

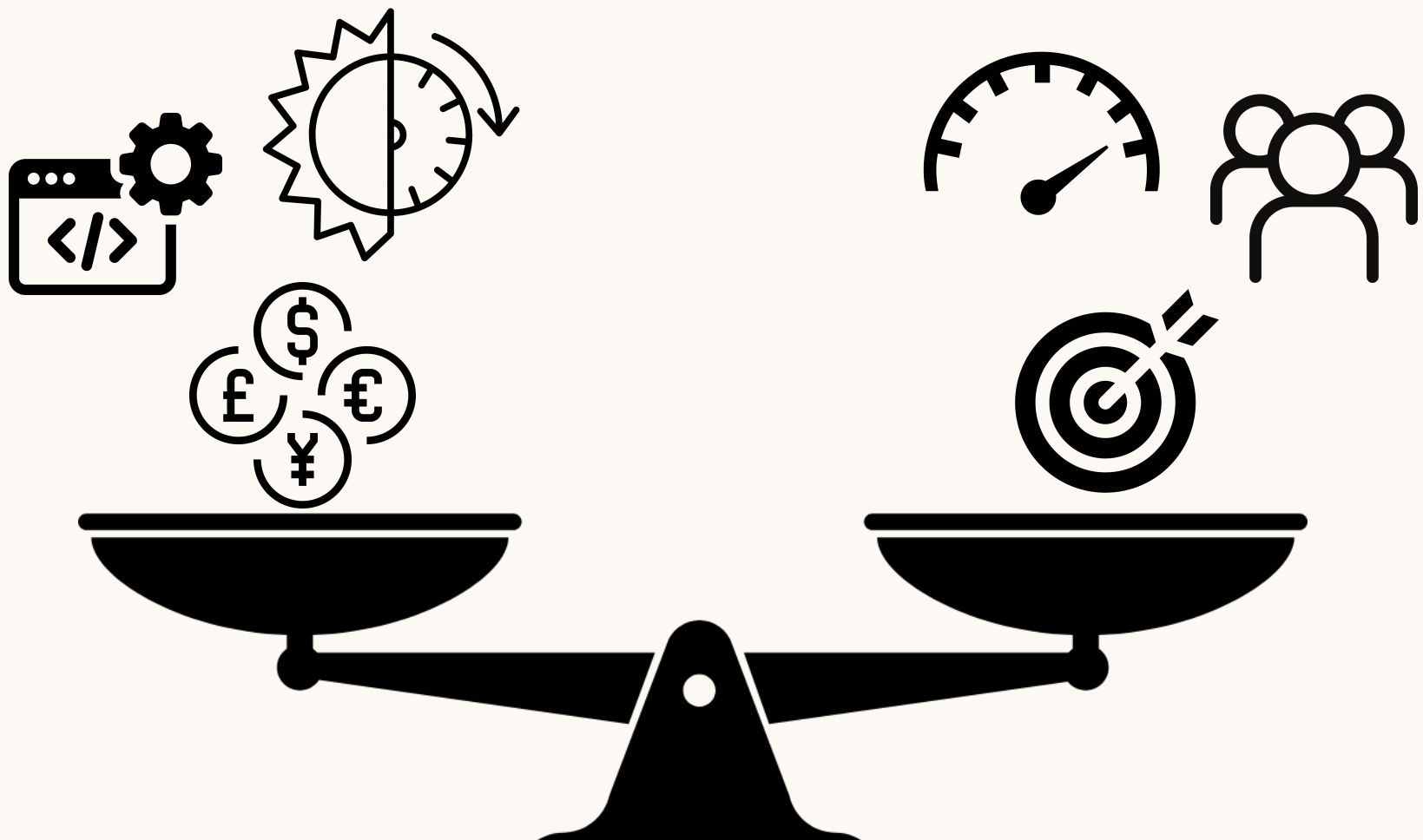
carta

# Navigators: Connecting Execs with Staff Plus Engineers to Shape Strategy

Dan Fike & Shawna Martell

StaffPlus NYC 2024

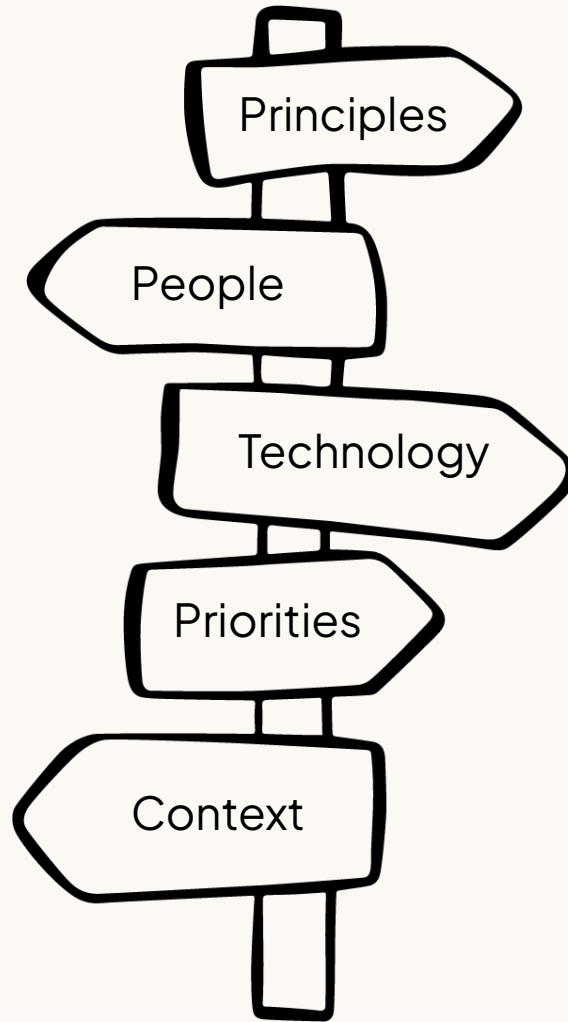






IT DEPENDS

IT DEPENDS



*“We don’t have an Engineering **strategy**.”*



Shawna Martell,  
Sr. Staff Eng



Dan Fike,  
Deputy **to the** CTO

# Navigators



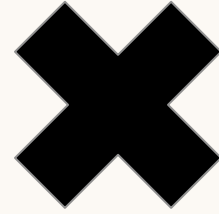
# Make Better Decisions More Quickly

- What failed?
- What else?
- What works?
- What's new?
- What now?





- Clear decision makers
- Regular availability

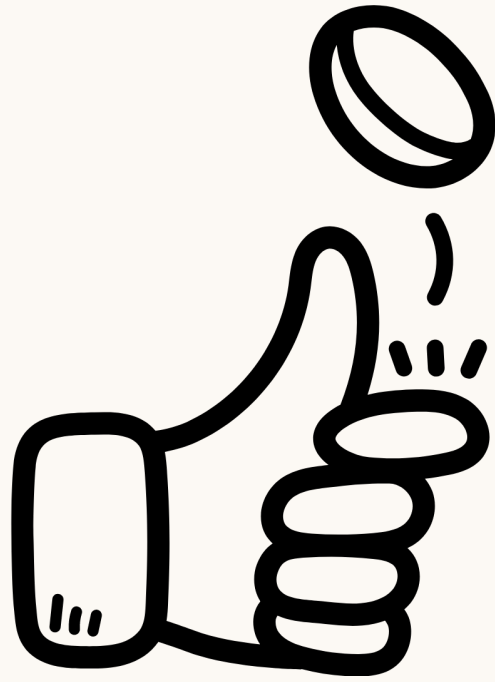


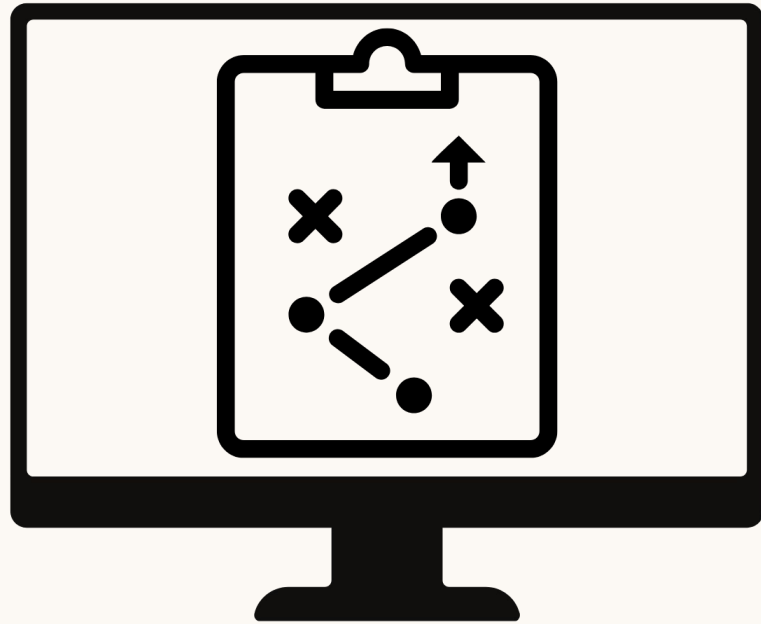
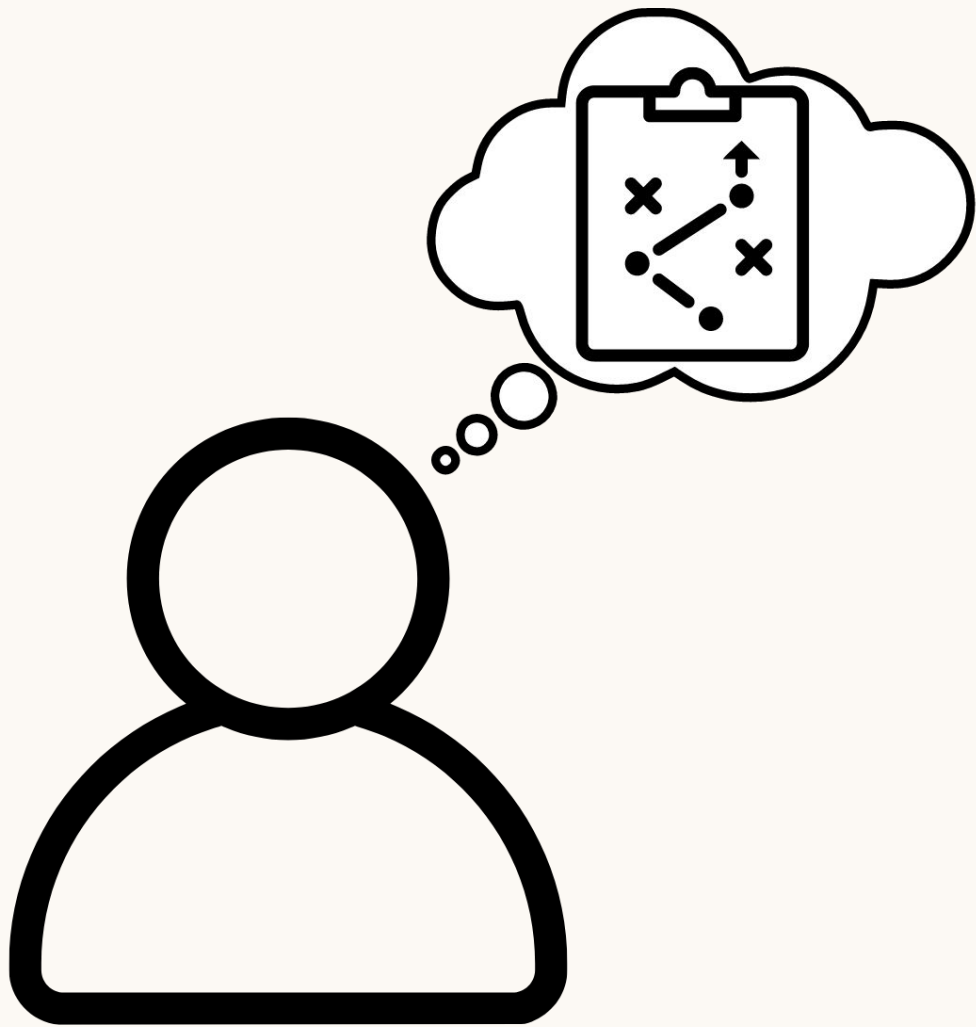
- Status symbol
- Long lead times

“We don’t have an Engineering **strategy**.”

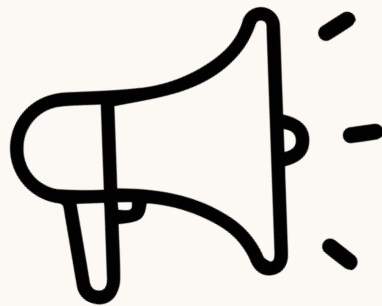
**The best platform solutions enable simplicity**, not complexity; and they are built simply.

**Prefer “buy” over “build”**





**carta**







Irrational Exuberance

Popular

## Will Larson

Want help with something? [Here are ways I'm available to help!](#)

Looking for my investments? [Here they are!](#)

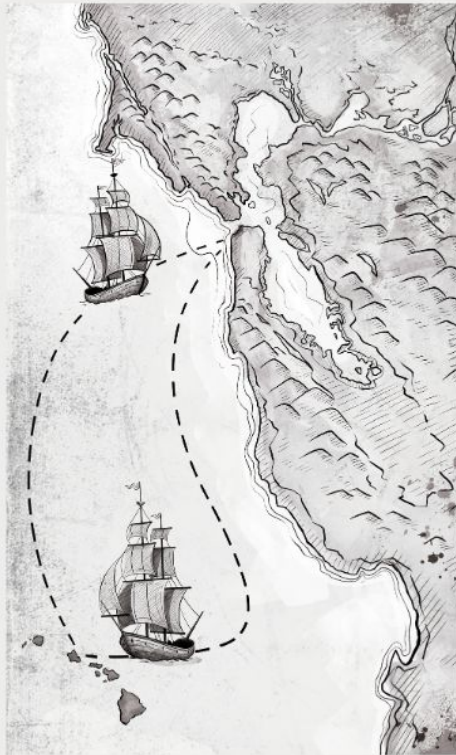
Bio and headshots for an event? [Check at the bottom.](#)

Hi folks,

I'm a writer and a software engineering leader. I currently have the good fortune to serve as the CTO for [Calm](#)'s engineering team. I've previously worked at [Stripe](#), [Uber](#), [Digg](#), [a few other places](#), and much earlier I [cofounded a misguided iOS gaming startup](#) with [Luke Hatcher](#), [Monocle Studios](#). (We made *tens of thousands* of dollars in a year, and spent the next six years trying to figure out how to stop paying taxes.)

In 2019, I published my first book, [An Elegant Puzzle](#), and in 2021 I published my second book, [Staff Engineer](#). I track sales numbers in my annual review posts, [the latest of which is for 2021](#). I'm not working on any book at the moment, although imagine I will write another sooner than later!

I've been writing on this blog, [Irrational Exuberance](#), since 2007. In 2018, I started doing [some speaking as well](#) most notably about [infrastructure migrations at QCon SF](#), and the [infamous Digg v4 launch](#) on the Internet History Podcast.



## Staff Engineer

Leadership beyond the management track

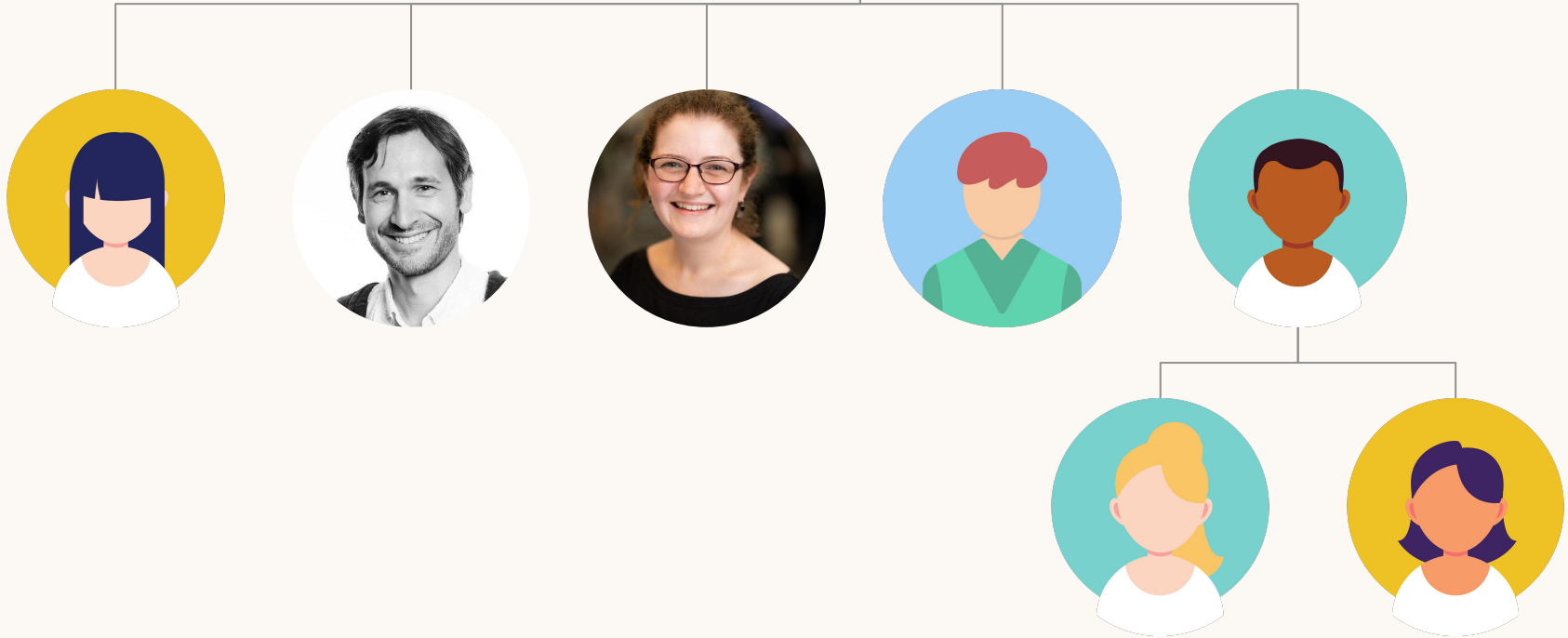
Will Larson

### ...y, consensus, and conviction: Three p styles for effective engineering

with one finely honed leadership style, and routinely stepped  
style for every situation. In this talk, Will shares the situations  
oolkit of leadership styles, the new styles he has added, and  
sing them.



**SEVERAL  
MONTHS  
LATER...**

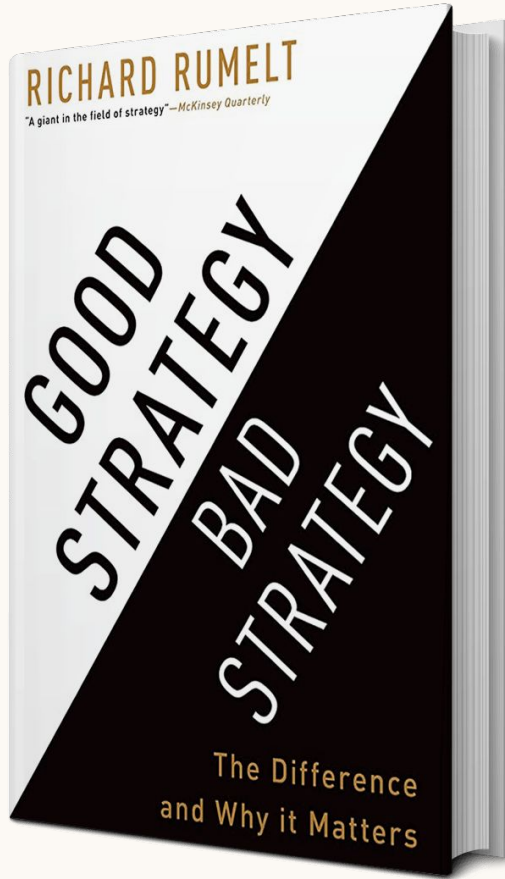


RICHARD RUMELT

"A giant in the field of strategy"—*McKinsey Quarterly*

GOOD  
STRATEGY  
BAD  
STRATEGY

The Difference  
and Why it Matters



# Diagnosis

The current situation

**Sometimes platform tech is viewed as introducing problems as much as solving them.** Over-generalized solutions are difficult to adopt and reason about for a specific use case, increasing the load on product teams.

---

# Guidance

Principles behind decisions

**Platforms should initially prioritize solving *one problem*** before attempting to solve *all problems*. Focus on addressing the needs and requirements of a specific customer, rather than attempting to create a one-size-fits-all solution.

---

# Actions

Steps we're taking

**No new platform projects without approval from the CTO.**

# Diagnosis

The current situation

**We continue to have unstaffed systems and features.** While some of these have well-defined owners who are accountable for them, that ownership is often not backed by sufficient staffing or familiarity with the system itself.

---

# Guidance

Principles behind decisions

**Finish it or delete it.** Leaving work unfinished often means we made most of the investment and reaped none of the impact. Only code that doesn't exist can't break.

---

# Actions

Steps we're taking





Pre-Strategy



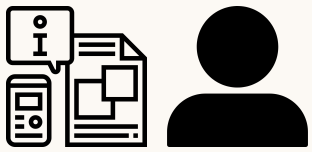
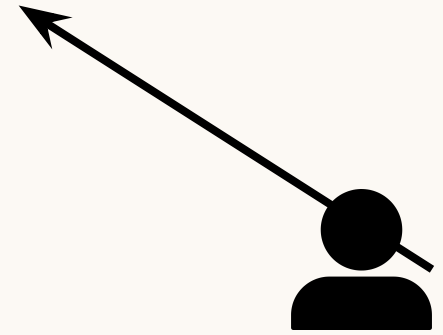
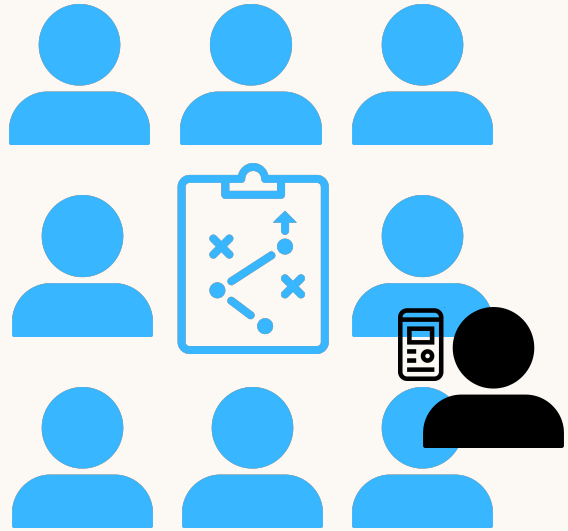
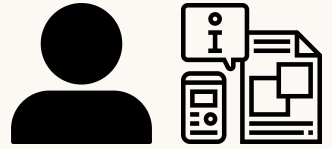
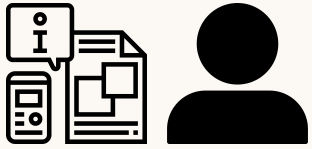
v1

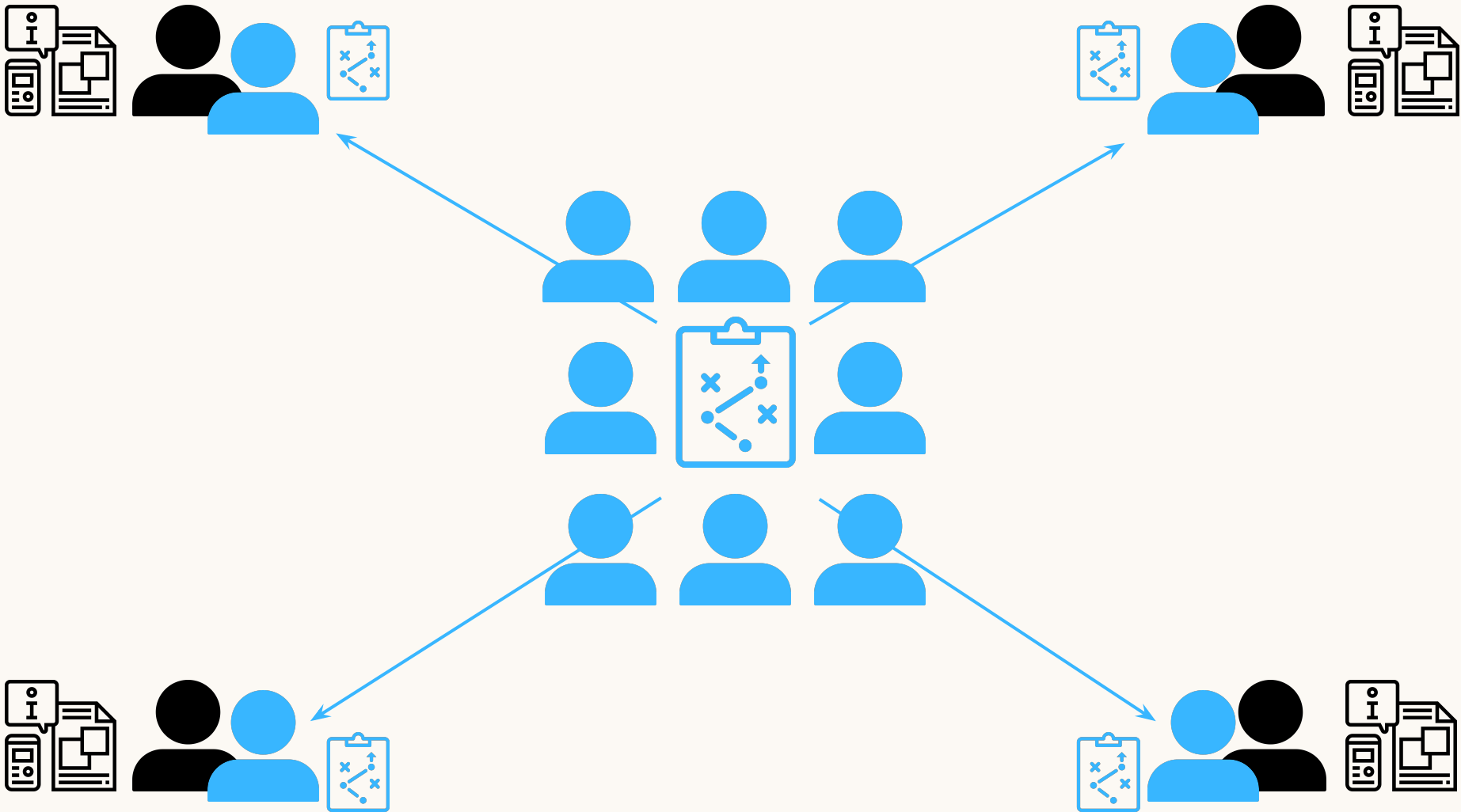


v2



# Navigators





# Navigators @ Carta

As we begin to explicitly document and iterate our Engineering Strategy, we need additional levers to help our engineering teams understand how their work intersects with that strategy. It can be difficult to navigate the delta between where a team is today and where the strategy is driving us. This is why we're going to explicitly identify a small set of Navigators across Carta who help teams in their area interpret and apply the strategy. Navigators are responsible for building/providing alignment between their teams and the broader engineering strategy.

Each navigator is an **existing technical leader** that will seek to **replace the need for wide consensus** and **boost velocity** by bringing their **technical context**, **domain context**, **strategic alignment**, and **judgment** together to **make engineering decisions more quickly**.

- Accountable to both the CTO (for acting in accordance with the Engineering Strategy) and their existing management (for representing the interests of their teams and making sound technical decisions). If management and the strategy are misaligned, escalate to the CTO
- Expected to find alignment with the CTO's strategy; this shouldn't just be "disagree and commit"
- Informing the CTO elements of the Engineering Strategy that are not working or are missing

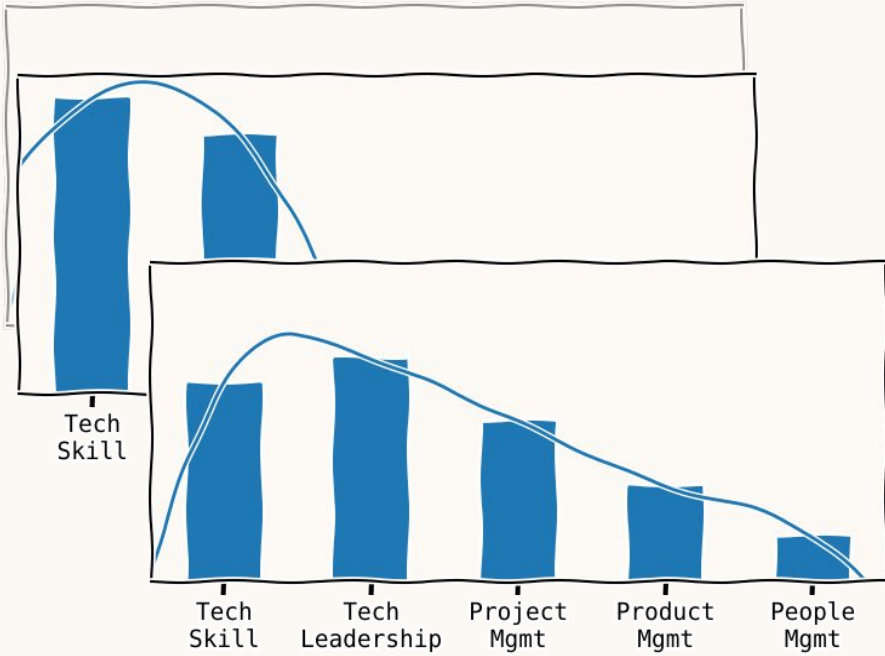
Navigators are NOT

- Micromanaging every design or implementation decision at Carta
- Architects responsible for writing all your TDDs
- Blockers; they are unblockers
- Responsible for execution speed/quality of their teams

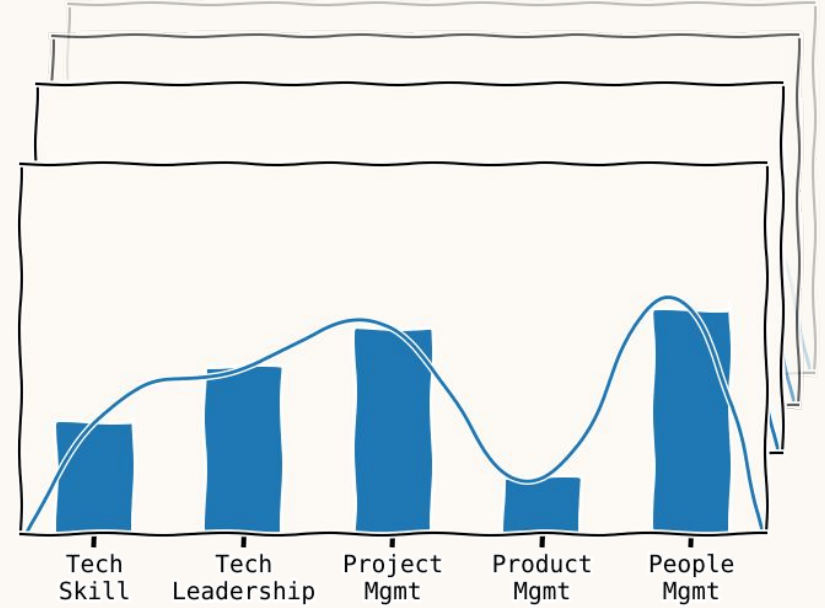
In many cases, the engineers we designate Navigators are already doing this work. It is not a new job description or a promotion. The intent behind formalizing this is to

- establish new chains of communication to make them even more effective
- bring that same behavior and judgment to a wider set of teams
- help others understand how to leverage the role better

# Senior ICs



# Eng Manager





## Don't

Micromanage

Write every design spec



## Do

Unblock tech decisions

Maintain team alignment

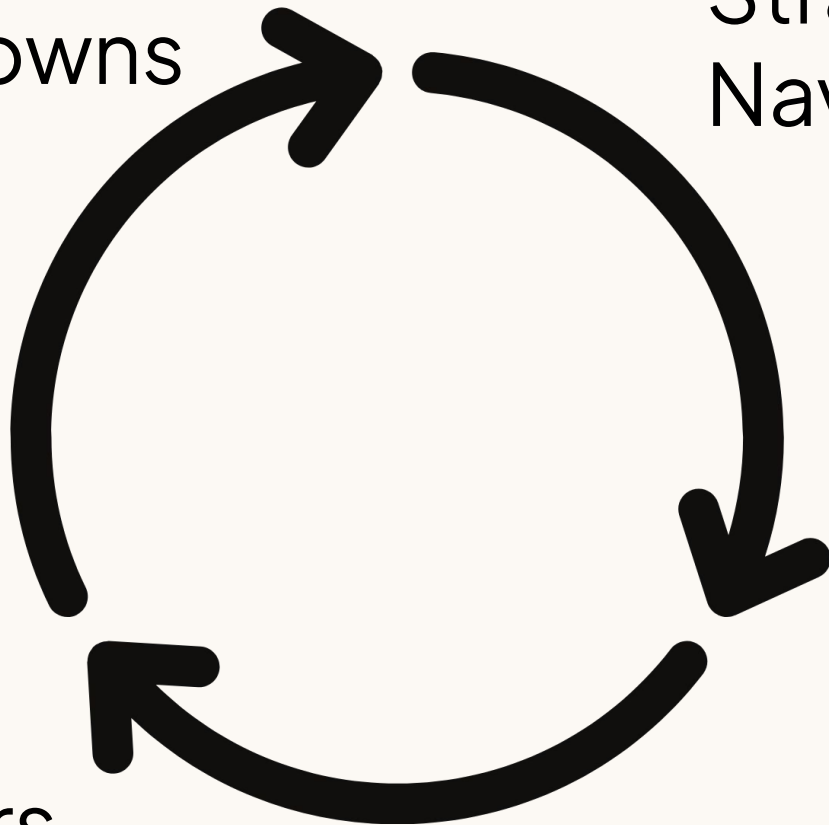
Maintain strategy alignment

Resolve conflict with strategy



Executive owns  
Strategy

Strategy guides  
Navigators



Navigators  
influence strategy





Extend the  
Product

Build a Platform

# Diagnosis

The current situation

Teams often view platform tech as introducing problems as much or more as solving them. Over-generalized solutions are difficult to adopt and reason about for a specific use case, increasing the load on product teams.

**Platforms should initially prioritize solving *one problem* before attempting to solve *all problems*.** Building in phases, each one built alongside their first adopter, powers incremental progress and iteration. **Focus on addressing the needs and requirements of a specific customer,** rather than attempting to create a one-size-fits-all solution.

# Actions

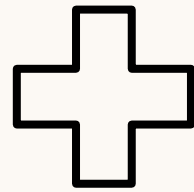
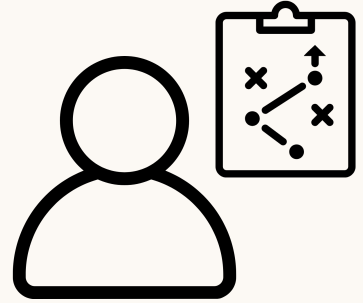
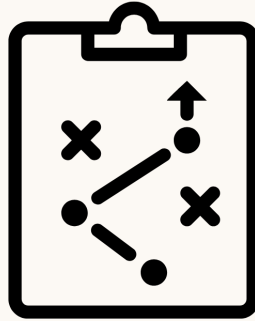
Steps we're taking

**Platforms are the exception, not the rule.** No new platform projects without approval from the CTO. Requirements that bundle capabilities and state but are not a long-lived platform team should be built into the product-specific application.

**WELL THAT ESCALATED**

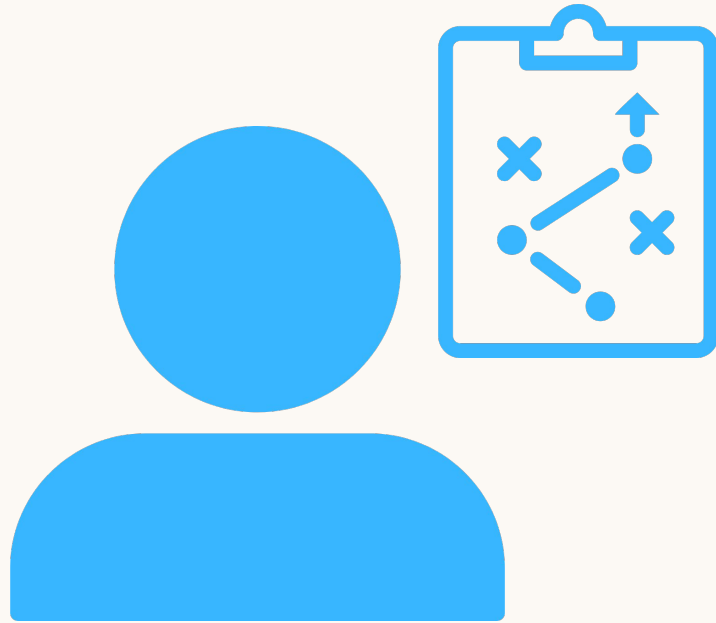


**RIGHT ON TIME**



*“Start wherever you are, and start now.”*





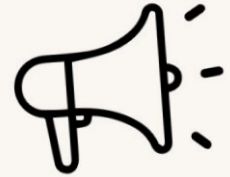
Documented Strategy

Navigators to interpret

Bypass consensus

Iterate

carta



Navigators  
influence strategy

