



Creating Technical Leadership in Context

Joy Zheng & Nathan Tindall
Plaid, Inc.

StaffPlus NYC 2024



Who here can remember a time they started a staff+ job in an area?



Been asked “what’s our technical strategy”?



Or “how do we fix our systems?”



And felt like you had no idea what you were doing?

(us too)



Who's in the middle of this experience *right now*?



PLAID



1. Storytime + reflections



2. Framework



1. Storytime + reflections



2. Framework



What can only engineering do?

Build great technology

to

Support the evolution of the business

Support the evolution of the business



by

Build great technology



Support the evolution of the business



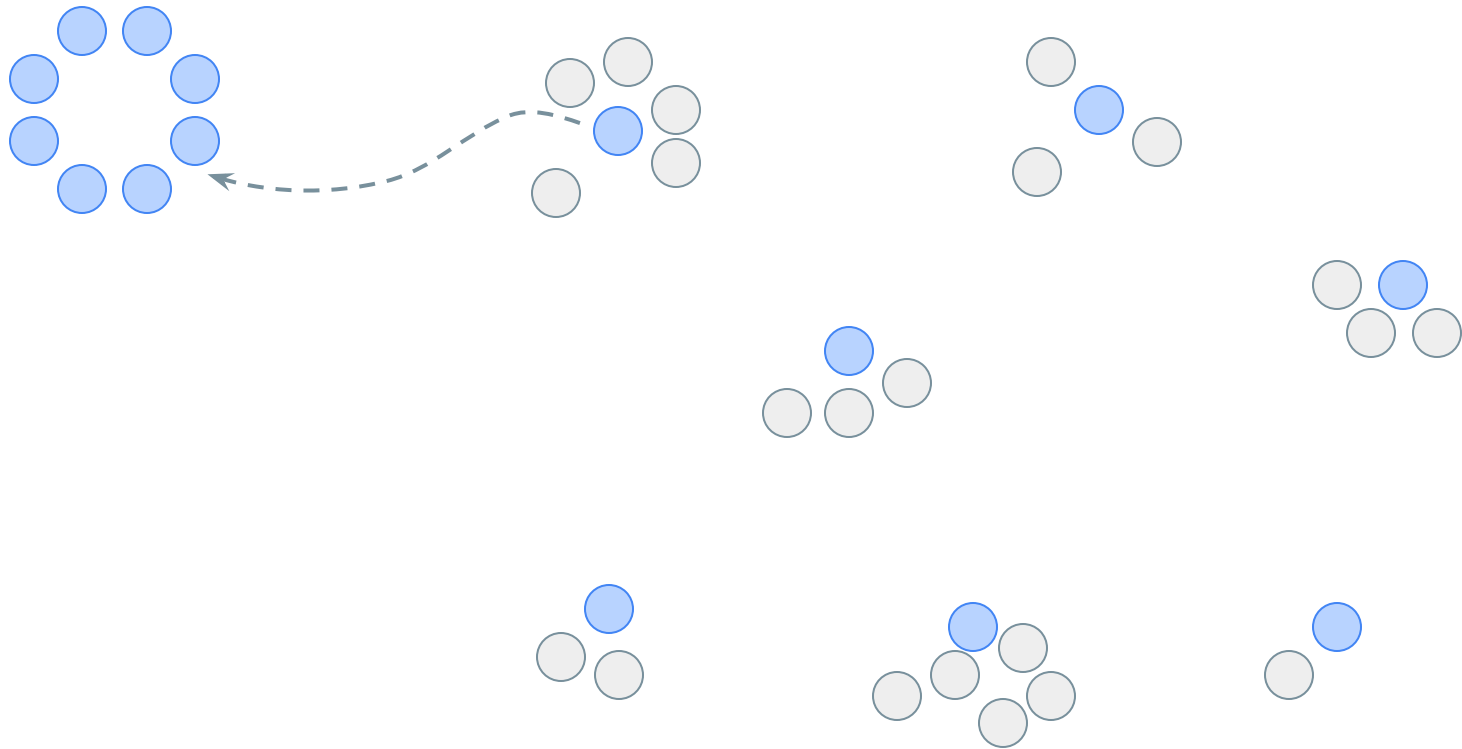
by



Build great technology



Reflections on 3(ish) iterations over time



30-80 Eng

2018-2020: Backend Council

ITERATION 0

ITERATION 1

ITERATION 2

Context #1: Growing eng team

ITERATION 0

ITERATION 1

ITERATION 2



Cross-team brainstorm

ITERATION 0

ITERATION 1

ITERATION 2

Context #2: Microservices(ish) Architecture

ITERATION 0

ITERATION 1

ITERATION 2



Backend technical principles for services

ITERATION 0

ITERATION 1

ITERATION 2

Context #3: Change in sources of system complexity

ITERATION 0

ITERATION 1

ITERATION 2



System tangle driven by newer products

ITERATION 0

ITERATION 1

ITERATION 2



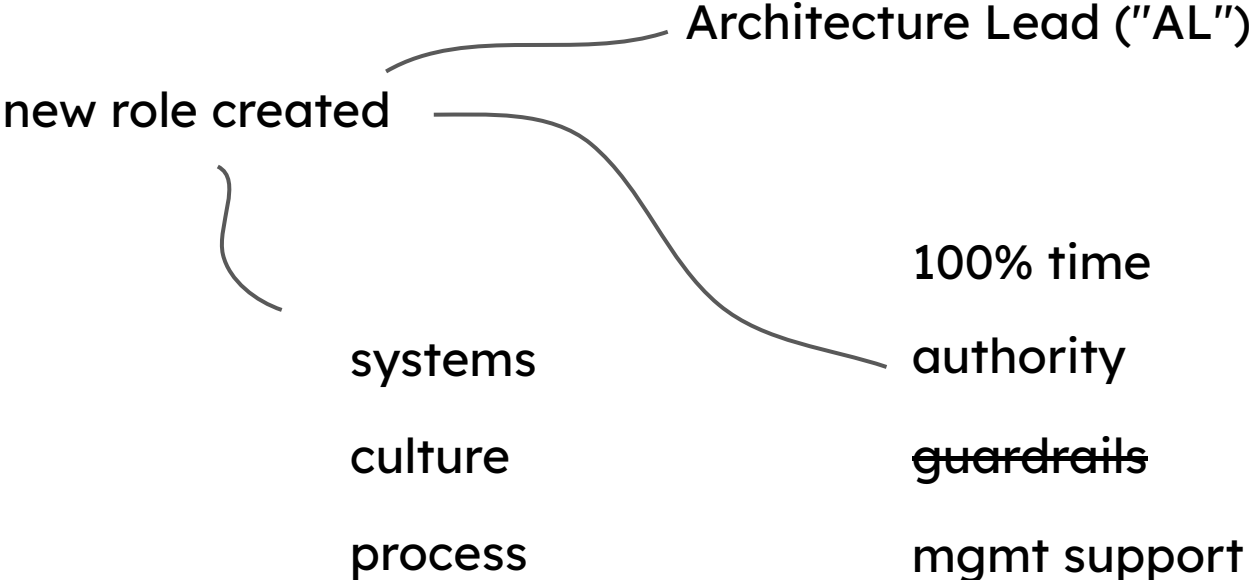
aka the Bermuda triangle

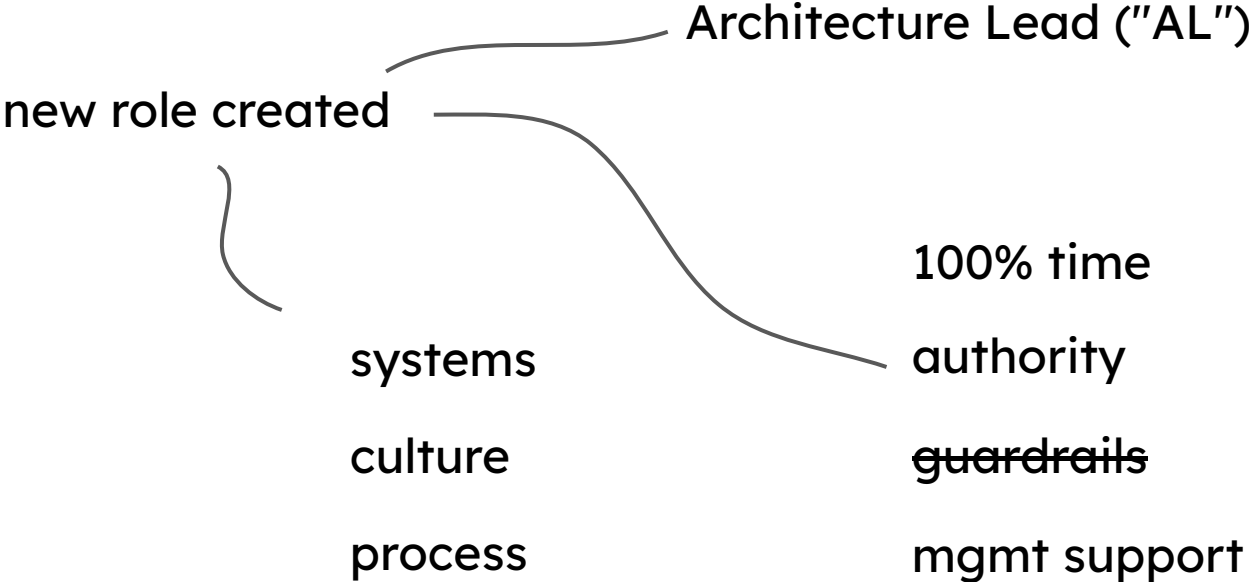
ITERATION 0

ITERATION 1

ITERATION 2

SOUND THE ALARM



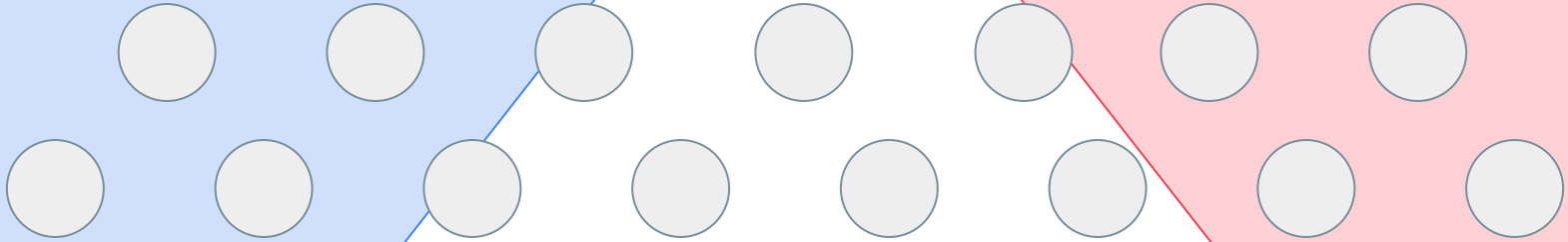


ITERATION 0

ITERATION 1

ITERATION 2

We created a supporting structure



100-200 Eng

2020-2021. Architecture Leads + Architecture Council

ITERATION 0

ITERATION 1

ITERATION 2

Context #1: System complexity popping up in new areas

ITERATION 0

ITERATION 1

ITERATION 2



Kick off a few key system re-architectures

ITERATION 0

ITERATION 1

ITERATION 2

Context #2: Decentralized eng organization

ITERATION 0

ITERATION 1

ITERATION 2



Missing eng processes

ITERATION 0

ITERATION 1

ITERATION 2

We shipped process before understanding the problem

ITERATION 0

ITERATION 1

ITERATION 2



Trying to recreate other company processes in isolation is ineffective

ITERATION 0

ITERATION 1

ITERATION 2

Context #3: Unclear how to spend time

ITERATION 0

ITERATION 1

ITERATION 2



Wasted time gap-filling for missing teams

ITERATION 0

ITERATION 1

ITERATION 2



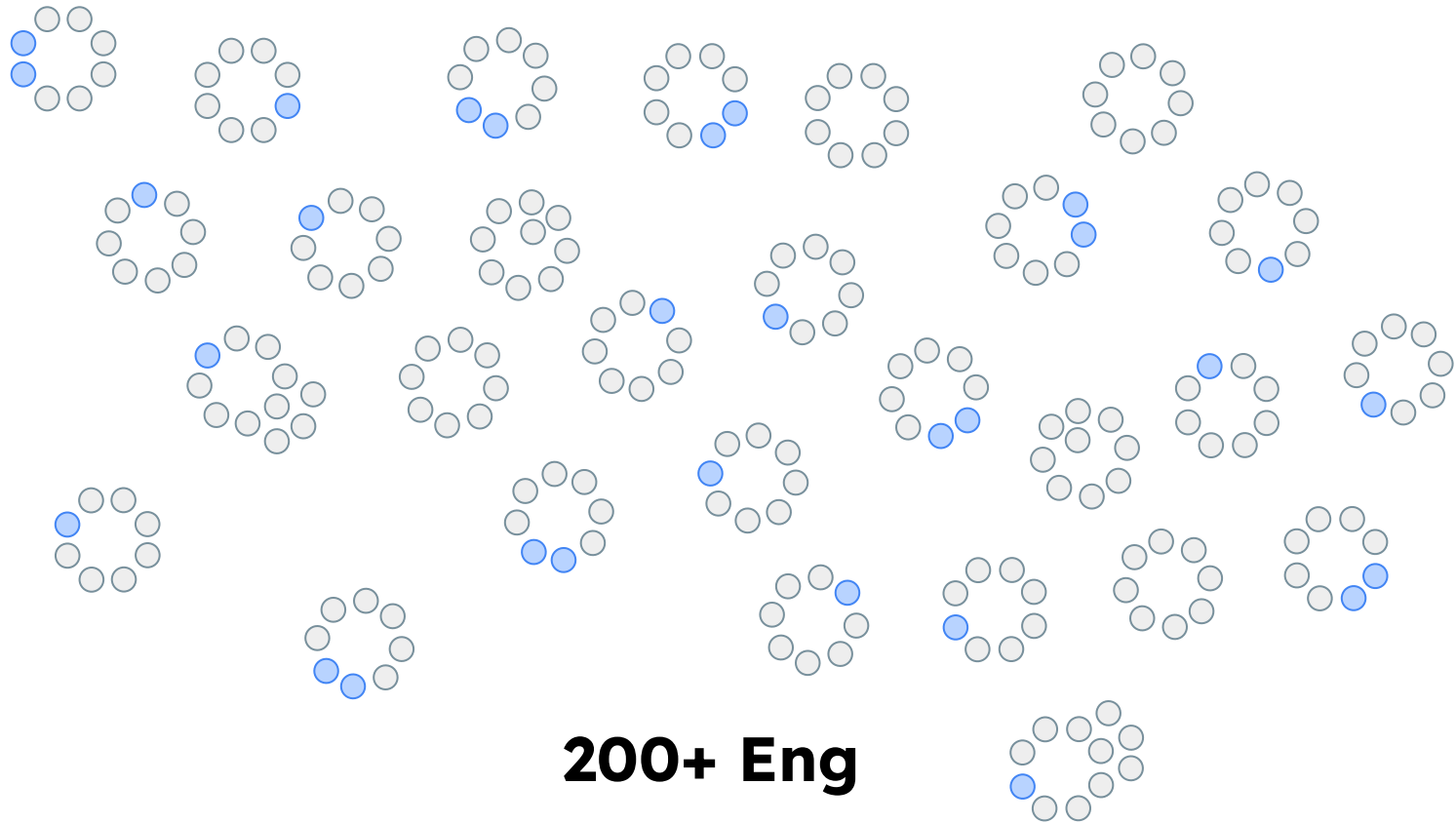
Scaling cluelessness

ITERATION 0

ITERATION 1

ITERATION 2

Out of band prioritization



2023+. Senior ICs

ITERATION 0

ITERATION 1

ITERATION 2

Context #1: Unclear how to spend time

ITERATION 0

ITERATION 1

ITERATION 2



Senior ICs aligned with business problems

ITERATION 0

ITERATION 1

ITERATION 2

Context #2: Decentralized eng organization

ITERATION 0

ITERATION 1

ITERATION 2



Lighter-weight, targeted processes

ITERATION 0

ITERATION 1

ITERATION 2

Context #3: Senior ICs scattered across the org

ITERATION 0

ITERATION 1

ITERATION 2



Onsite & brainstorm

What did we learn from all of this?



What advice do we wish we give our 2020 selves?

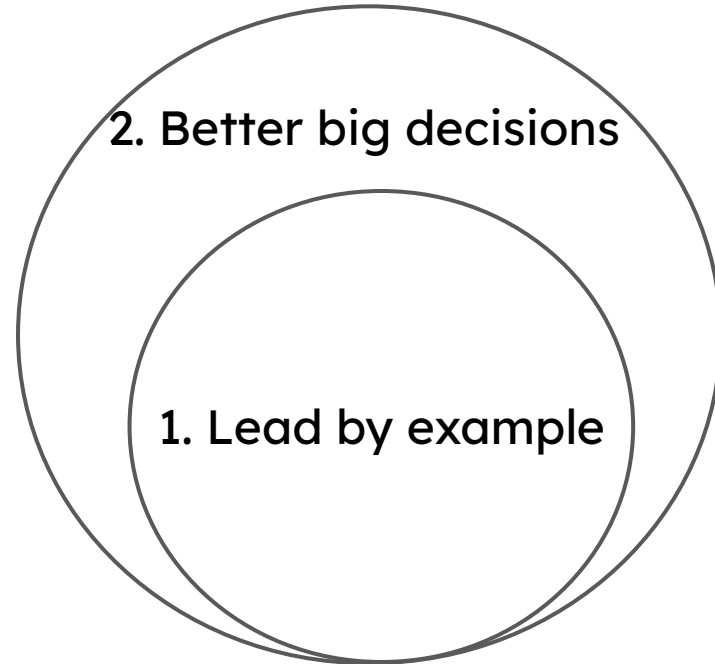
What advice do we wish we give our 2020 selves?

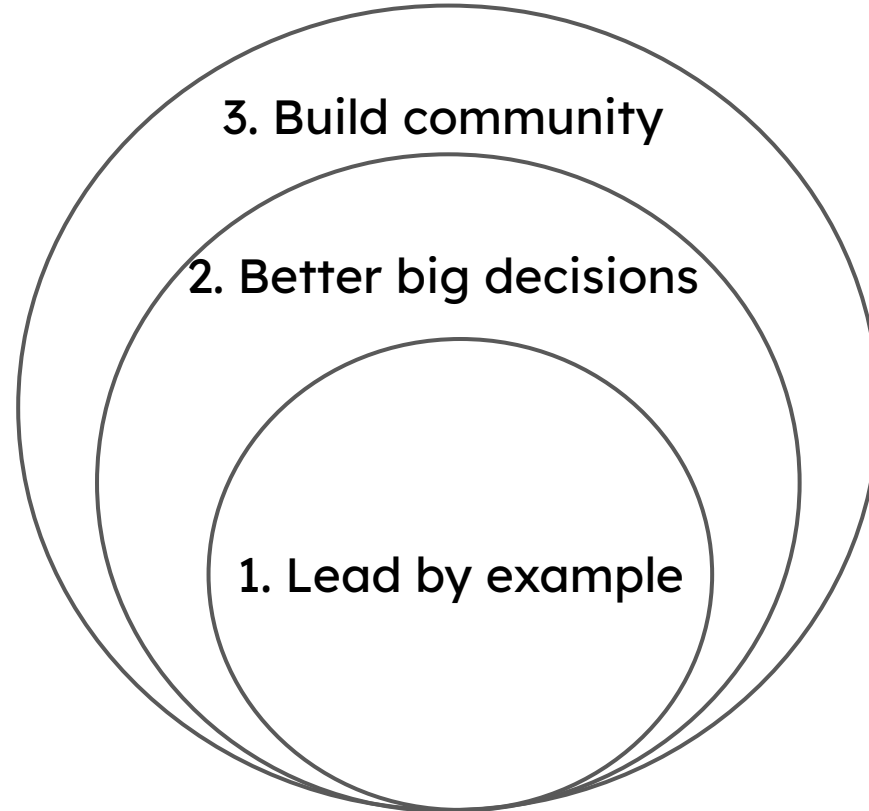
IT DEPENDS (™)

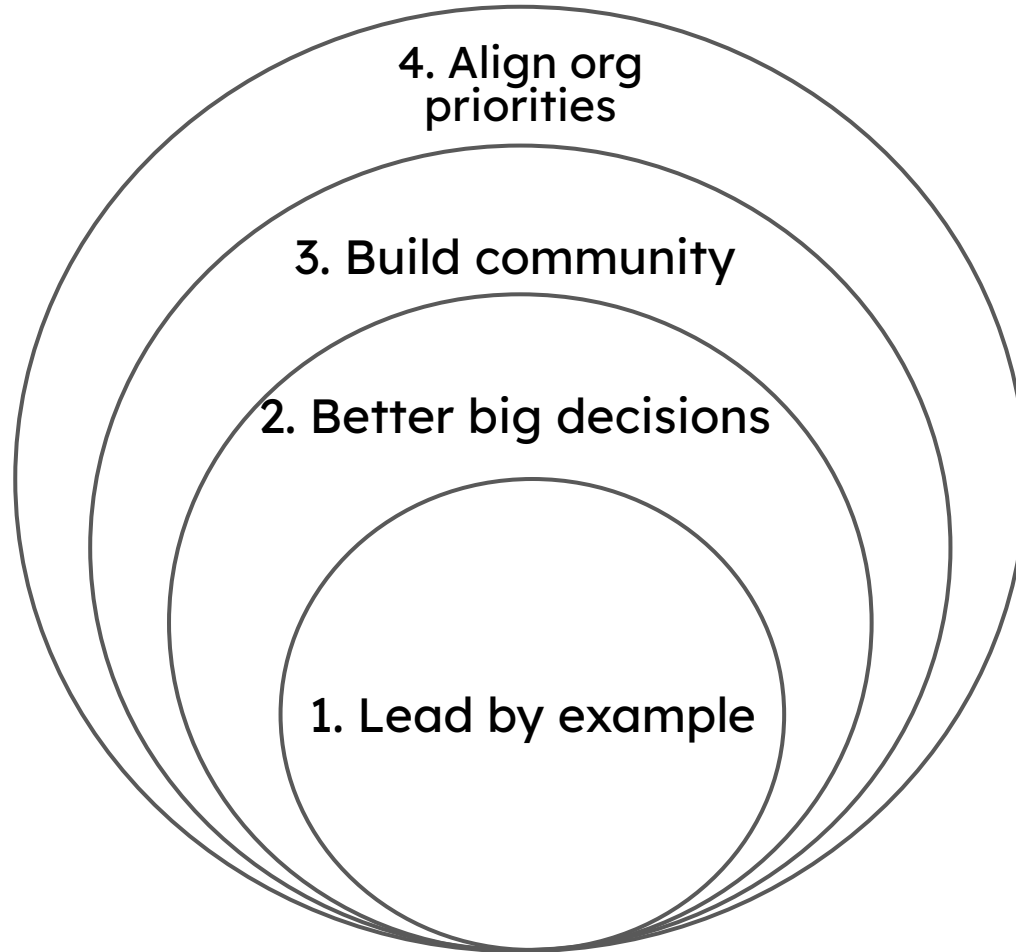
IT DEPENDS (™)

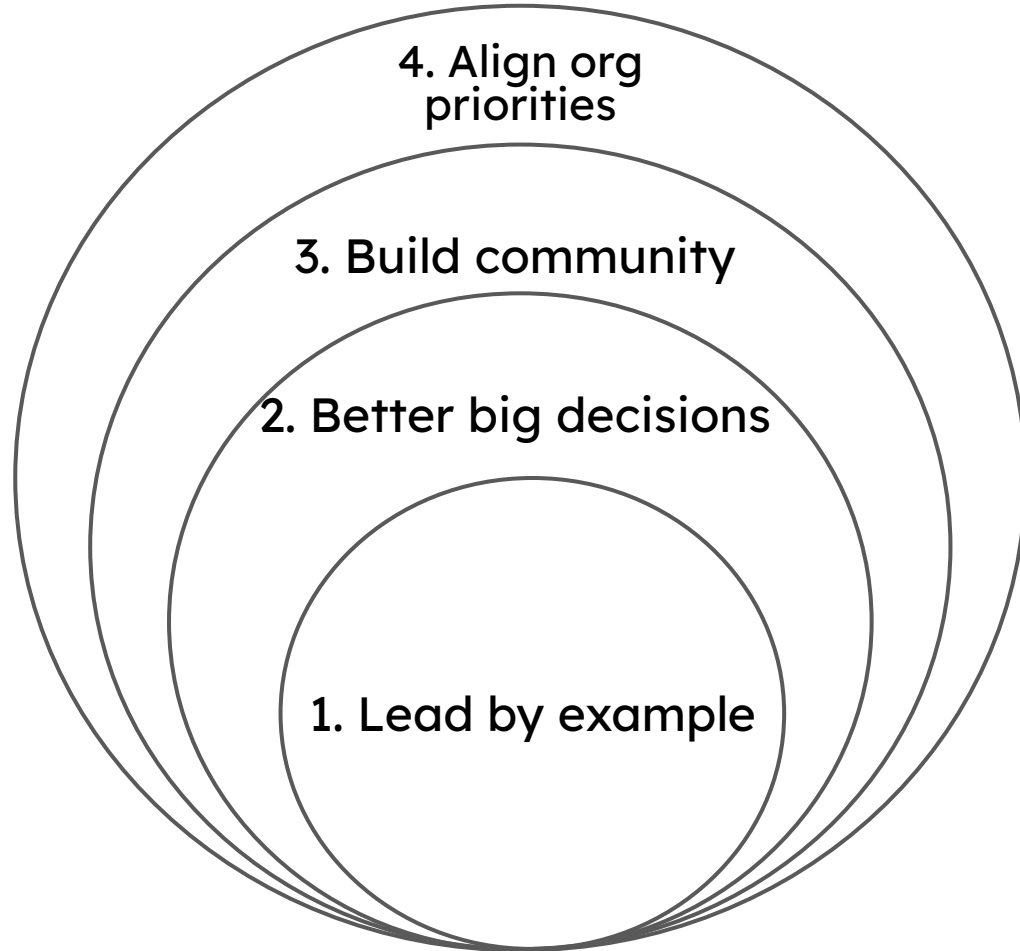


1. Lead by example









4. Align org priorities

3. Build community

2. Better big decisions

1. Lead by example



1. Lead by example

**1. LEAD BY
EXAMPLE**

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES



It's tempting to start at the broad level

**1. LEAD BY
EXAMPLE**

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES

This didn't work out well in practice for us

**1. LEAD BY
EXAMPLE**

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES

Focus on a problem statement where you can add leverage

**1. LEAD BY
EXAMPLE**

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES

Example 1: unstick systems slowing down key projects

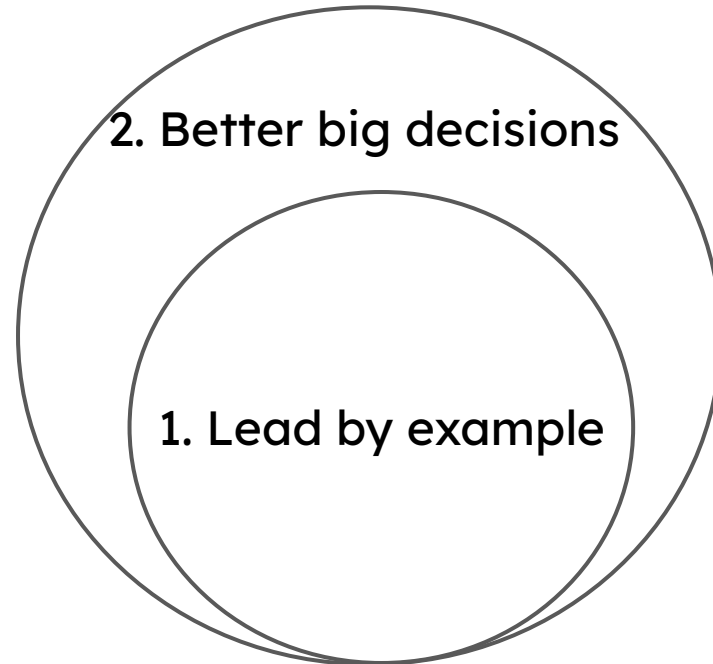
**1. LEAD BY
EXAMPLE**

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES

Example 2: accelerate key projects



1. LEAD BY
EXAMPLE

2. **BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES



We know some **BIG DECISIONS** not getting made...

1. LEAD BY
EXAMPLE

**2. BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES

... which leads to incidental system complexity

1. LEAD BY
EXAMPLE

**2. BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES

We can do better

1. LEAD BY
EXAMPLE

**2. BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES

Diagnosis: why are these **BIG DECISIONS** not getting made?

1. LEAD BY
EXAMPLE

2. **BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES



Scenario 1:
We don't know how
to make a decision

1. LEAD BY
EXAMPLE

2. **BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES



Scenario 1:
We don't know how
to make a decision



Context:
Conflict-avoidant
eng culture

1. LEAD BY
EXAMPLE

2. **BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES



Scenario 1:
We don't know how
to make a decision



Context:
Conflict-avoidant
eng culture



Our Solution:
RFC Process

1. LEAD BY
EXAMPLE

2. **BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES



Scenario 2:
We're making and
ignoring decisions

1. LEAD BY
EXAMPLE

2. **BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES



Scenario 2:
We're making and
ignoring decisions



Context:
No regular forum to
circulate knowledge

1. LEAD BY
EXAMPLE

**2. BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES



Scenario 2:
We're making and
ignoring decisions



Context:
No regular forum to
circulate knowledge



Our Solution:
Engineering Review

1. LEAD BY
EXAMPLE

2. **BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES



Scenario 3:
We're making bad
decisions (and then
ignoring them)

1. LEAD BY
EXAMPLE

2. **BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES



Scenario 3:
We're making bad
decisions (and then
ignoring them)



Context:
Big decisions getting
buried in the details

1. LEAD BY
EXAMPLE

2. **BETTER BIG
DECISIONS**

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES



Scenario 3:
We're making bad
decisions (and then
ignoring them)



Context:
Big decisions getting
buried in the details



Our Solution:
Call out big
decisions in spec
template

1. LEAD BY
EXAMPLE

2. **BETTER BIG
DECISIONS**

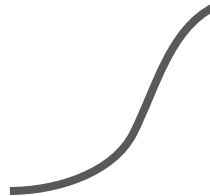
3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES

Diagnose

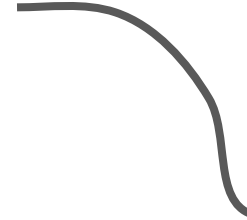


Standardize
(RFC Framework)

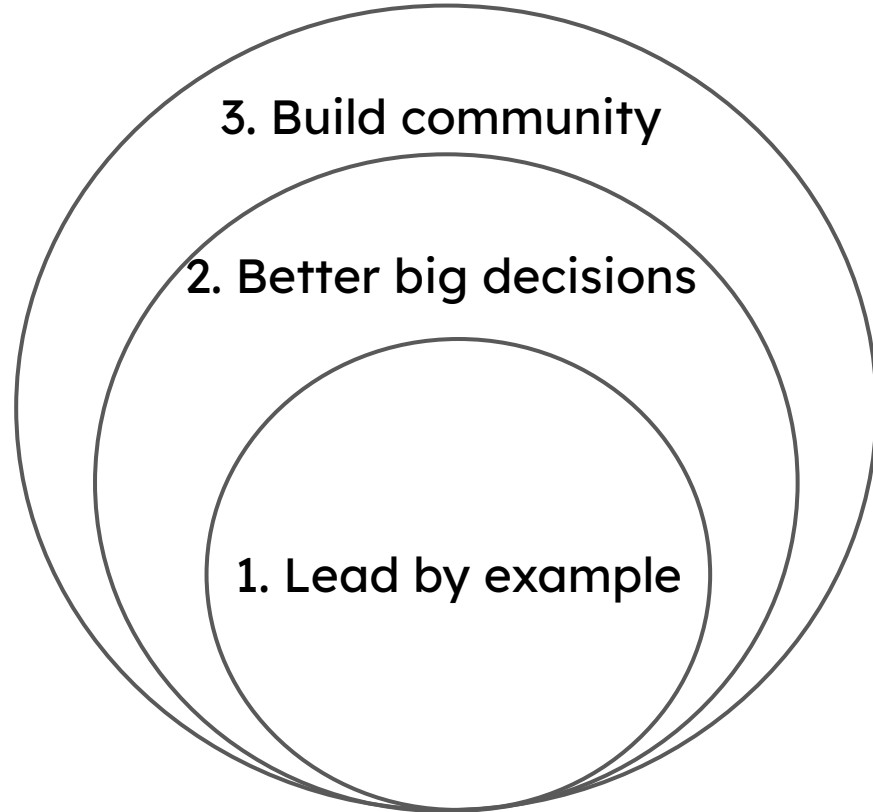


(Eng Review)

Socialize



Coach
(Template + Culture)

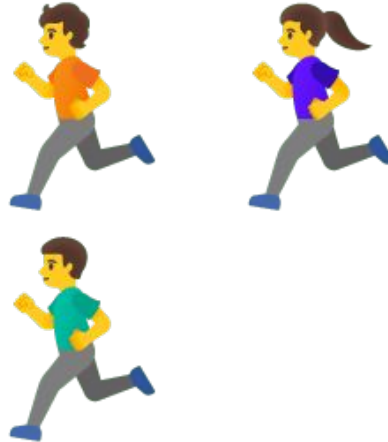


1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

**3. BUILD
COMMUNITY**

4. ALIGN ORG
PRIORITIES



There are engineers across the company that have the energy to create change

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

**3. BUILD
COMMUNITY**

4. ALIGN ORG
PRIORITIES



... but they aren't talking to each other enough

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

**3. BUILD
COMMUNITY**

4. ALIGN ORG
PRIORITIES

So now they're frustrated that problems are going unsolved

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

**3. BUILD
COMMUNITY**

4. ALIGN ORG
PRIORITIES



Channel this energy by getting people together

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

**3. BUILD
COMMUNITY**

4. ALIGN ORG
PRIORITIES

Curate the important topics

1. LEAD BY
EXAMPLE

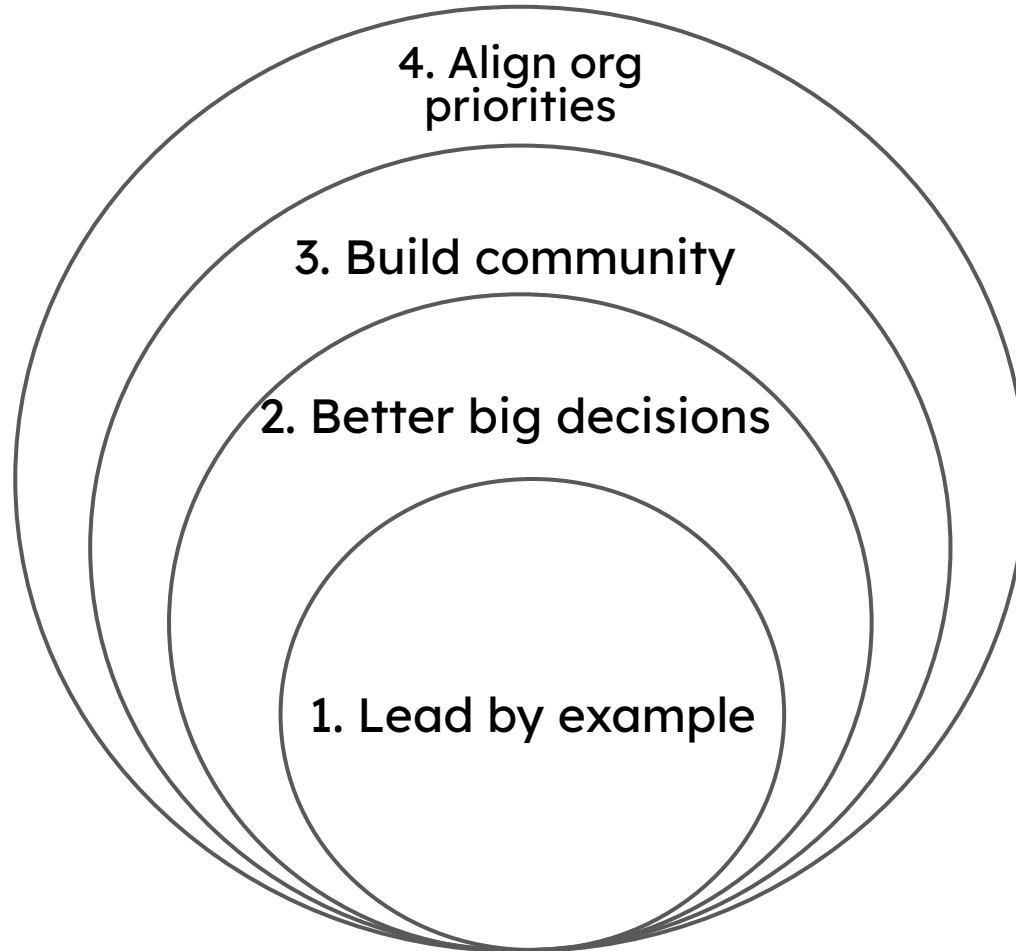
2. BETTER BIG
DECISIONS

**3. BUILD
COMMUNITY**

4. ALIGN ORG
PRIORITIES



So that the lurking **BIG DECISIONS** get surfaced



1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

**4. ALIGN ORG
PRIORITIES**

There are still dire problems not being prioritized

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

**4. ALIGN ORG
PRIORITIES**

Two patterns preventing resourcing by decision makers

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

**4. ALIGN ORG
PRIORITIES**



Diagnosis 1:
Hidden problem



Solution:
Articulate problem,
success criteria, and
expertise needed

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

**4. ALIGN ORG
PRIORITIES**



Diagnosis 1:
Hidden problem



Solution:
Articulate problem,
success criteria, and
expertise needed

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

**4. ALIGN ORG
PRIORITIES**

Diagnosis 2:
Mysterious solution



Solution 1:
Prototype

Solution 2:
North Star

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

**4. ALIGN ORG
PRIORITIES**

Diagnosis 2:
Mysterious solution



Solution 1:
Prototype

Solution 2:
North Star

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

**4. ALIGN ORG
PRIORITIES**

Diagnosis 2:
Mysterious solution



Solution 1:
Prototype

Solution 2:
North Star

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

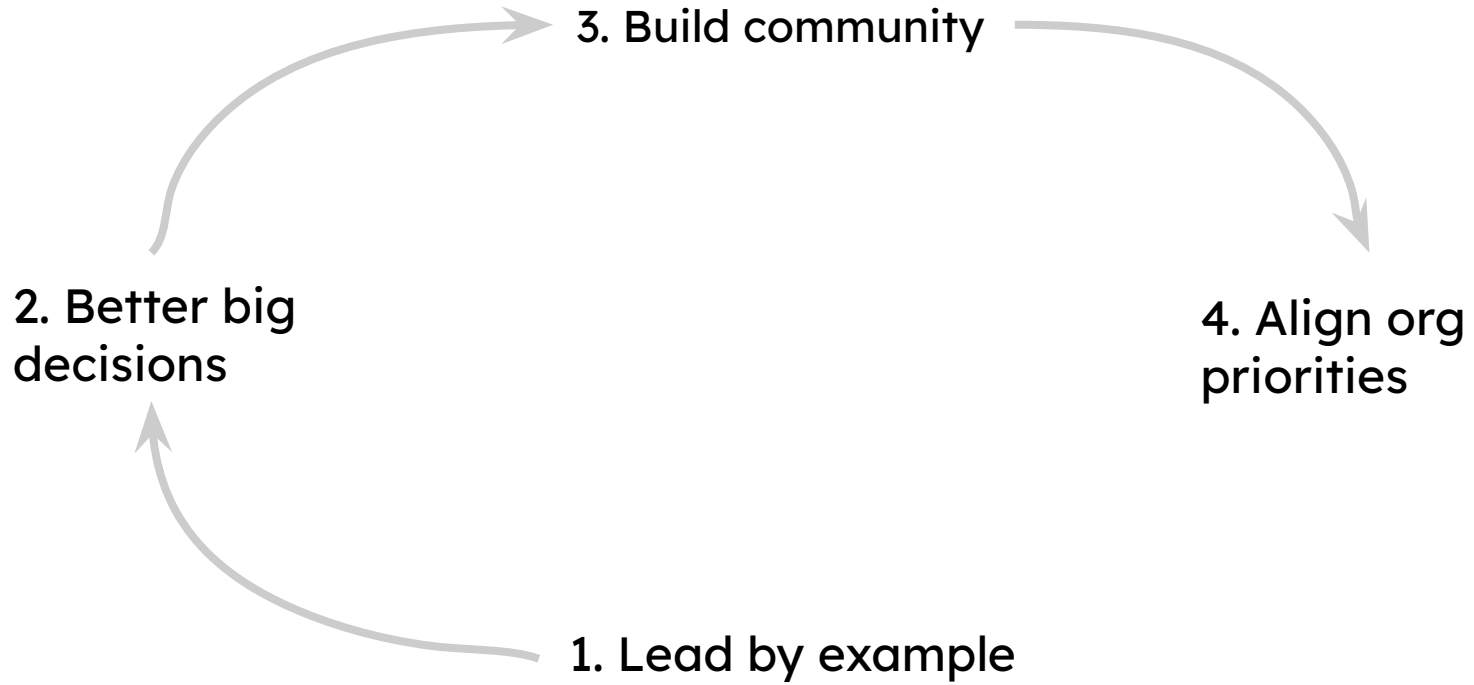
3. BUILD
COMMUNITY

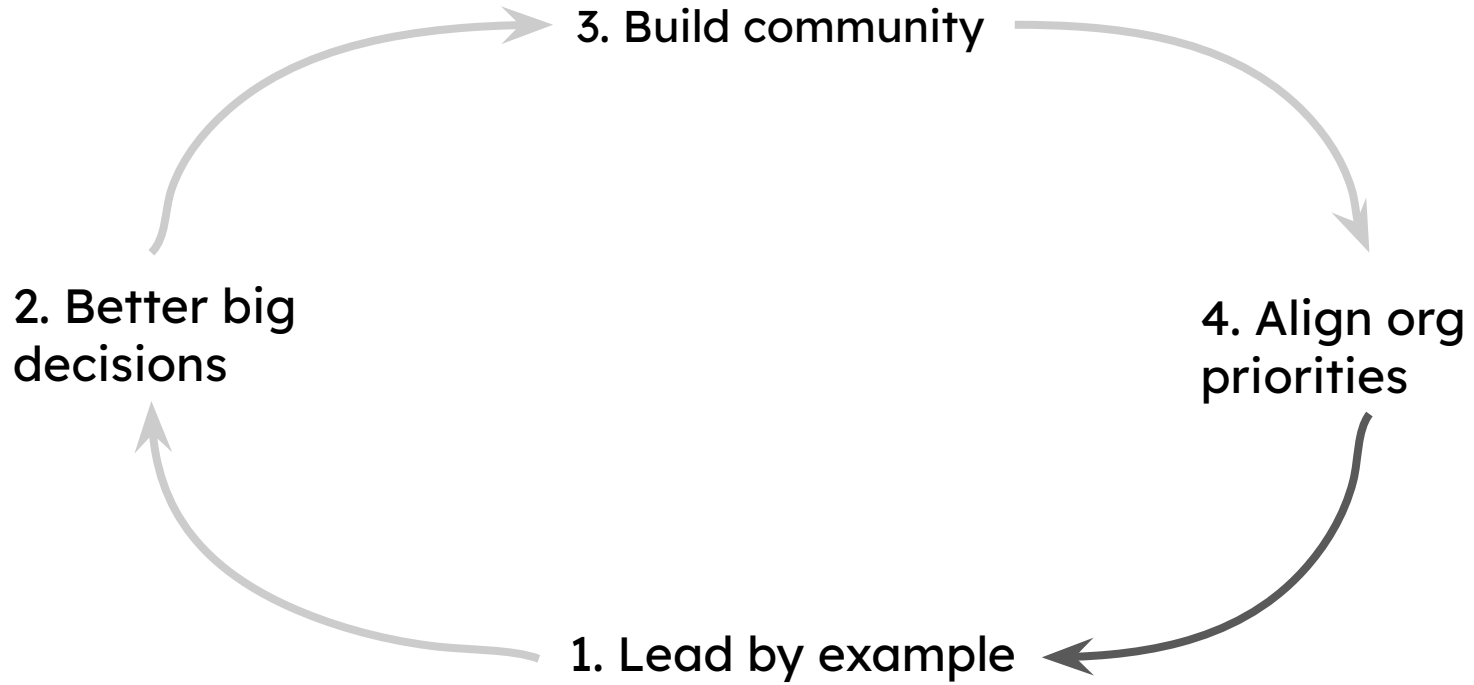
**4. ALIGN ORG
PRIORITIES**

**CLEARER PROBLEM
STATEMENTS**

DEMYSTIFIED SOLUTIONS

These are both ways in which you can uniquely add leverage.





1. LEAD BY EXAMPLE

focused > broad
unstick key projects
accelerate new opportunities

2. BETTER BIG DECISIONS

diagnose
standardize
socialize
coach

3. BUILD COMMUNITY

bring people together
curate topics
surface big decisions

4. ALIGN ORG PRIORITIES

state hidden problems
demystify solutions

Diagnose context to determine the right solution at each step above:

- Context on the evolution of the business
- Context on where the technology does(n't) support it

Don't scale cluelessness: scale solutions only after you have a strong POV on what success and failure looks like

INTENTIONALLY BLANK



TECHNICAL LEADERSHIP IN CONTEXT



WHAT: GENERIC FRAMEWORK

1. LEAD BY
EXAMPLE

2. BETTER BIG
DECISIONS

3. BUILD
COMMUNITY

4. ALIGN ORG
PRIORITIES

HOW: SPECIFIC CONTEXT

GREAT
TECHNOLOGY

EVOLUTION OF
THE BUSINESS

GREAT TECHNOLOGY

EVOLUTION OF THE BUSINESS

Support the evolution of the business



by

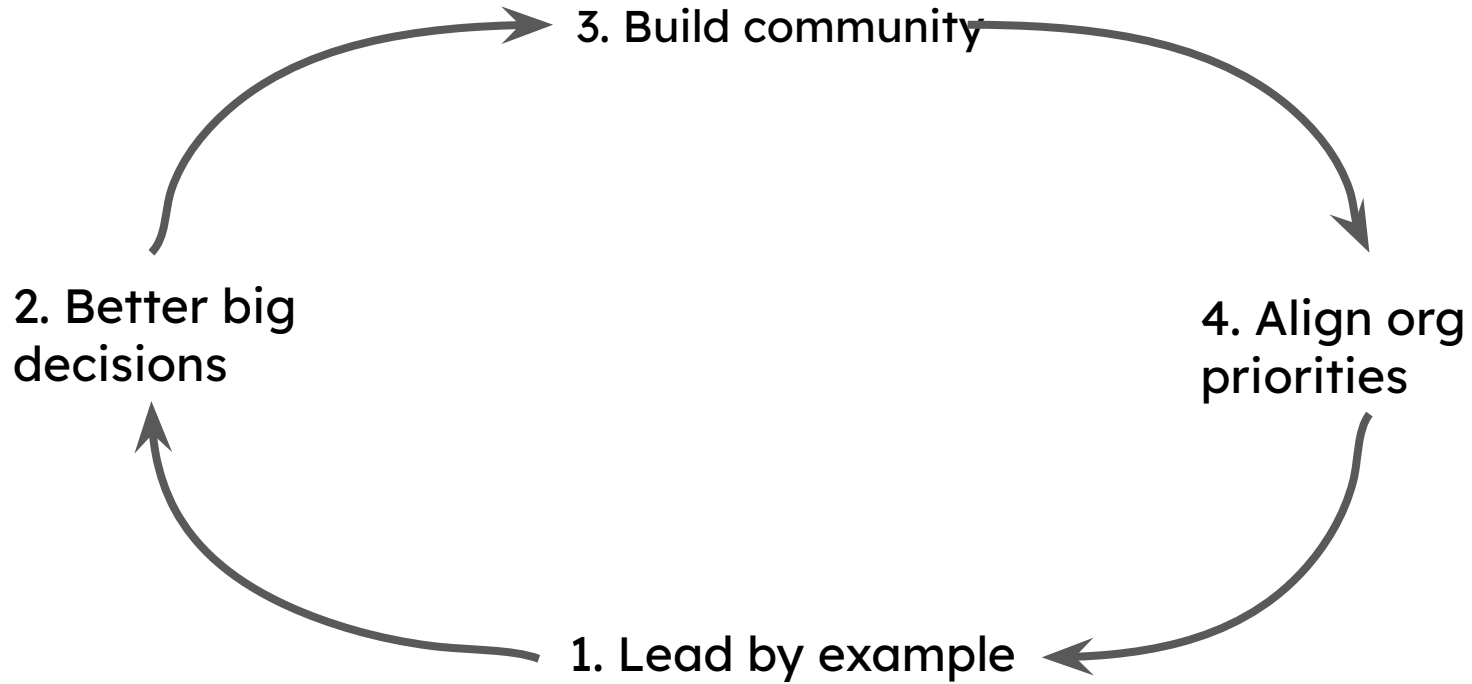
Build great technology

Business

Technology

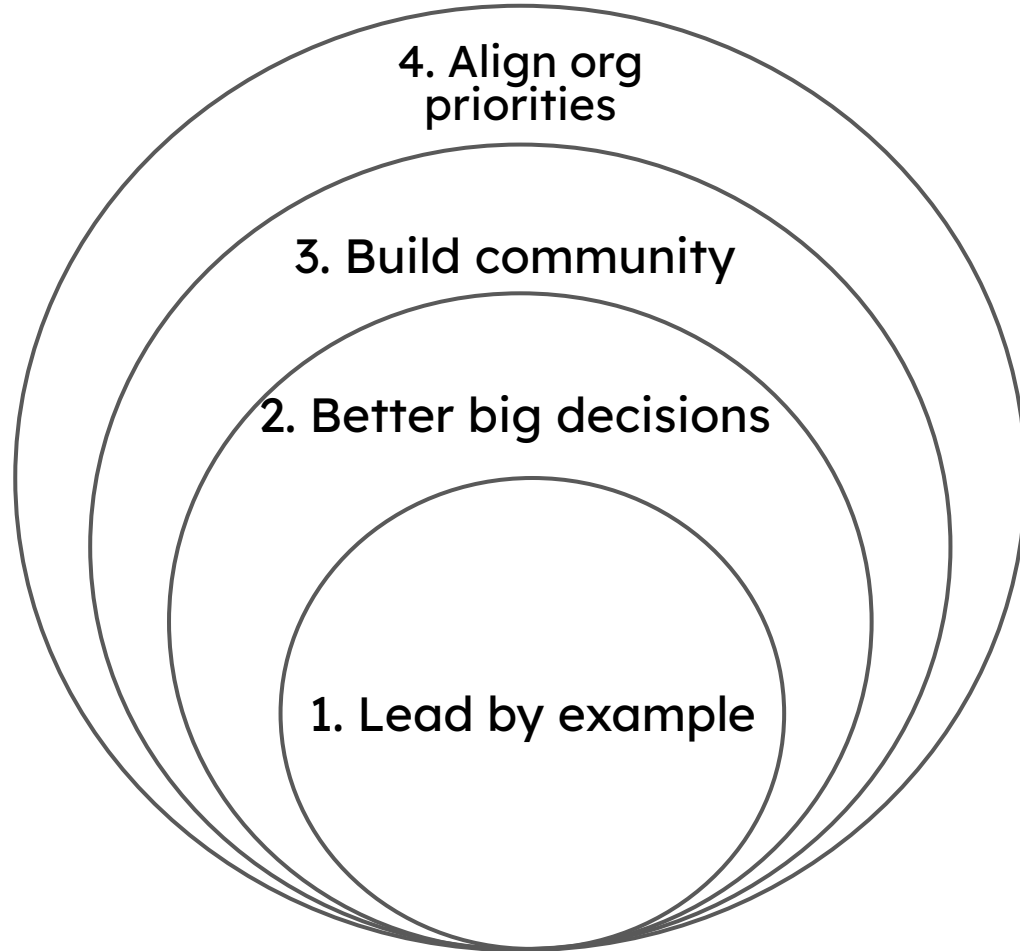
IT DEPENDS (™) pt 1: Business

IT DEPENDS (™) pt. 2: Technology



**GREAT
TECHNOLOGY**

**TO SUPPORT THE
EVOLUTION OF THE
BUSINESS**





natural
instinct



natural
instinct



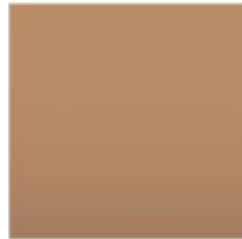
wisdom

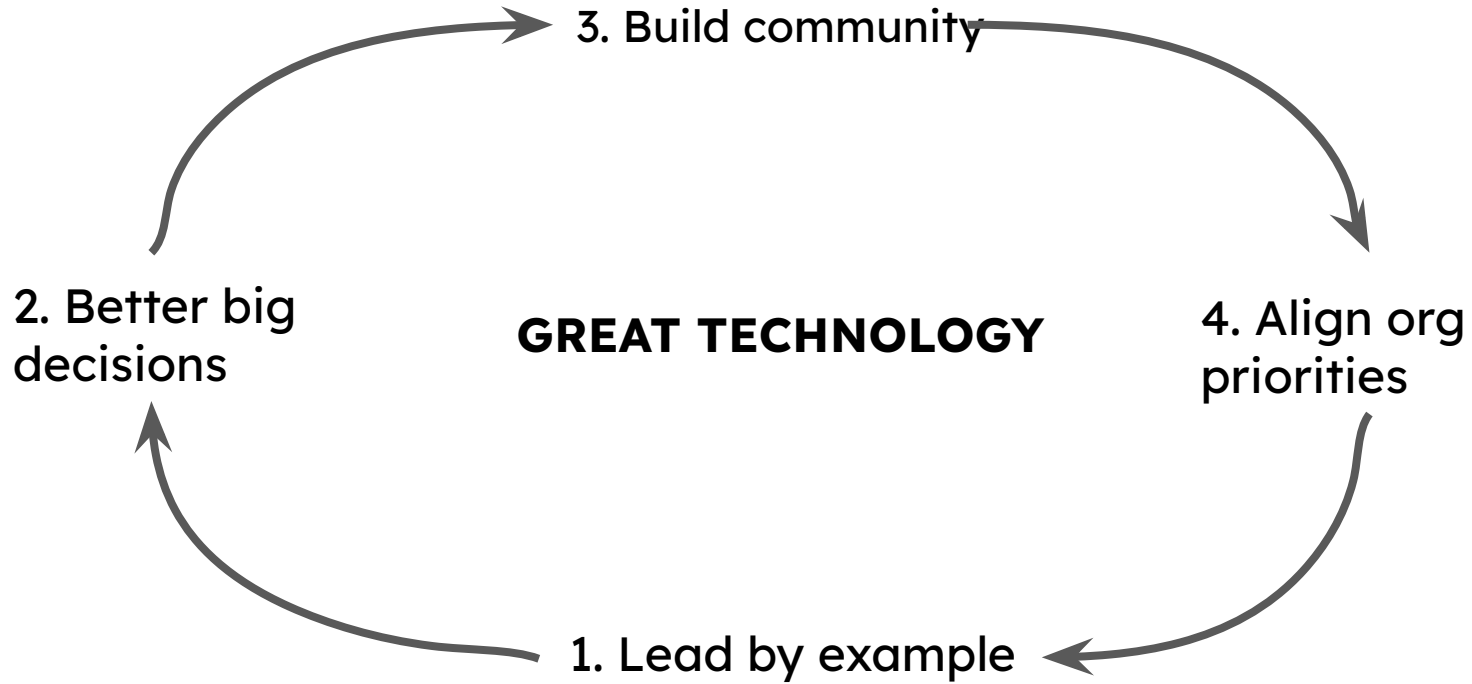


natural
instinct



wisdom





EVOLUTION OF THE BUSINESS

SOUND THE ALARM

TEMPLATE SLIDES COME AFTER

PROCESS > PROBLEM

SCALING CLUELESSNESS

Now let's talk about two traps we ran into as well

**1. LEAD BY
EXAMPLE**

**2. BETTER BIG
DECISIONS**

**3. BUILD
COMMUNITY**

**4. ALIGN ORG
PRIORITIES**



Scenario 1:
We don't know how
to make a decision



Diagnosis:
Unclear when to
commit to a decision



Our Solution:
RFC Process

TEMPLATE CAPTION

**1. LEAD BY
EXAMPLE**

**2. BETTER BIG
DECISIONS**

**3. BUILD
COMMUNITY**

**4. ALIGN ORG
PRIORITIES**

TEMPLATE CAPTION

TEMPLATE

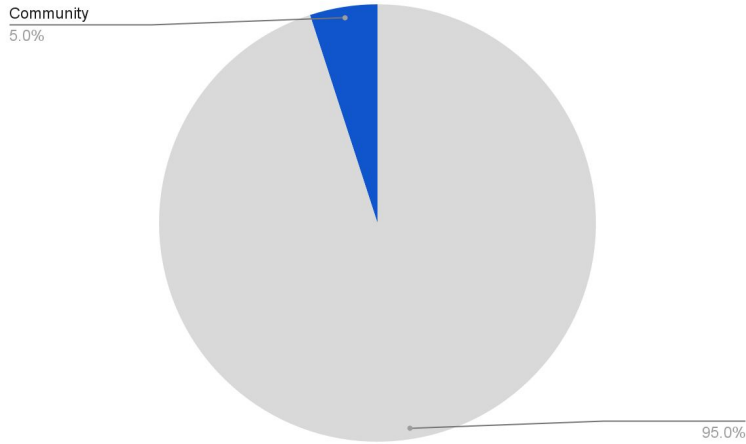
TEMPLATE

TEMPLATE

ITERATION 0

ITERATION 1

ITERATION 2



✓ Share context to surface problems

✓ Opinionated technical principles

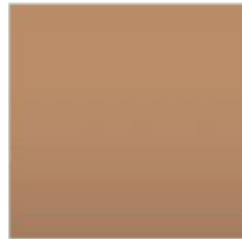
✗ Building solutions

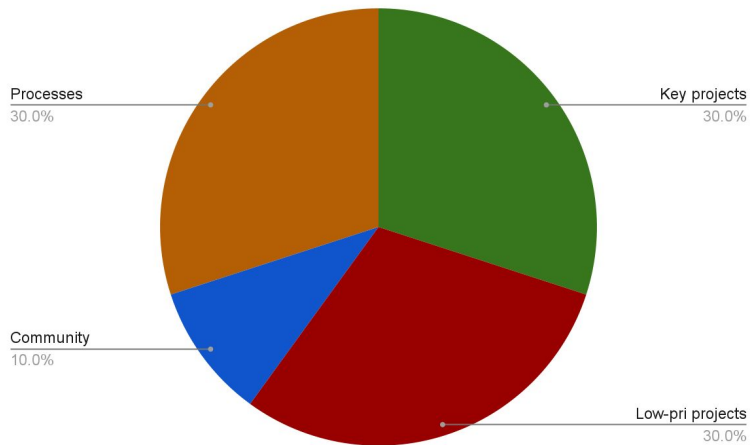
Side-project committee

natural
instinct



wisdom





✓ Identified & shipped a few large projects

✗ Spent a lot of time gap-filling

! Created ineffective (aka context-free) process

Full-time job

TEMPLATE CAPTION

1. Lead by example

**2. Better big
decisions**



1. Lead by example

