

Starting from nothing

Lawrence Jones

StaffPlus (Sept 2024)





Hi!

- Lawrence Jones (@lawrjones)
- Previously: Principal Engineer at GoCardless
- Now: Product Engineer at incident.io
- Joined three years ago as first hire
- Now 80 people, 50 in London, 30 in New York





**Big problems.
That's your job.**



What we'll cover

Planning
Execution
Stepping away

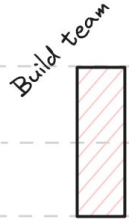


 **incident.io**

On-call as it should be

Planning

July
w2
w3
w4
Aug



Formed team

1 PM, 4 eng
1 designer

2-3 eng

4 eng (ONX)

2 eng
+ rotating pair



Add 2

Bootstrap
new team



Take 1



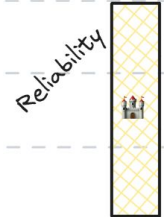
Mobile



Polish



Take from
existing OC teams



Execution

Sept
Oct
Nov
Dec
Jan
Feb
Mar
April



Planning

July
w2
w3
w4
Aug

Build team

Discovery

Formed team

1 PM, 4 eng
1 designer

2-3 eng

4 eng (ONX)

2 eng
+ rotating pair

Scoping

Add 2

Bootstrap
new team

Execution

Sept
Oct
Nov
Dec
Jan
Feb
Mar
April

- Alerts MVP
- Routing
- Escalations
- Schedules
- Onboarding
- Reliability support
- Polish

Take 1

Mobile

Polish

Take from
existing OC teams

Alerts GA

Schedule
improv

Polish

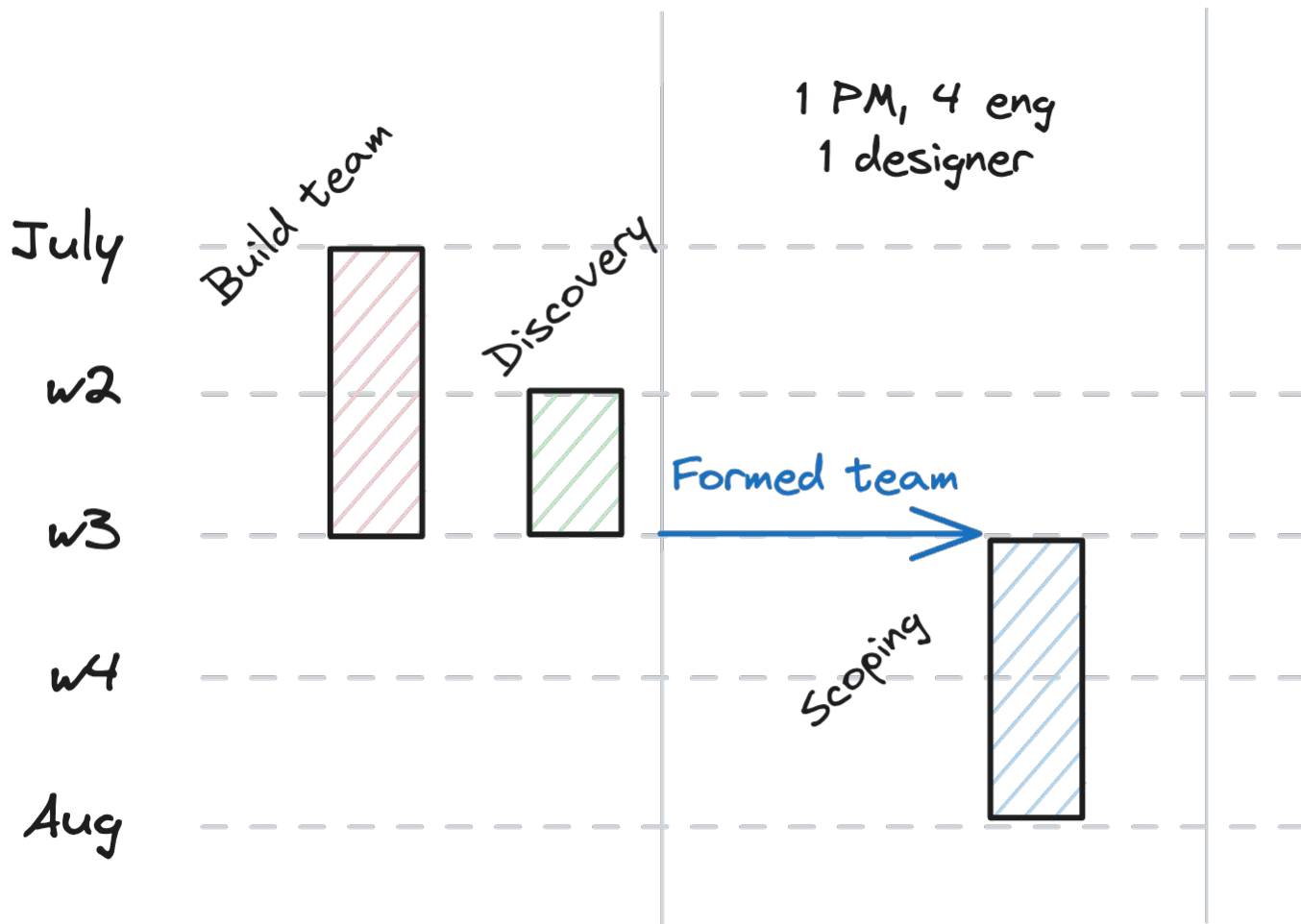
Reliability

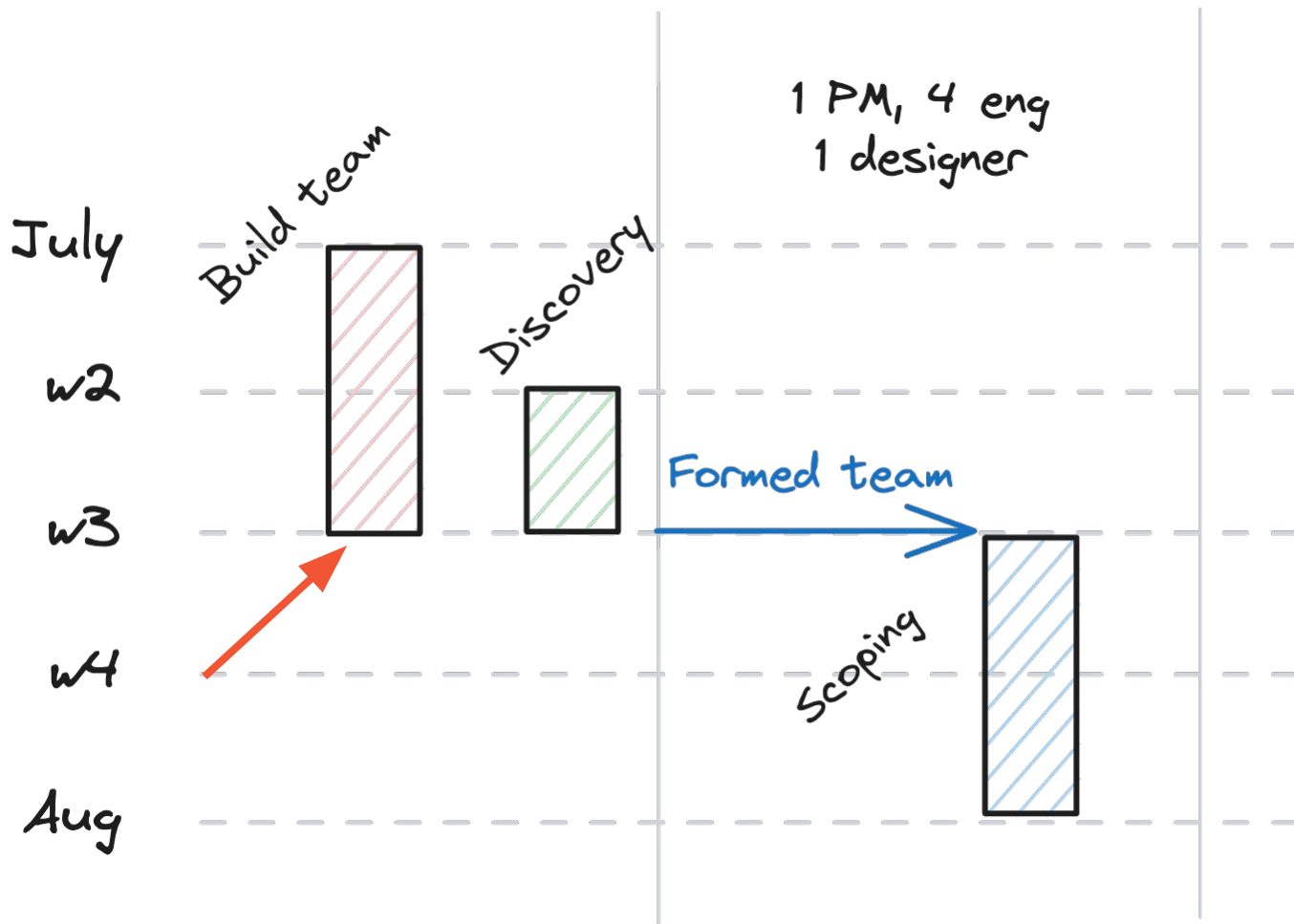
Stepping away



What we'll cover

Planning
Execution
Stepping away







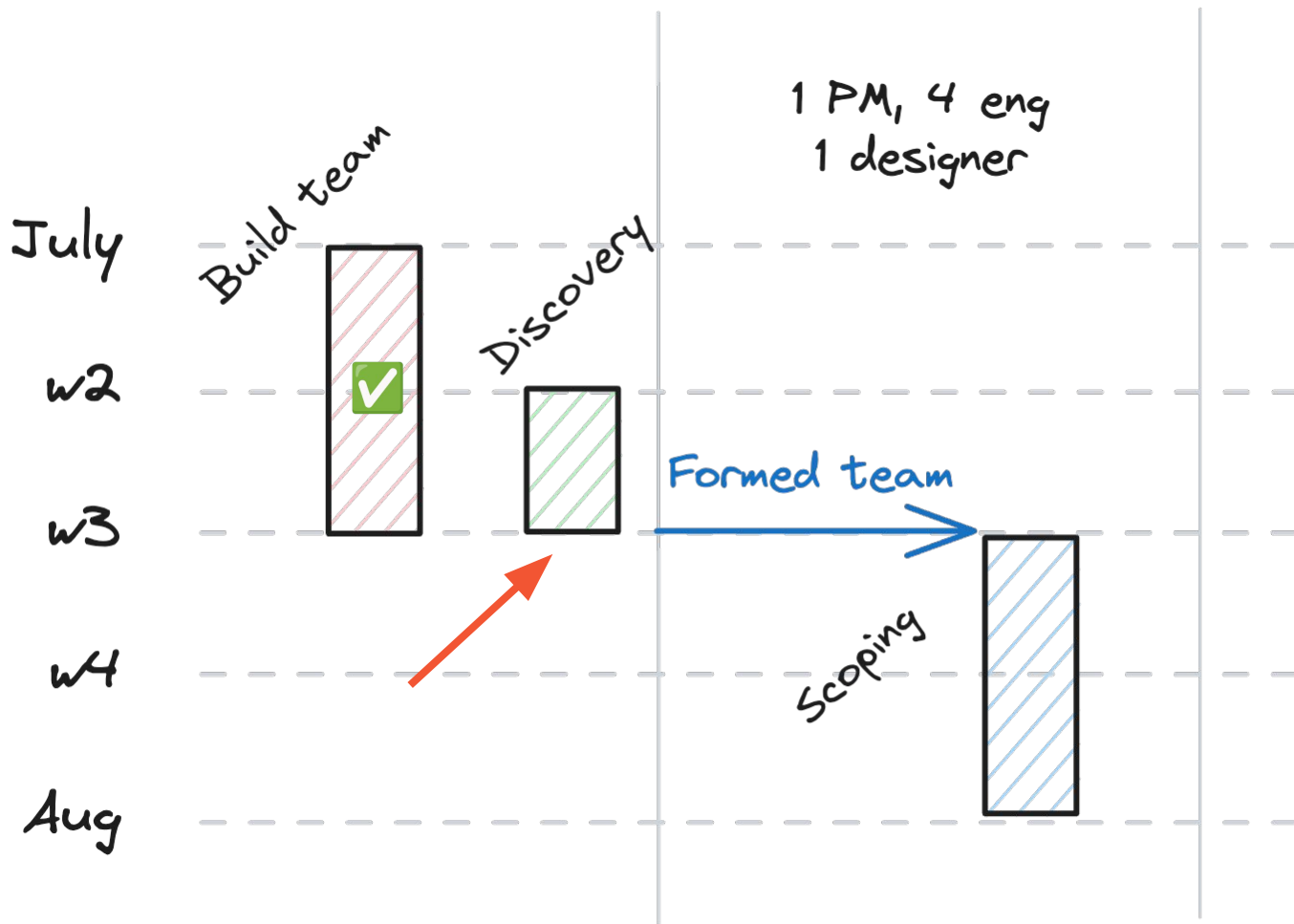
Building the team

- Bring perspective and context to your manager
- Start small, plan ahead
- Think carefully about your Tech Lead



Building the team

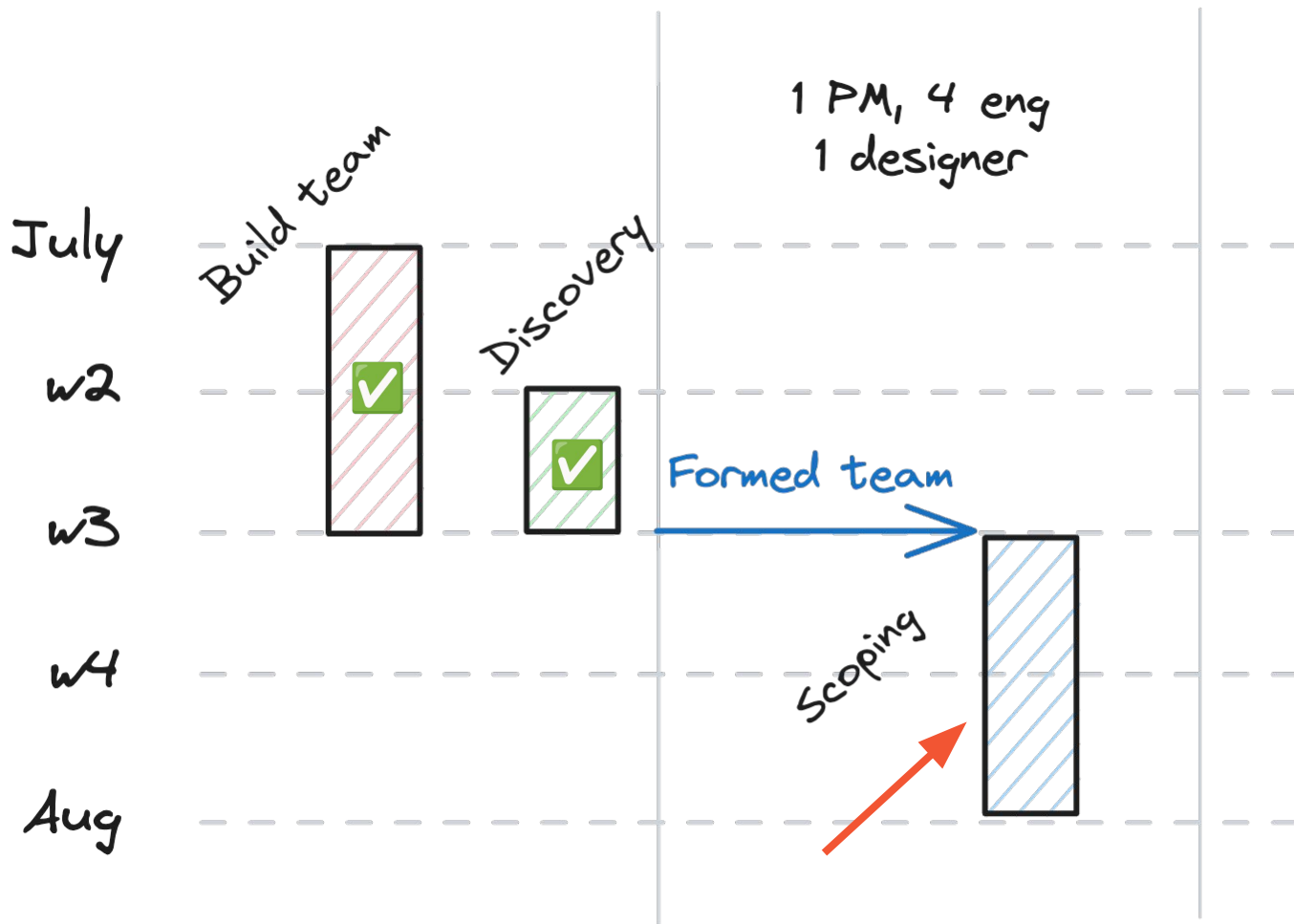
- Bring perspective and context to your manager
- Start small, plan ahead
- Think carefully about your Tech Lead





Pre-team discovery

- You need to have answers
- Identify themes of work
- Form opinions





Scoping

**Intense, exhausting,
fun.**

"Collaboration station"





How do you help scoping?

1. Break down the problem: ultimately, this is on you
2. Maintain the 'right' level of ambiguity
3. Provide context, catch problems, join the dots



What we'll cover

Planning
Execution
Stepping away



Planning

July
w2
w3
w4
Aug

Build team

Discovery

Formed team

1 PM, 4 eng
1 designer

2-3 eng

4 eng (ONX)

2 eng
+ rotating pair

Scoping

Add 2

Bootstrap
new team

Stepping away

Execution

Sept
Oct
Nov
Dec
Jan
Feb
Mar
April

Alerts MVP

Routing

Escalations

Schedules

Onboarding

Reliability support

Polish

Take 1

Mobile

Polish

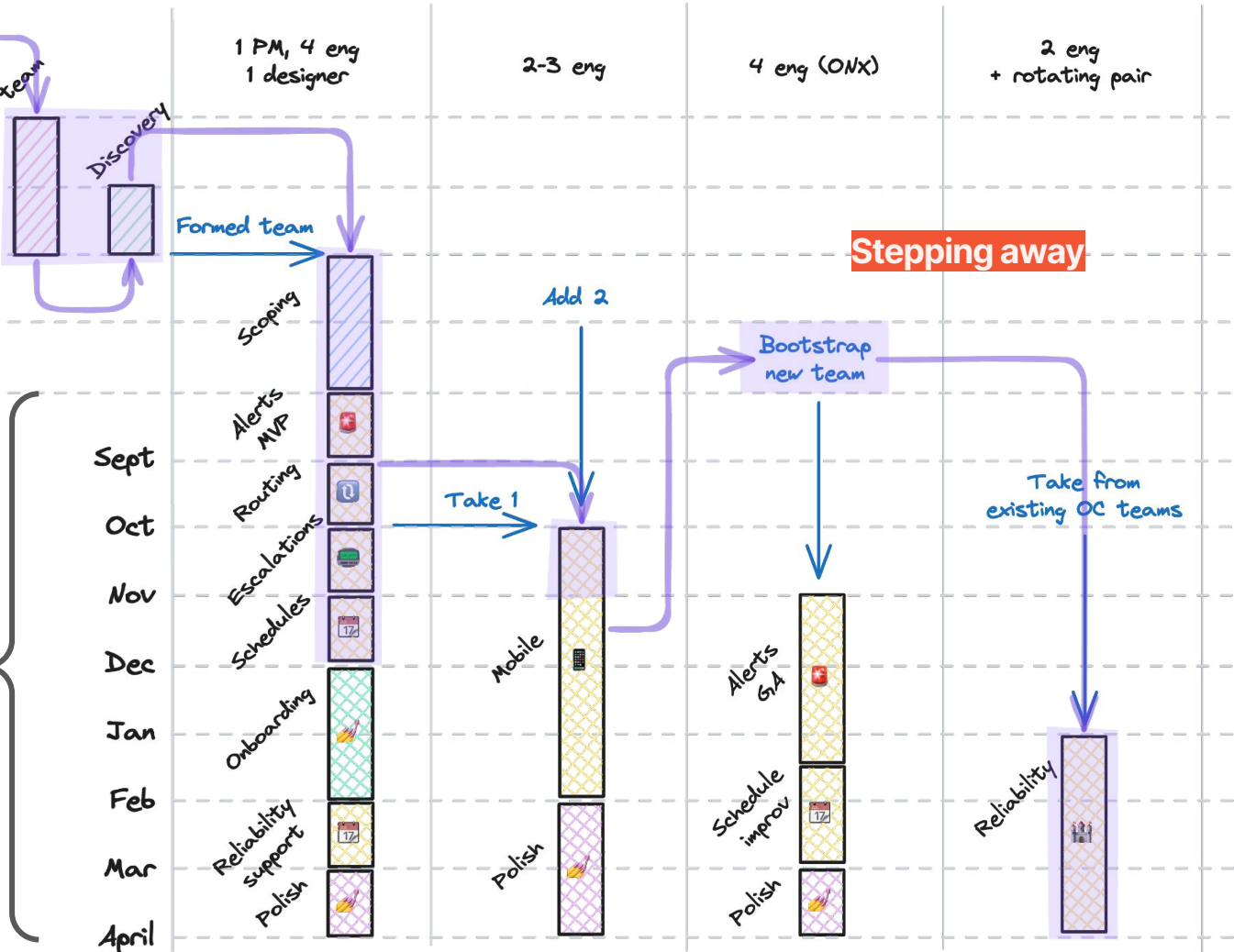
Alerts GA

Schedule improv

Polish

Take from
existing OC teams

Reliability





Working with the team

- Don't shy away from building
- Three modes: all-in, supporting, on other things
- Your team should always know what mode you're in



What we'll cover

Planning Execution Stepping away



Planning

July
w2
w3
w4
Aug

Build team

Discovery

Formed team

1 PM, 4 eng
1 designer

2-3 eng

4 eng (ONX)

2 eng
+ rotating pair

Scoping

Add 2

Bootstrap
new team

Stepping away

Execution

Sept
Oct
Nov
Dec
Jan
Feb
Mar
April

Alerts MVP

Routing

Escalations

Schedules

Onboarding

Reliability support

Polish

Take 1

Mobile

Polish

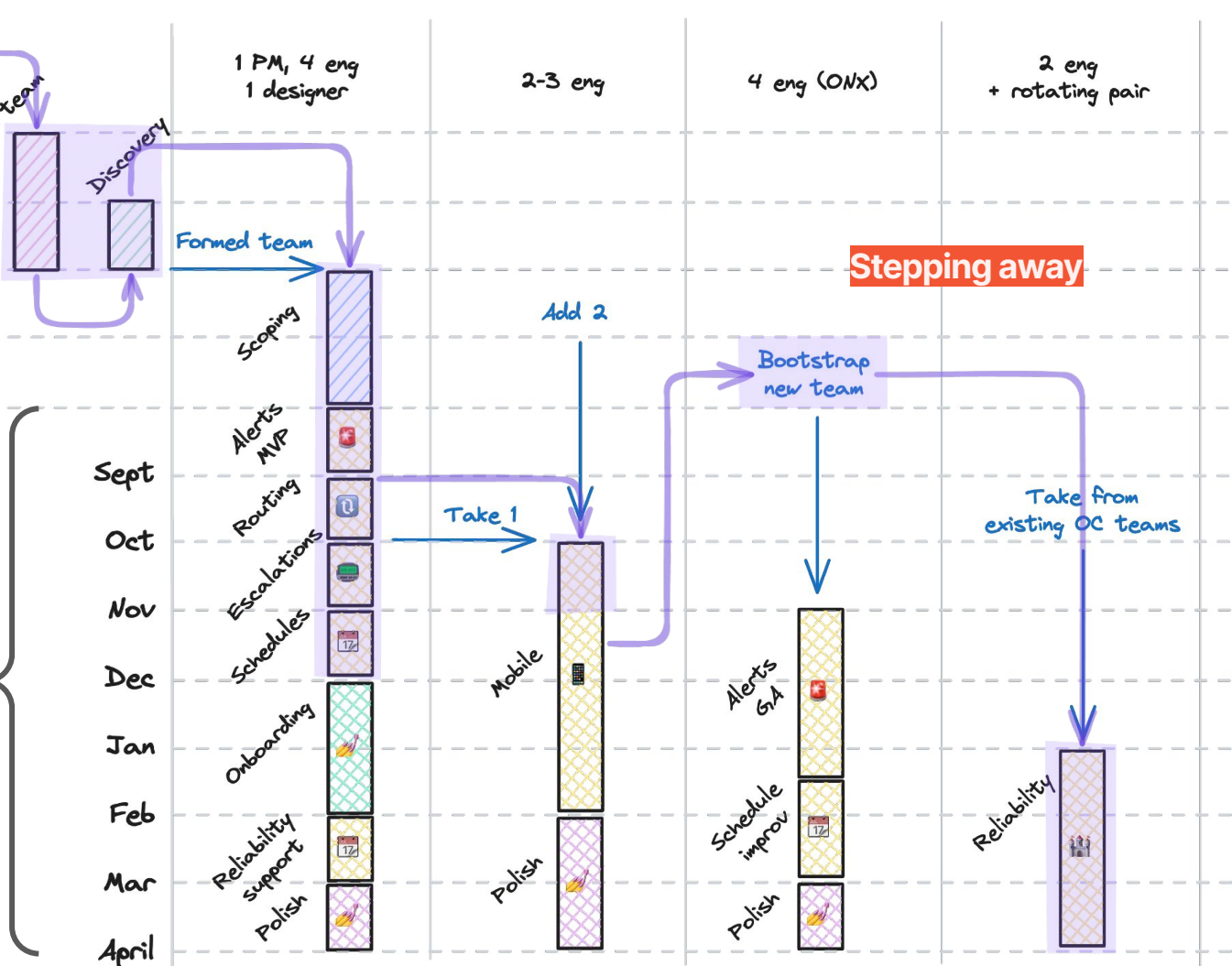
Alerts GA

Schedule improv

Polish

Take from
existing OC teams

Reliability





Stepping away

- Be flexible and ready to move
- Dial back your IC work
- Keep syncing with team leads
- Over-communicate your priorities



Maintaining alignment

- Up, laterally, down
- It may feel pointless, until it isn't
- Don't let it slip!



**It's not easy, but
it is fun.**



Remember...

- This work is rewarding
- It's always a people problem
- Win together

Catch me at office hours for questions!

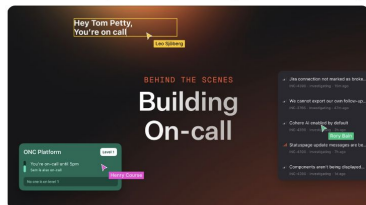
Building On-call

<https://incident.io/hubs/building-on-call>



Behind-the-scenes building On-call

In this series we go behind-the-scenes of how we built our on-call product. From close collaboration with design partners, to rigorous load testing and reliability drills, this is how we built a product that reimagines the on-call experience. (80 kB) ▾

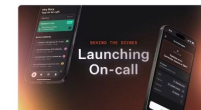


Read more here!

Behind-the-scenes building On-call

Creating an on-call product is hard: it has to be rock-solid, capable of handling major storms, and be designed to minimize the impact of on-call on the lives of those responding.

In this series, we share behind-the-scenes details of how we built our on-call product. From collaborating closely with our design partners to running rigorous load testing and reliability drills, we'll share the journey of developing a product that reimagines the on-call experience.



Engineering

Building On-call: Launching On-call

We like to ship: if then shout about it, all the time. Building On-call was different.

Henry Course
July 9, 2024 · 8 min read

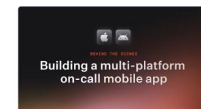


Engineering

Building On-call: The complexity of phone networks

Making a phone call is easy...right? It's time to re-examine the things you thought were true about phone calls and SMS.

Leo Sjöberg
July 16, 2024 · 7 min read

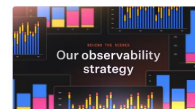


Engineering

Building On-call: Building a multi-platform on-call mobile app

What does it take to build a greenfield mobile app in 2024? When we launched On-call earlier this year, we had to find out.

Rory Bain
July 15, 2024 · 17 min read

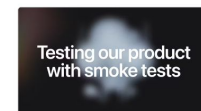


Engineering

Building On-call: Our observability strategy

Our customers count on us to sound the alarm when their systems go sideways—so keeping our on-call service up and running isn't just important; it's non-negotiable. To nail the reliability our customers need, we lean on some serious observability (or as the cool kids say, o1y) to keep things running smoothly.

Martha Lambert
August 22, 2024 · 21 min read



Engineering

Building On-call: Continually testing with smoke tests

Launching On-call meant we had to make our system rock-solid from the get-go. Our solution? Smoke tests to let us continually test product health and make sure we're comfortable making changes at pace.

Rory Malcolm
August 9, 2024 · 11 min read