

YOU ARE THE MICRO CTO

Managing Your Team to
Effectiveness

FRANCISCO TRINDADE

Vice President of Engineering at Braze

franciscotrindade.me

THE STORY OF A VERY SUPPORTIVE MANAGER



THE STORY OF A **VERY SUPPORTIVE MANAGER**



Their focus was to protect their team and engineers

Ensure every engineer could prioritize working on what they were mostly interested in

Would fight back any conversation about new work to avoid change direction

THE STORY OF A
**VERY
SUPPORTIVE
MANAGER**

*‘Everyone is happy, but
the team doesn’t deliver’*



WHAT IS THE PROBLEM IN THIS SITUATION?

Focusing the EM role on individual happiness will create systems optimized for individuals

THE STORY OF A **VERY SUPPORTIVE MANAGER**



WHAT IS THE PROBLEM IN THIS SITUATION?

THE STORY OF A **VERY SUPPORTIVE MANAGER**

*“Leave people alone. Let the
engineers do their stuff”*



THE STORY OF A **VERY SUPPORTIVE MANAGER**



WHAT IS THE PROBLEM IN THIS SITUATION?

That often leads to ineffectiveness:

- People working in misalignment
- Teams getting blocked on simple issues
- Lack of drive towards a goal

And it ends up **worse for the individuals.** Less impact. Fewer rewards

ENGINEERING
LEADERSHIP IS NOT ONLY
ABOUT MANAGING
INDIVIDUALS

**IT'S ABOUT LEADING A
TEAM IN DELIVERING
VALUE. VALUE ALL THE
WAY DOWN.**

ENGINEERING
LEADERSHIP IS NOT ONLY
ABOUT MANAGING
INDIVIDUALS

**IT'S ABOUT LEADING A
TEAM IN DELIVERING
VALUE. VALUE ALL THE
WAY DOWN.**

*Effectively, to be a leader.
A CTO for your team*

HI, I'M FRANCISCO TRINDADE



HI, I'M
**FRANCISCO
TRINDADE**



ENGINEERING LEADERSHIP AT BRAZE

Previously:

- Engineering in different companies
- Consulting at Thoughtworks for 6 years
- Started companies twice
- Managing ~100 person team

I've been:

- An engineer
- A founder
- A manager of small teams
- A manager for large teams

BEING
A LEADER
IS NOT EASY



BEING
A LEADER
IS NOT EASY

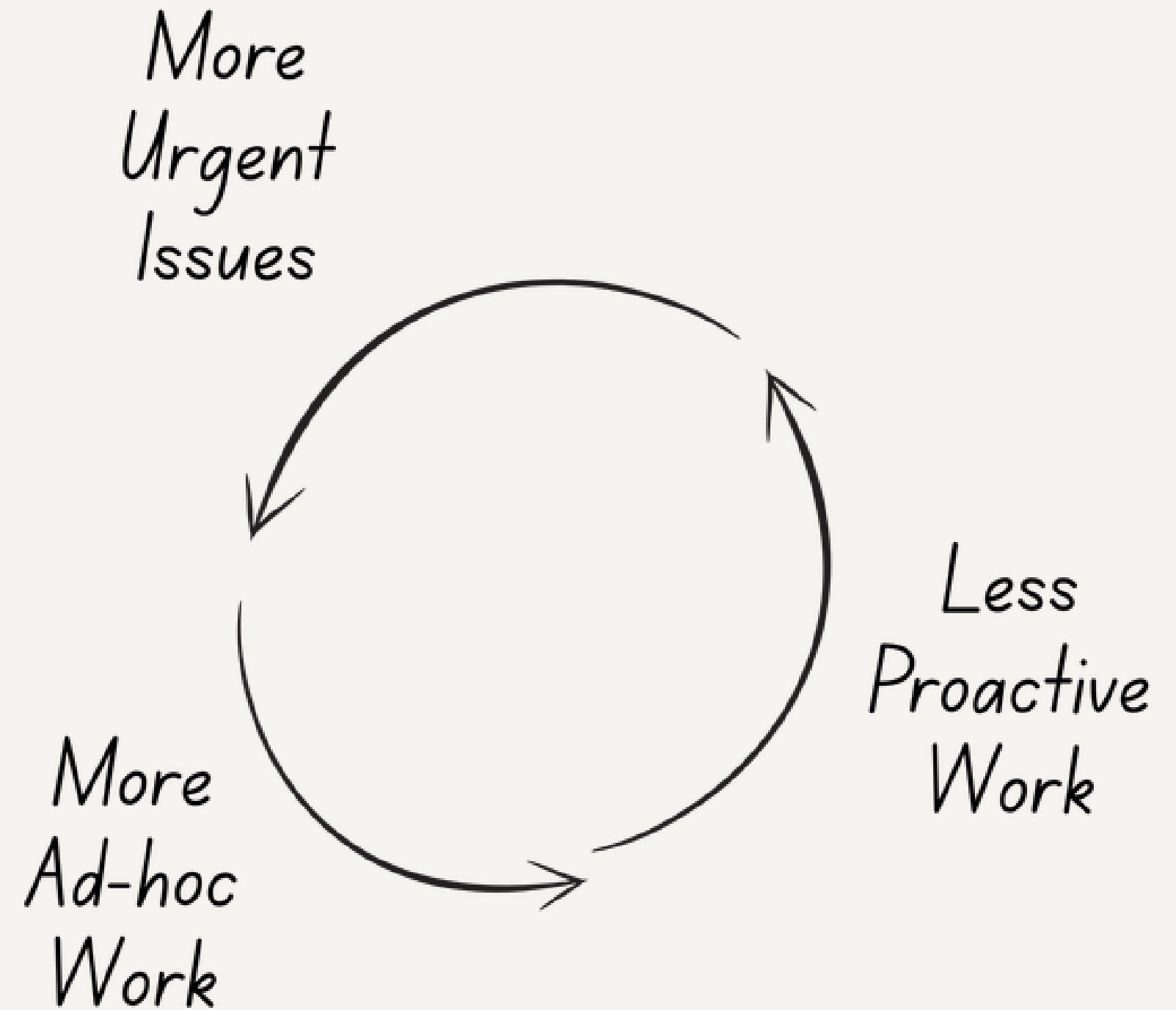


There are too many things to do. Too many opinions. Too many problems.

Focusing on each one separately leaves managers with no time.

It also creates a reactive pattern.

**BEING
A LEADER
IS NOT EASY**



BEING
A LEADER
IS NOT EASY



AND THAT IS THE PATTERN

Technical managers that deliver a lot of code and end up blocking the team

People focused managers that treat each individual as a series of needs to be attended

Process focused managers that create bureaucratic teams

BEING
A LEADER
IS NOT EASY



**AND THAT IS THE
PATTERN**

Managers that focus on **the parts**
and not on the whole

LEADING
IS SEEING THE
**FOREST AND
THE TREES**



LEADING
IS SEEING THE
**FOREST AND
THE TREES**



YOU NEED TO DO BOTH

Understand the high level perspective. What is the value?


Be an expert on your team so you know which levers to pull.

LEADING
IS SEEING THE
**FOREST AND
THE TREES**



In fact, it's not only one system...

Business
Outcomes



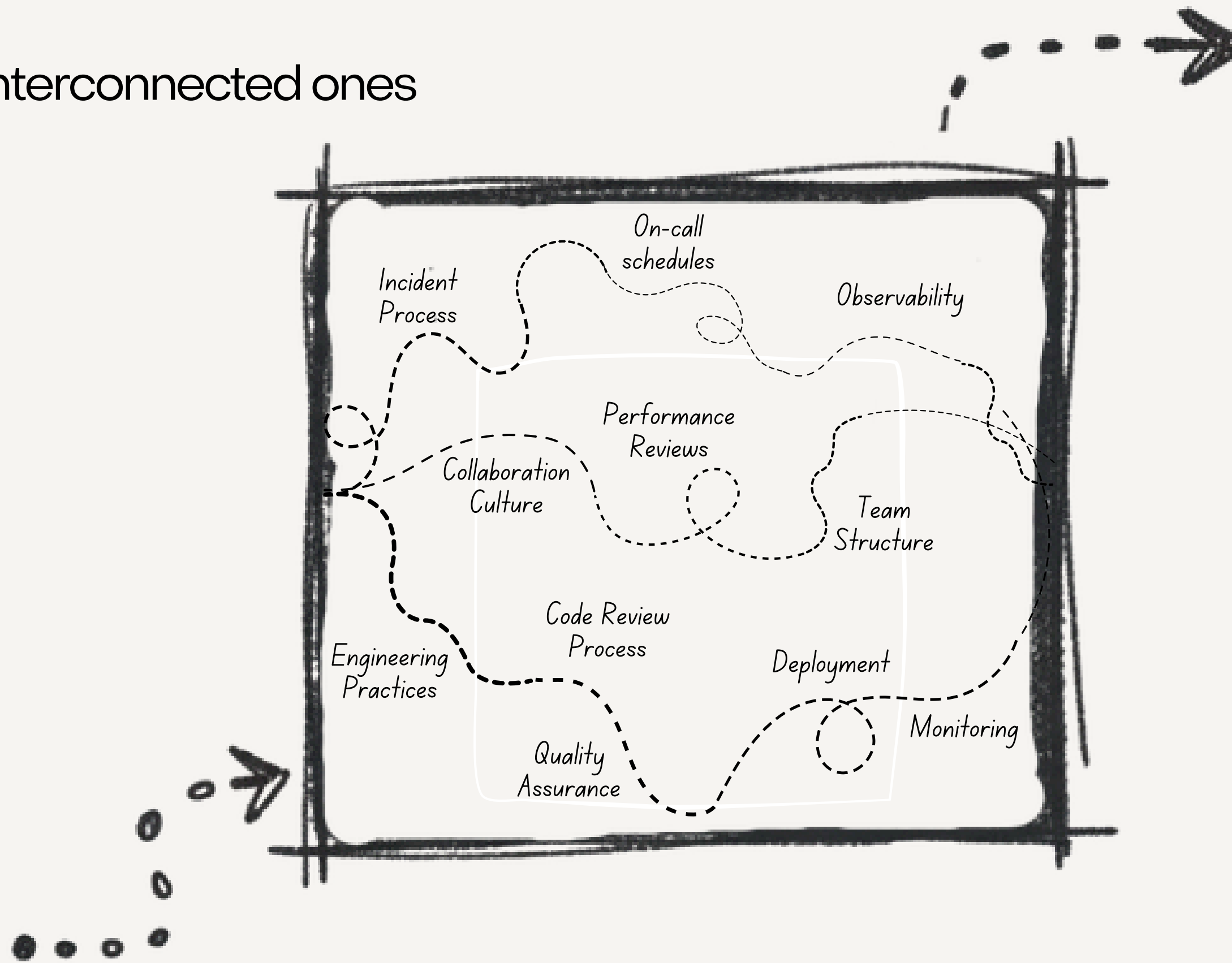
YOUR TEAM
AS A SYSTEM

Customer
Needs

But multiple interconnected ones

Business Outcomes

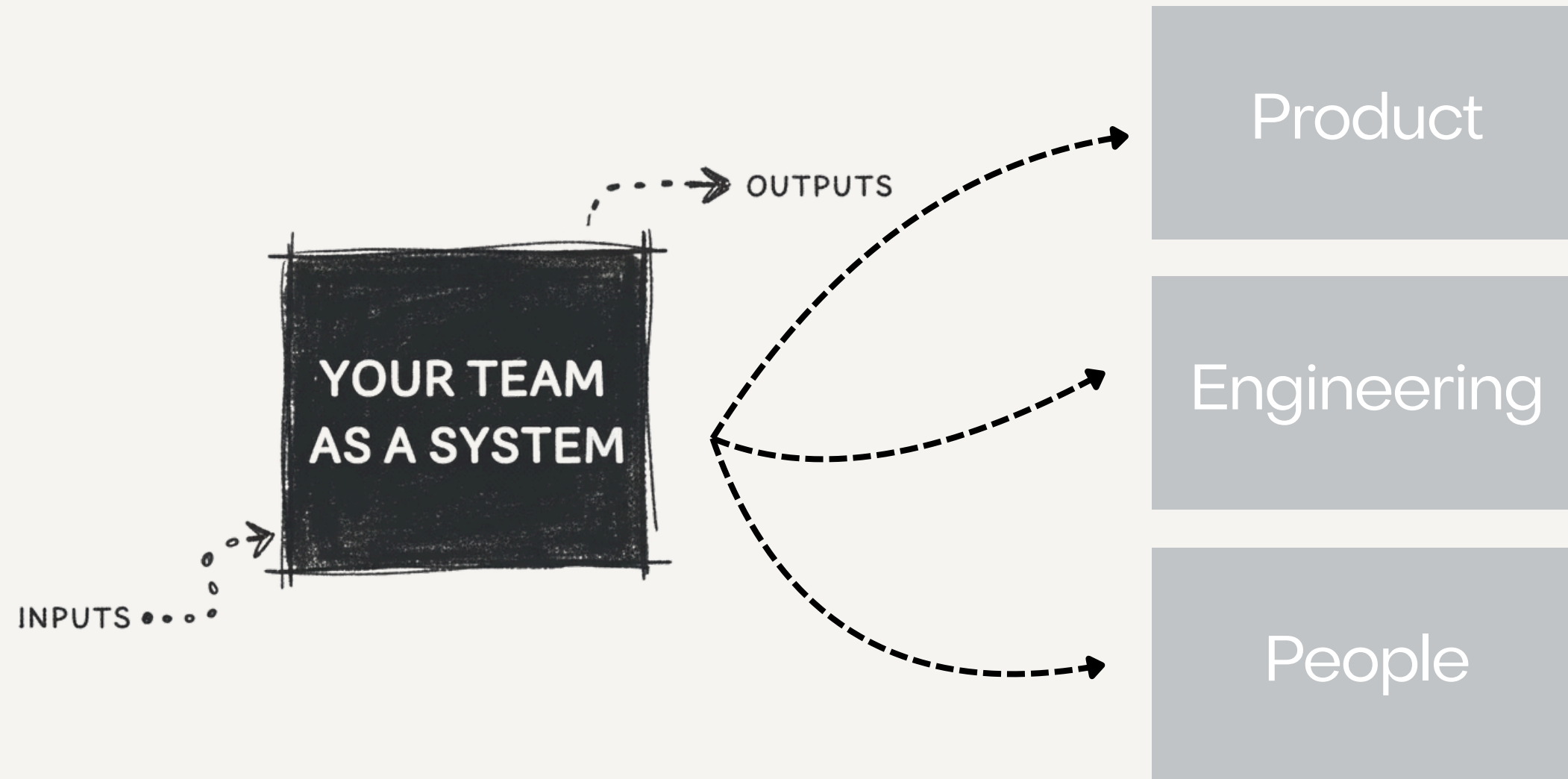
Customer Needs



In practice, to be a micro CTO



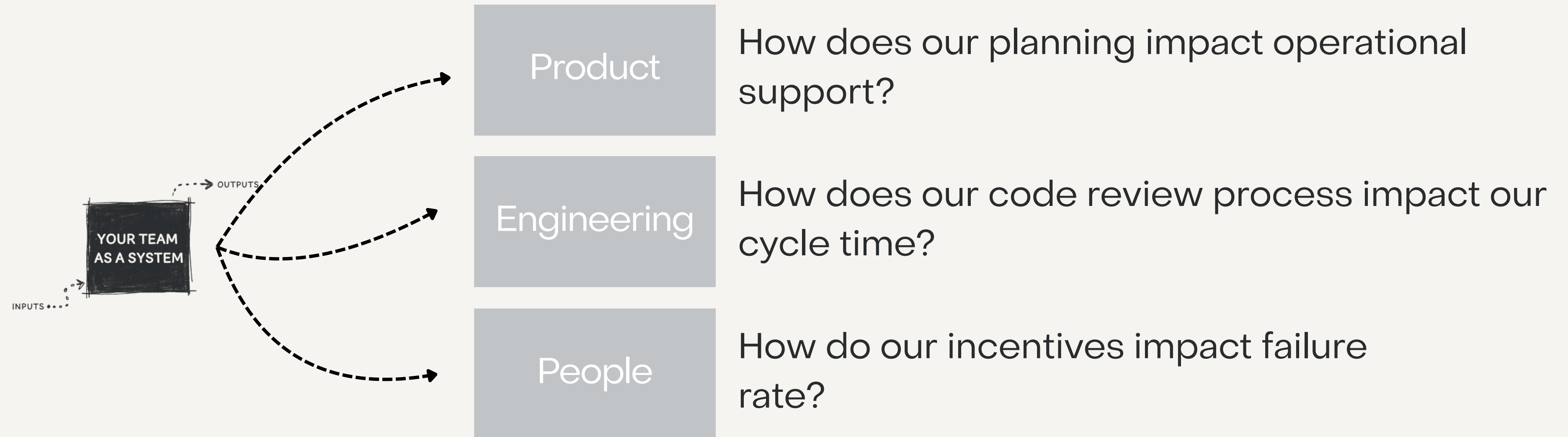
In practice, to be a micro CTO
you need to understand areas needed for success



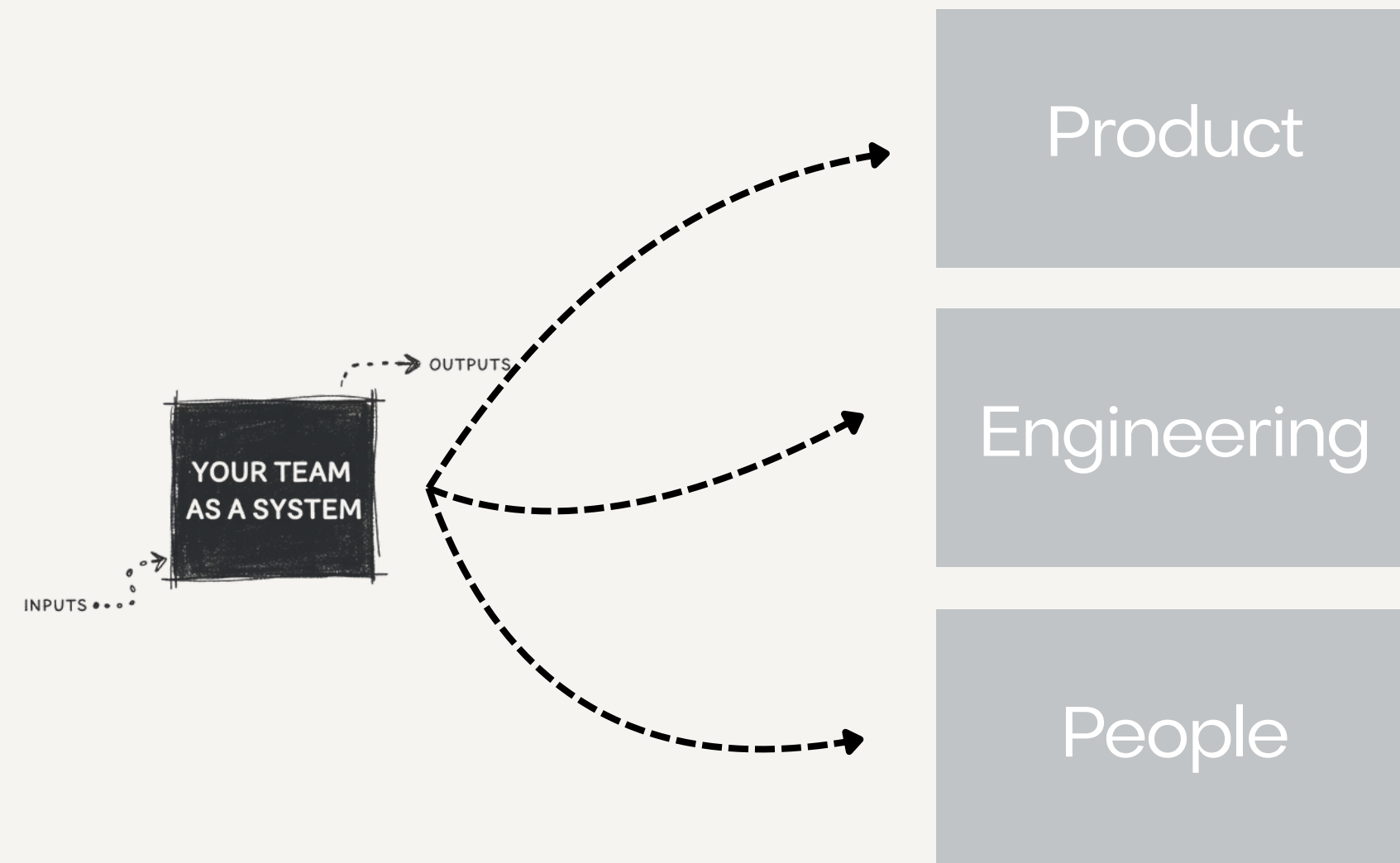
In practice, to be a micro CTO
you need to understand areas needed for success



In practice, to be a micro CTO
you need to understand areas needed for success
and how they interact



In practice, to be a micro CTO
you need to understand areas needed for success
and how they interact



Manage **the whole**, not the parts

DID IT IMPROVE?

THE STORY OF A
**VERY
SUPPORTIVE
MANAGER**



THE STORY OF A **VERY SUPPORTIVE MANAGER**



DID IT IMPROVE?

It did.

Once the manager moved his focus **away** from individuals and **towards** team effectiveness

- The team became more effective
- The company perceived the impact
- Individuals were rewarded

ANOTHER STORY, OF A **VERY** **HELPFUL** **MANAGER**



ANOTHER STORY,
OF A **VERY**
HELPFUL
MANAGER



They lead a platform team, that kept getting change requests from other product teams

The manager wanted to help everyone, so took every ask

And started contributing directly as well to help out

ANOTHER STORY, OF A **VERY** **HELPFUL** **MANAGER**



Soon, all the engineers (and the manager) were overwhelmed.

And the manager couldn't change anything, because they were also too busy writing code.

**HOW DID WE IMPROVE
IT?**

ANOTHER STORY,
OF A **VERY
HELPFUL
MANAGER**



ANOTHER STORY, OF A **VERY HELPFUL MANAGER**



HOW DID WE IMPROVE IT?

Once the manager focused on improving the **prioritization** system.

- The process was clearer for everyone
- The team was able to say no
- Team members ended up with a better work balance

Thinking in parts leads
to **reactive**
management

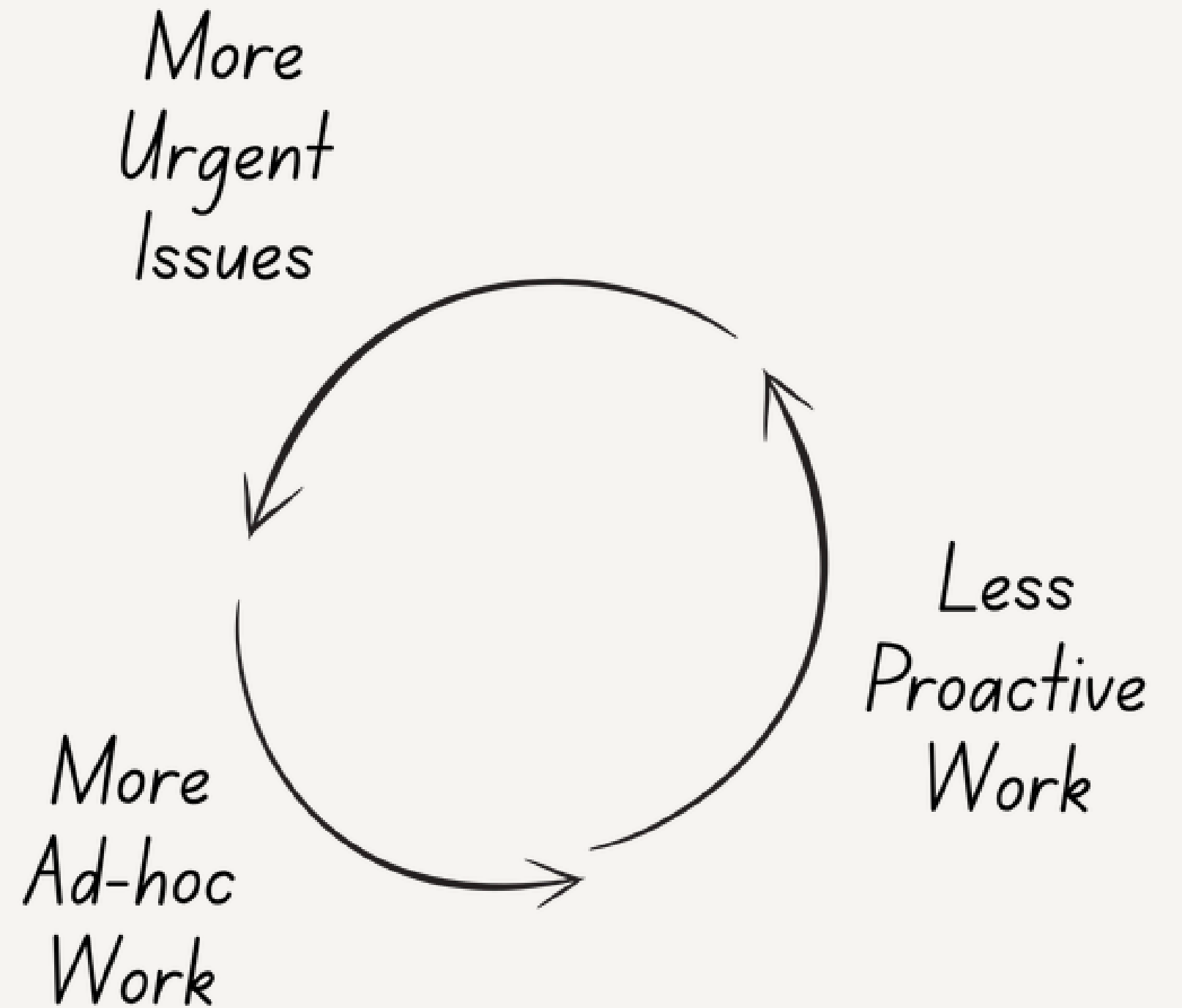
I need to improve
someone's
performance

I need to change how
our process works

Thinking in parts leads
to **reactive**
management

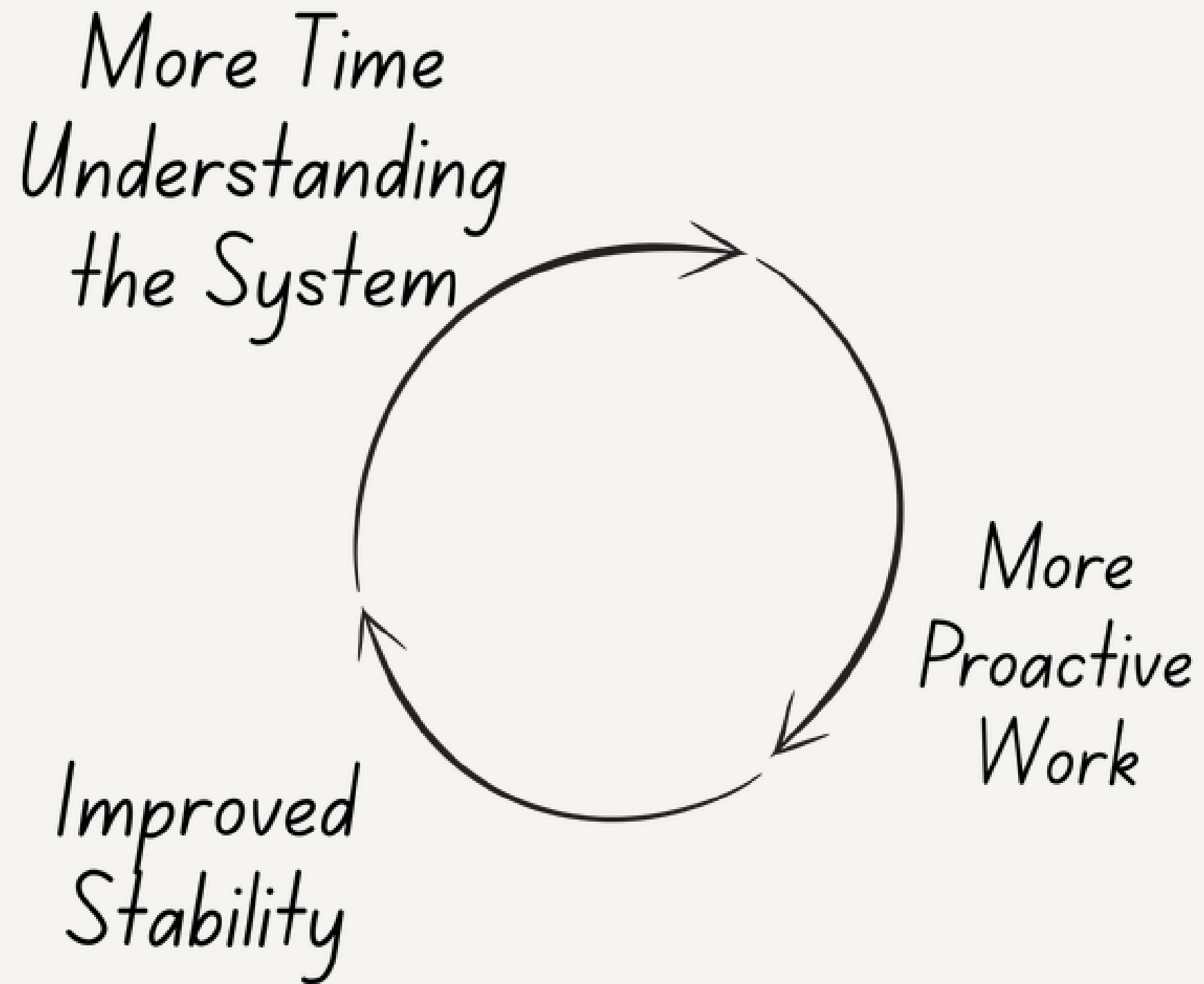
I need to improve
someone's
performance

I need to change how
our process works



Thinking about the whole leads to **proactive** leadership

I can improve how my team works so that everyone can perform well.

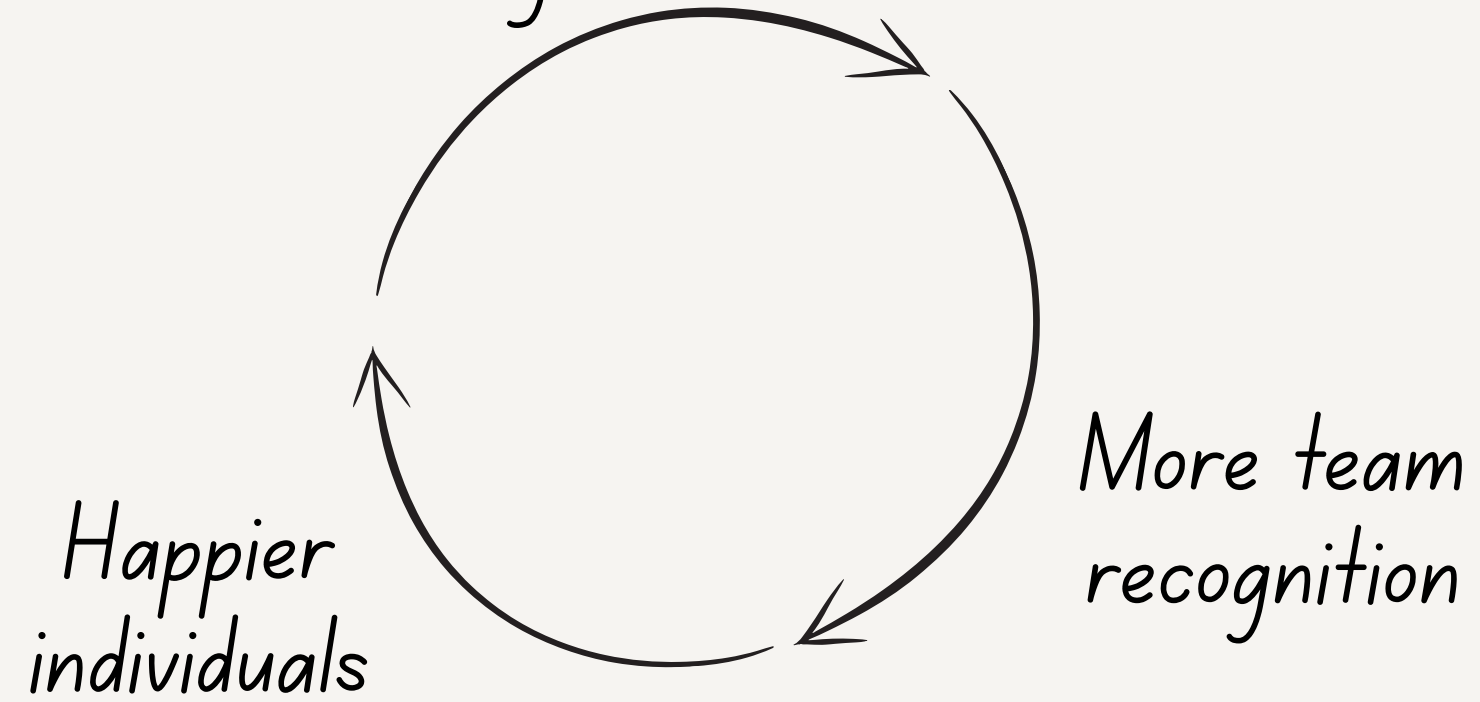


Thinking about the whole leads to **proactive** leadership

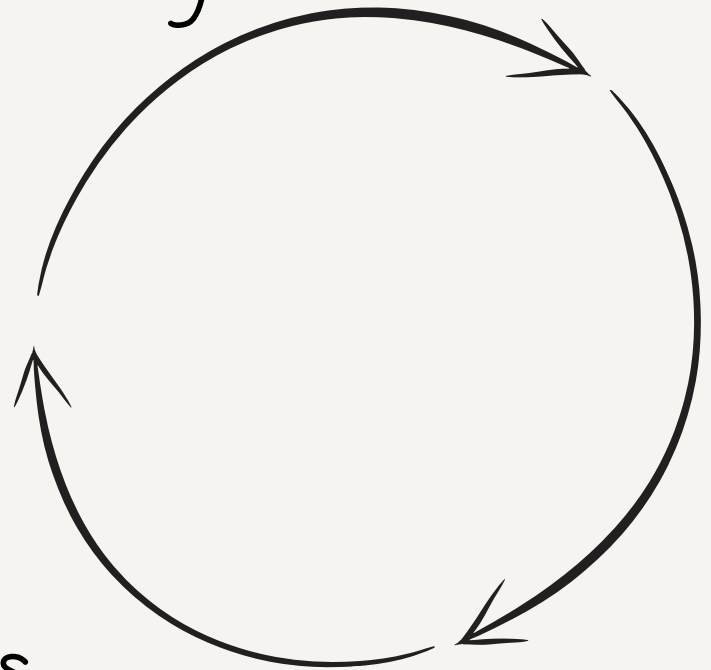
I can improve how my team works so that everyone can perform well.



More focus on
team delivery



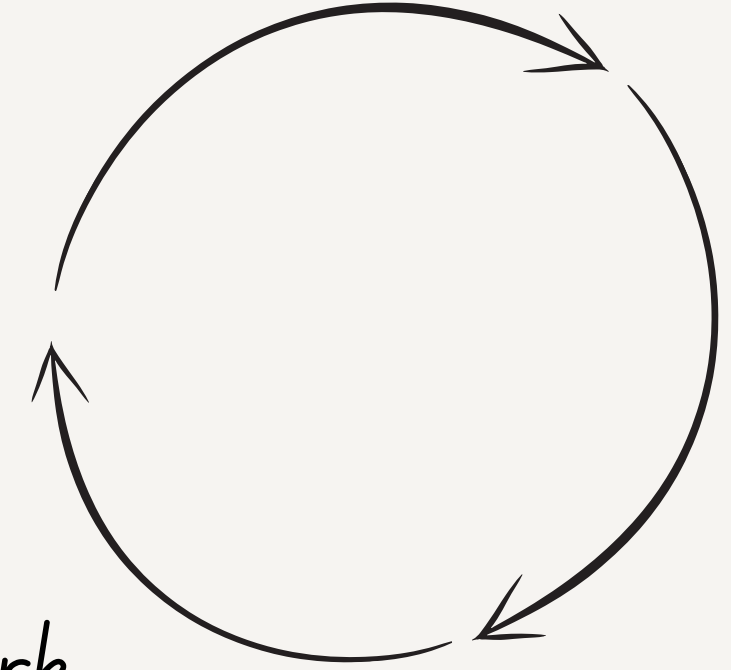
More focus on team delivery



Happier individuals

More team recognition

Stronger focus on prioritization



Less overwork

More refined backlog



MY STORY, AN EAGER **HEAD OF ENGINEERING**



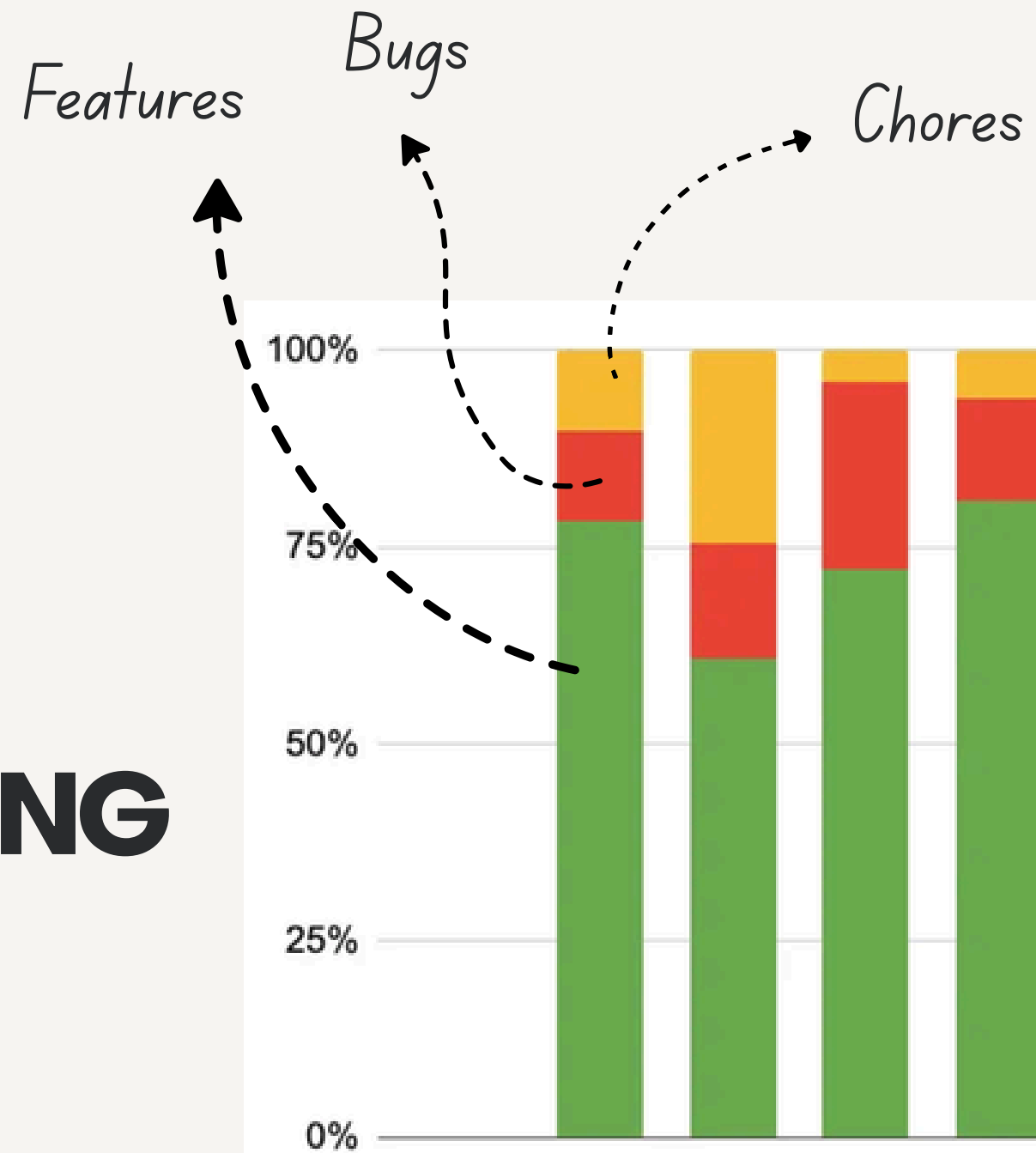
MY STORY,
AN EAGER
**HEAD OF
ENGINEERING**



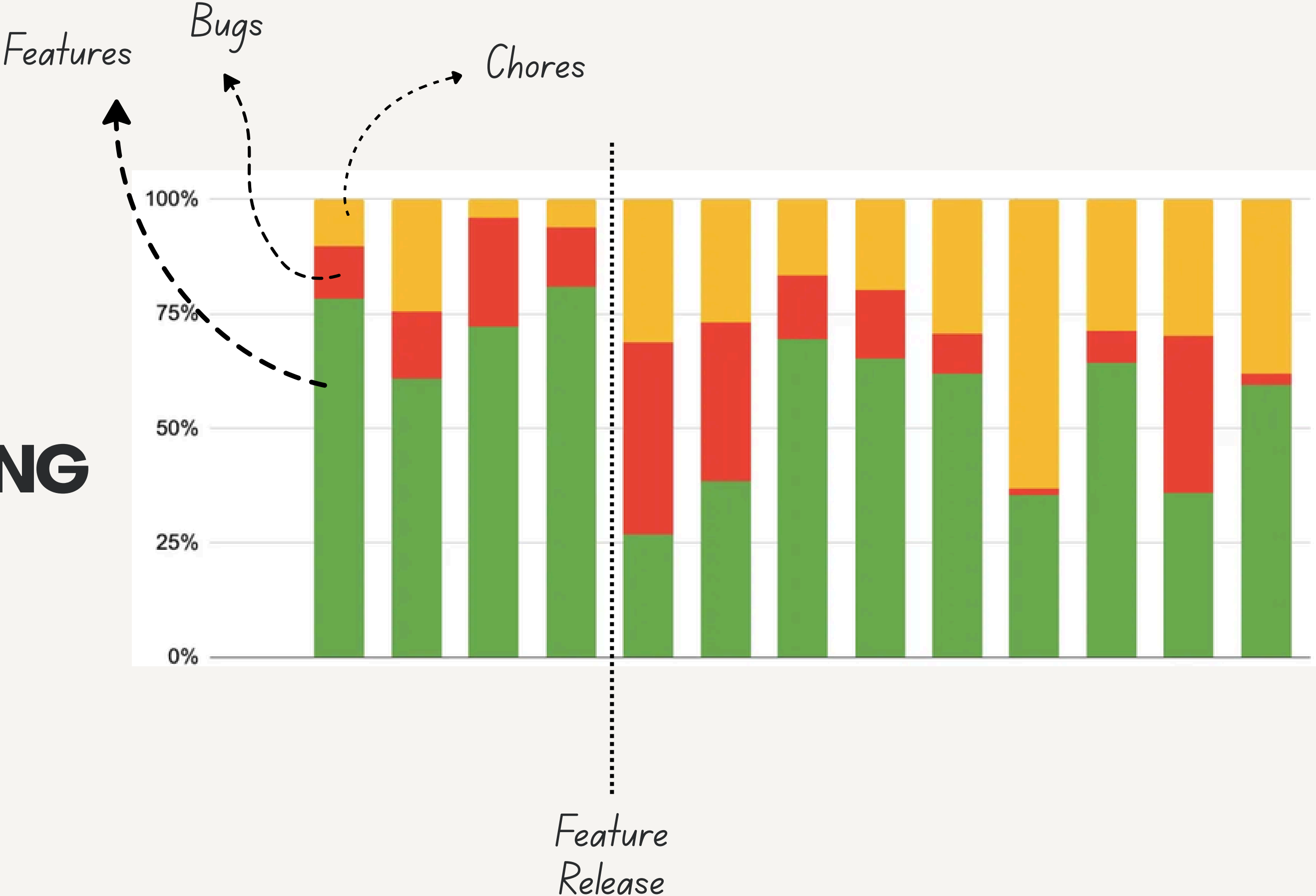
Leading a startup that delivered features at a very fast pace

Where engineers could not get technical and operational work prioritized

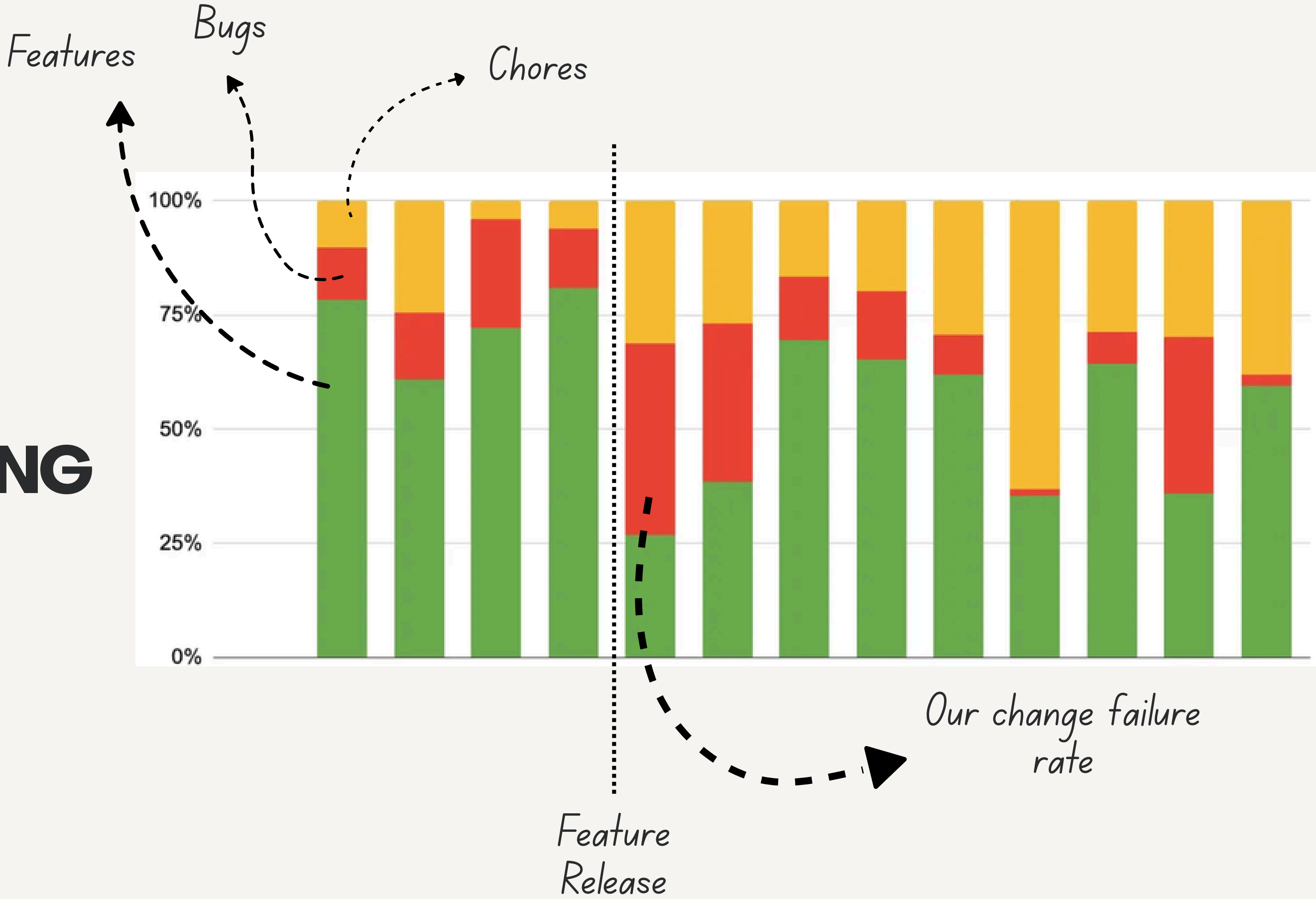
MY STORY, AN EAGER HEAD OF ENGINEERING



MY STORY, AN EAGER HEAD OF ENGINEERING



MY STORY, AN EAGER HEAD OF ENGINEERING



MY STORY,
AN EAGER
**HEAD OF
ENGINEERING**



Making the quality tradeoff visible to the CEO changed the way we planned.

Better planning created a more stable system.

A stable system allowed for engineering investment.



“A bad system will beat a good person every time.”

W. Deming



You need to be a **Micro-CTO**
to build a better system



THANK YOU!

FRANCISCO TRINDADE

franciscotrindade.me

