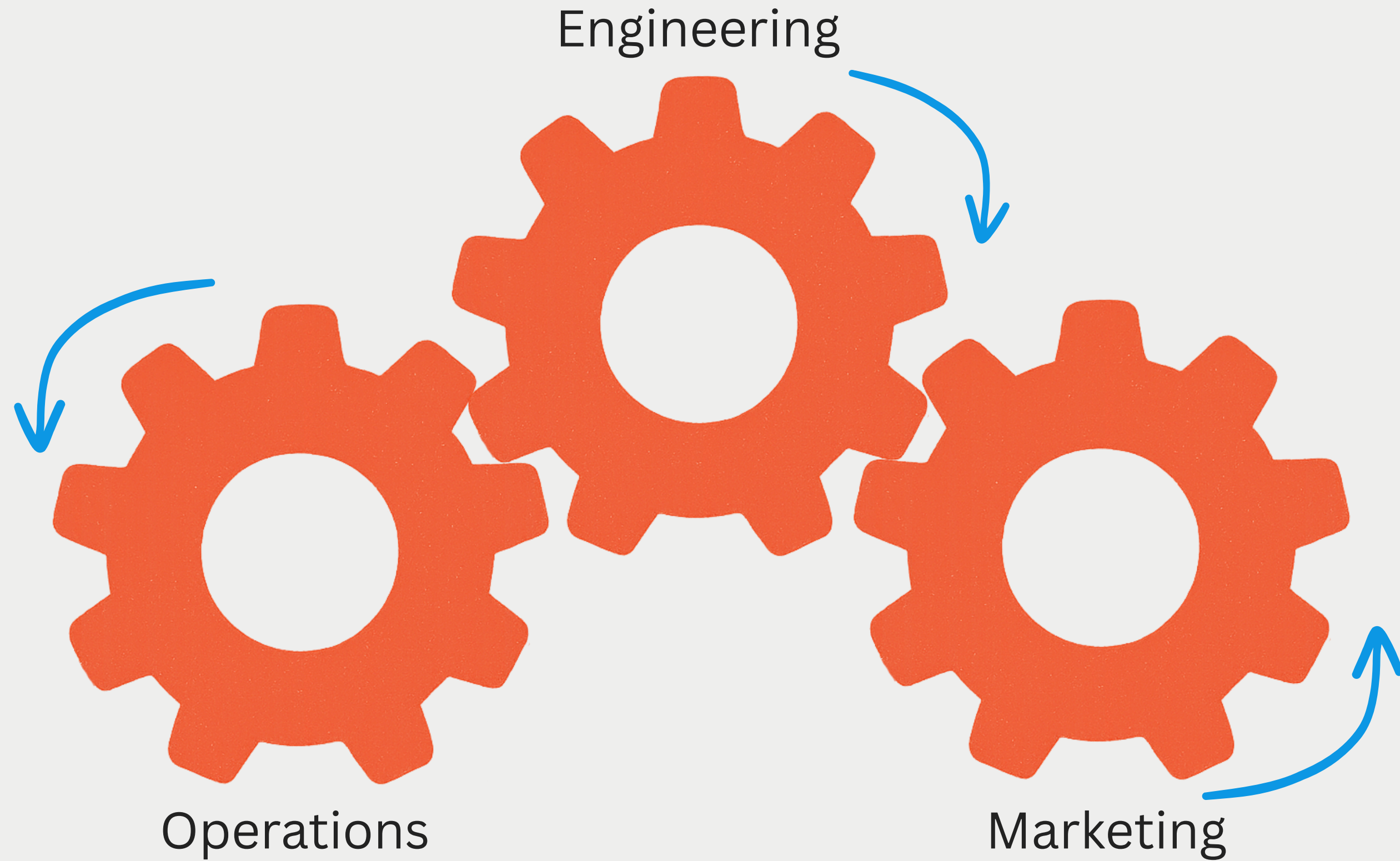
**One team delivers too quickly,
others can't keep up**



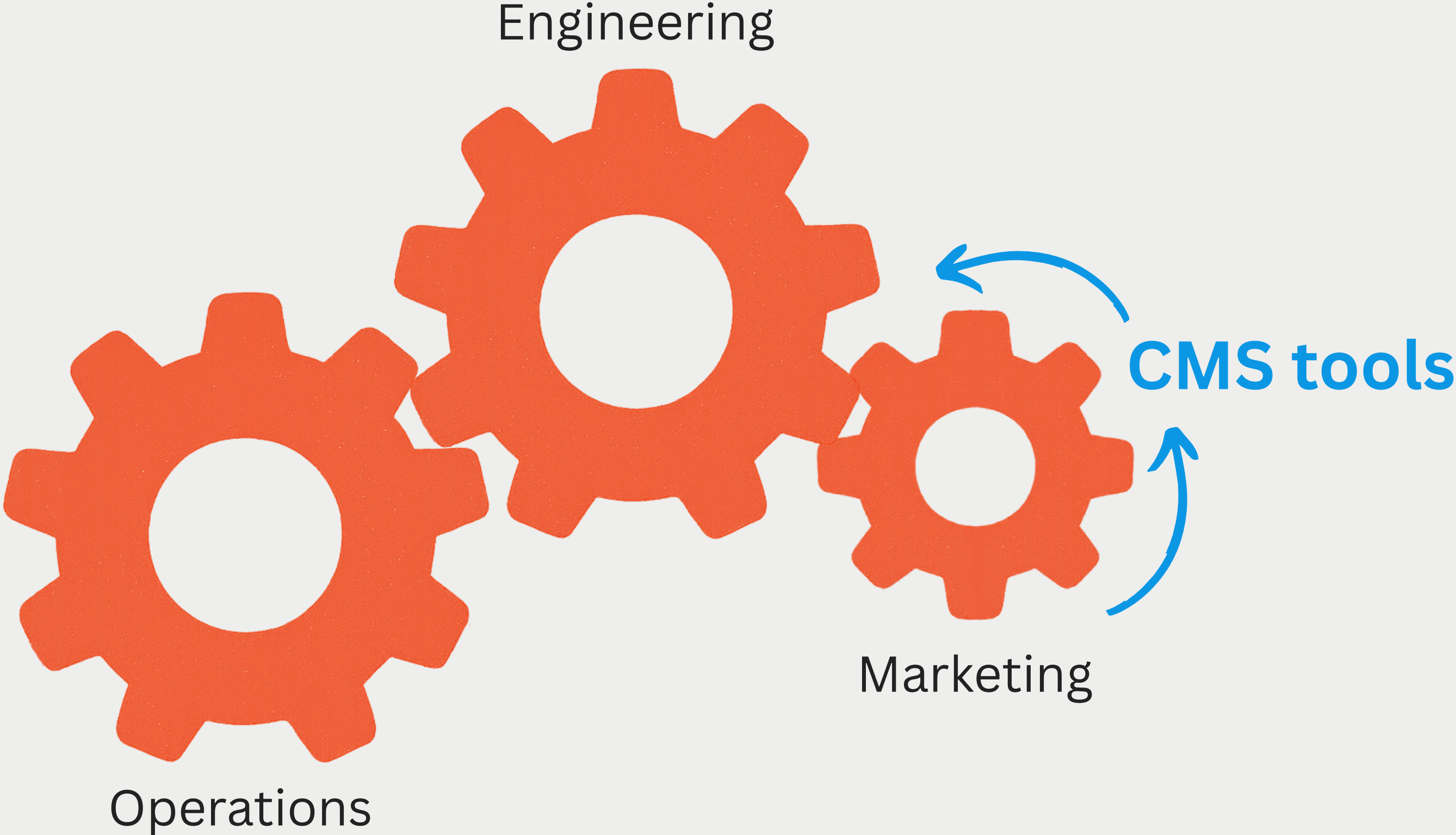
One team delivers unpredictably



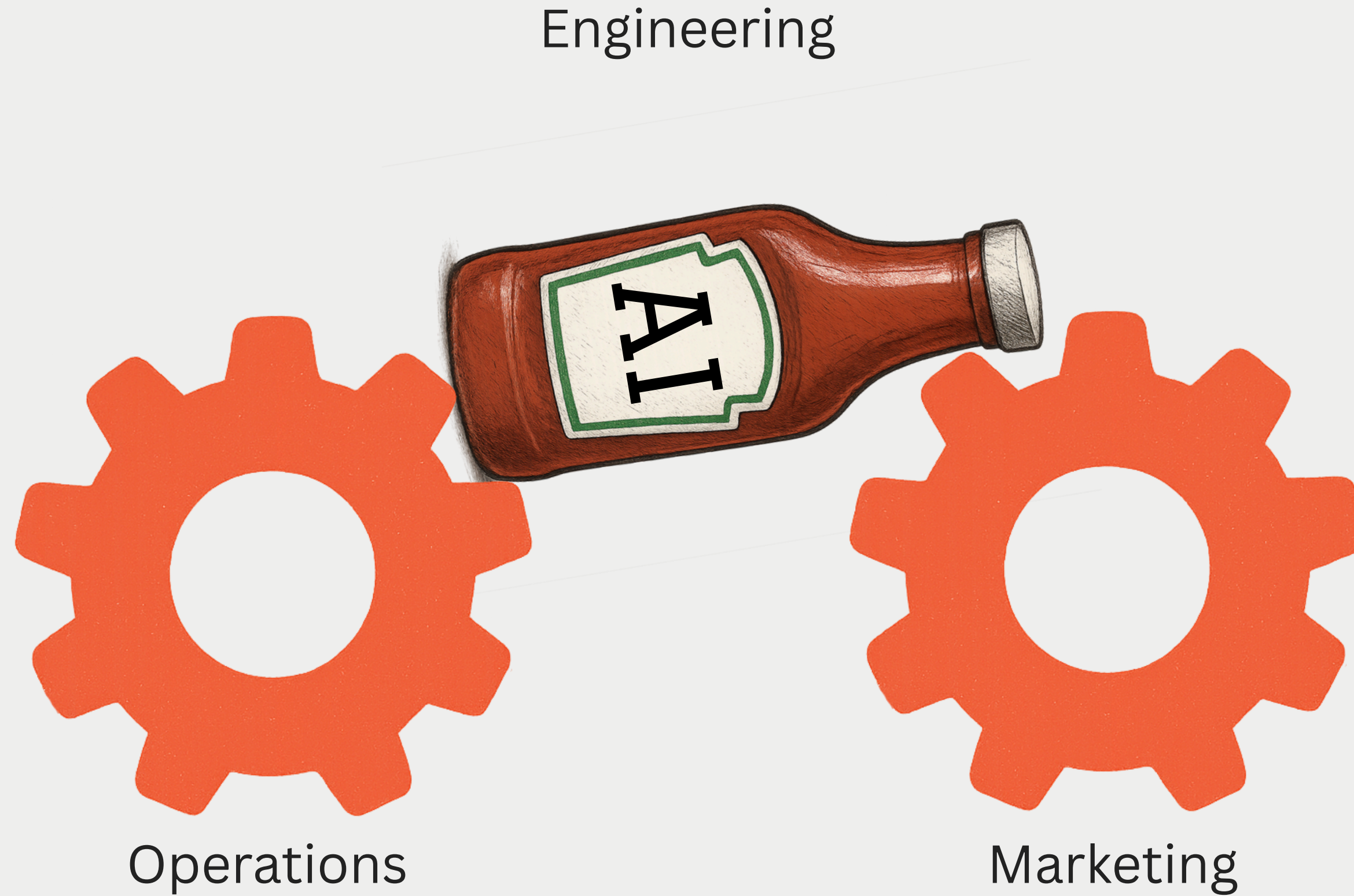
Truly Optimizing = Higher RPM



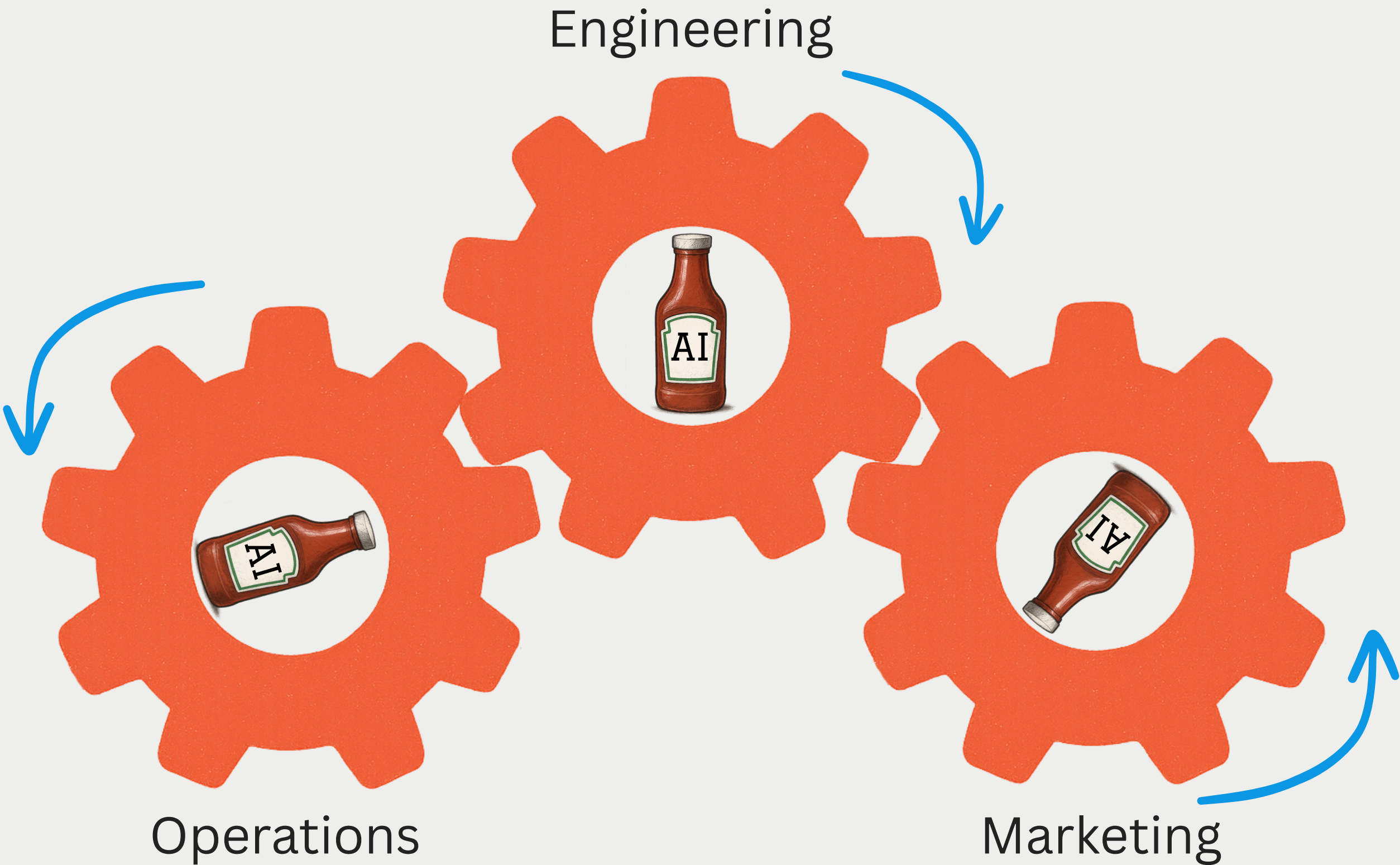
Truly Optimizing = Higher Torque



Skipping Steps = **NOT** Optimizing



Apply AI to optimize functions, not replace them



ASK YOURSELF:

WHAT IS THE PRIMARY REASON WHY MY TEAM IS NOT DELIVERING PREDICTABLY?

Common root causes:

- Last min changes from stakeholder review
- Not accounting for tech design and testing
- Code changes have unexpected consequences
- Saving riskiest work for last
- ...



DEVS ADD TOO MUCH BUFFER TIME

Probable cause: Tracking work too granularly

- **Projects** have a time commitment, **Individual tasks** do NOT
- **Squads** have an average velocity, **Individual developers** do NOT
 - Make it safe to “miss” on a story points estimate
- **Use teamwork to normalize the noise**
 - Someone will finish early, someone else will get stuck
 - Set *shared* sprint objectives, and finish on time

1st - Fast and Predictable

2nd - **Slow and Predictable**

3rd - Fast but Unpredictable

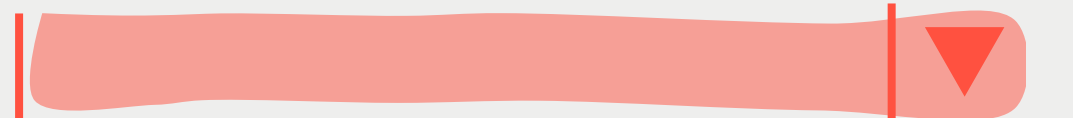


**SLOW &
PREDICTABLE** > **FAST &
UNPREDICTABLE**





AI =



PREDICTABILITY

Communication

Speed

& more...

THANK YOU

Connect with me at

<https://www.linkedin.com/in/seebig/>



Thanks to [slidescarnival.com](https://www.slidescarnival.com) for the presentation template and graphics