
Chevrolet
CHEVROLET
MAKES SENSE FOR

1978

The last good year




1978
1980
1984
Detroit, Michigan.

1978

1980

1984







1978

1980

1984

Record sales.

GM

$63bn

Ford

CHRYSLER

1969

1973

1978

1978

1980

1984

Record profits.



GM

$63.2bn
sales

$3.5bn
profit

1969

1973

1978

1980

1978

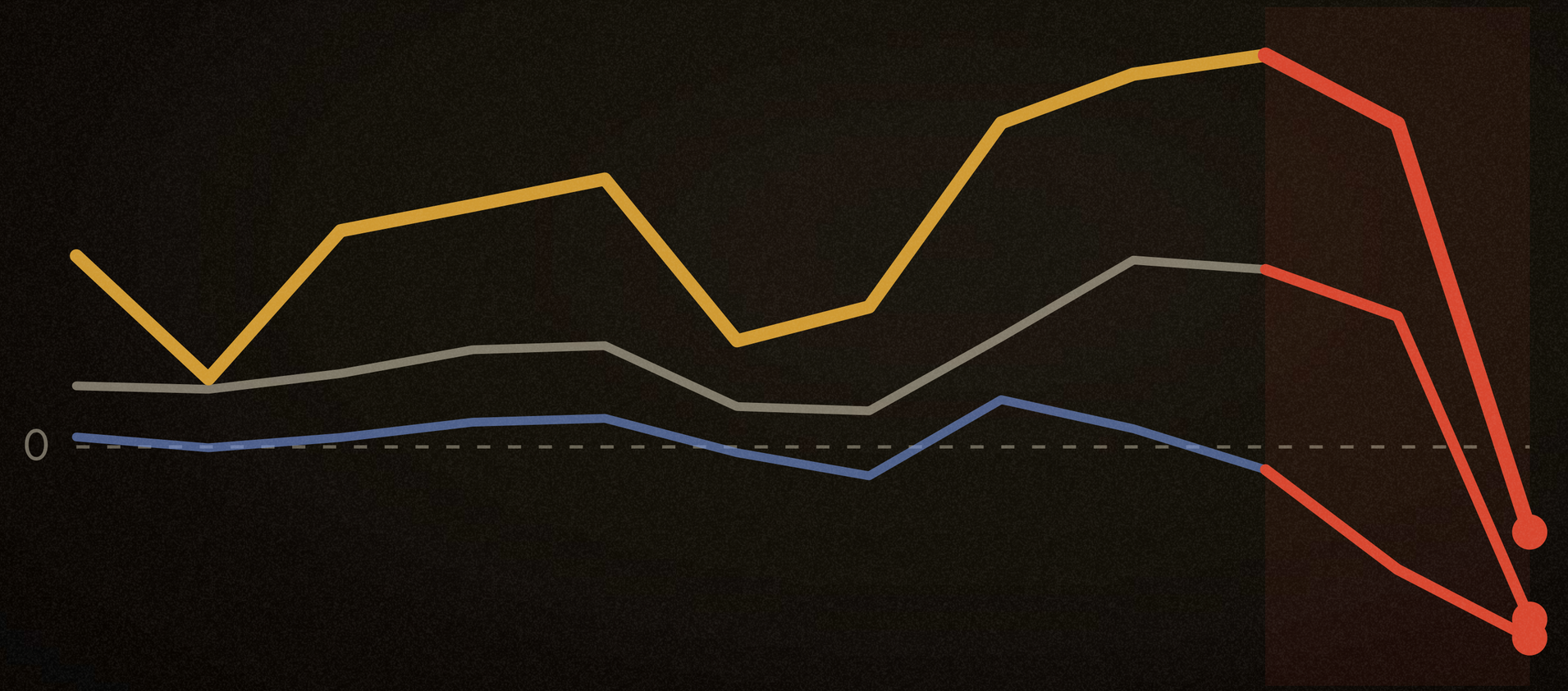
1984

Two years later...

1980

1978

1984



GM –$0.8bn

Ford –$1.5bn

CHRYSLER –$1.7bn

$4 billion

lost by the Big Three, 1980

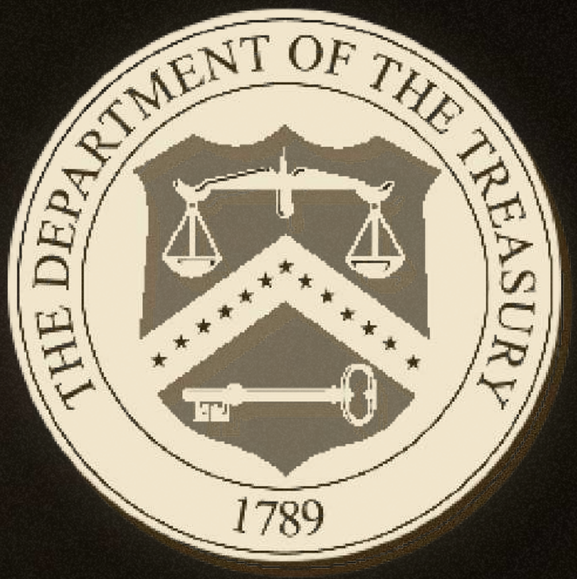
1980

1978

1984

CHRYSLER

—



Chrysler asked for a bailout.


BELNOR


TOYOTA

Smaller.

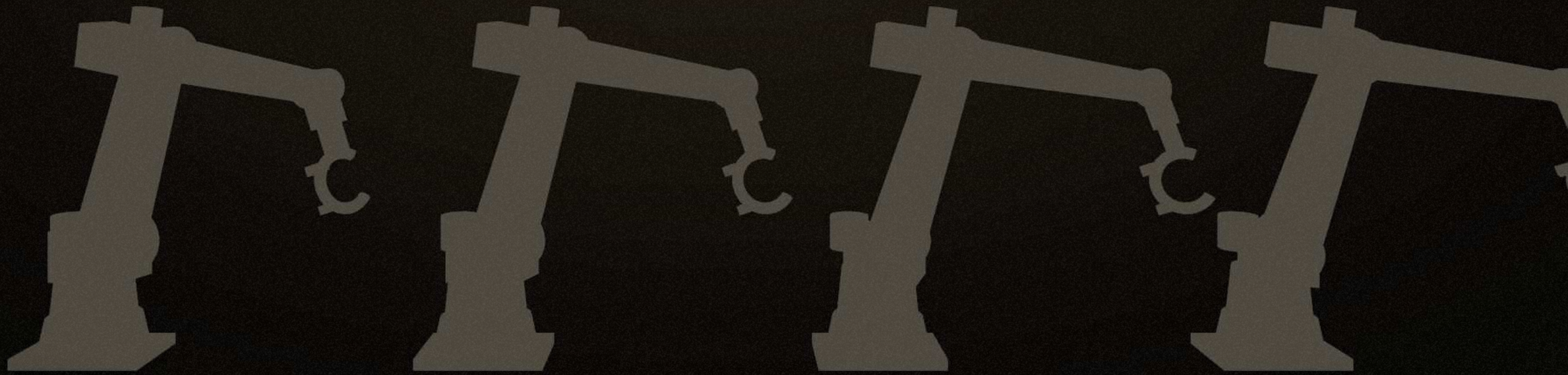
Smaller.

Cheaper.

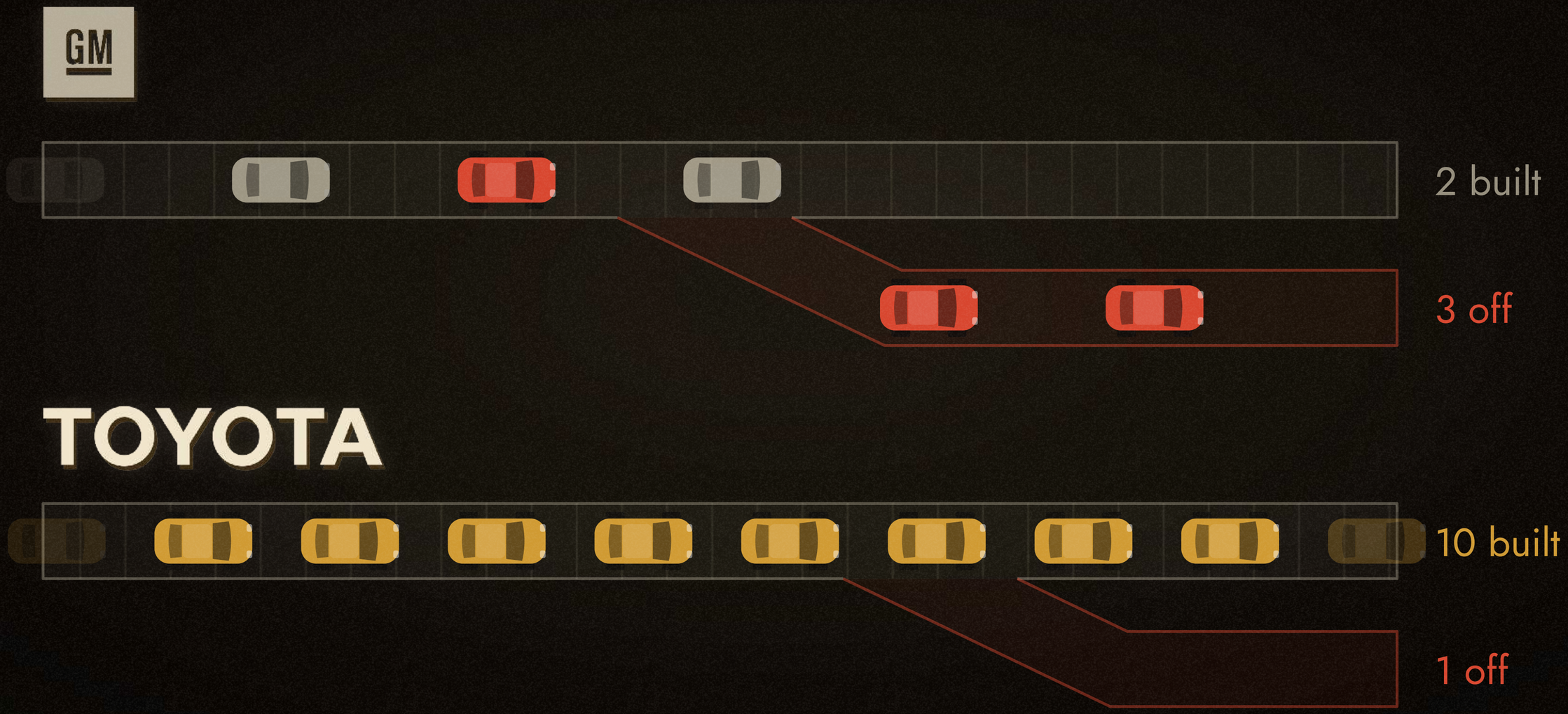
Smaller.
Cheaper.
Better built.

Tens of billions on automation.

GM alone: over $45bn on acquisitions and automation, 1980–85



It didn’t.


GM
2 built
3 off
TOYOTA
10 built
1 off

GM

Framingham, 1987

31 h per car

TOYOTA

Takaoka, 1987

16 h per car

MIT, 1987 plant data (Womack, Jones & Roos, 1990)

GM

Framingham, 1987

31 h per car

135

defects per 100 cars

TOYOTA

Takaoka, 1987

16 h per car

45

defects per 100 cars

MIT, 1987 plant data (Womack, Jones & Roos, 1990)

Build the right thing.

Build the right thing.

Build it right.

Build the right thing.
Build it right.
Speed follows.


2026

Shopify

“Reflexive AI usage is now a baseline expectation.”

Shopify

"Reflexive AI usage is now a baseline expectation."

Microsoft

"Using AI is no longer optional."

Shopify

“Reflexive AI usage is now a baseline expectation.”

Microsoft

“Using AI is no longer optional.”

Duolingo

“Duolingo is going to be AI-first.”

Shopify

"Reflexive AI usage is now a baseline expectation."

Microsoft

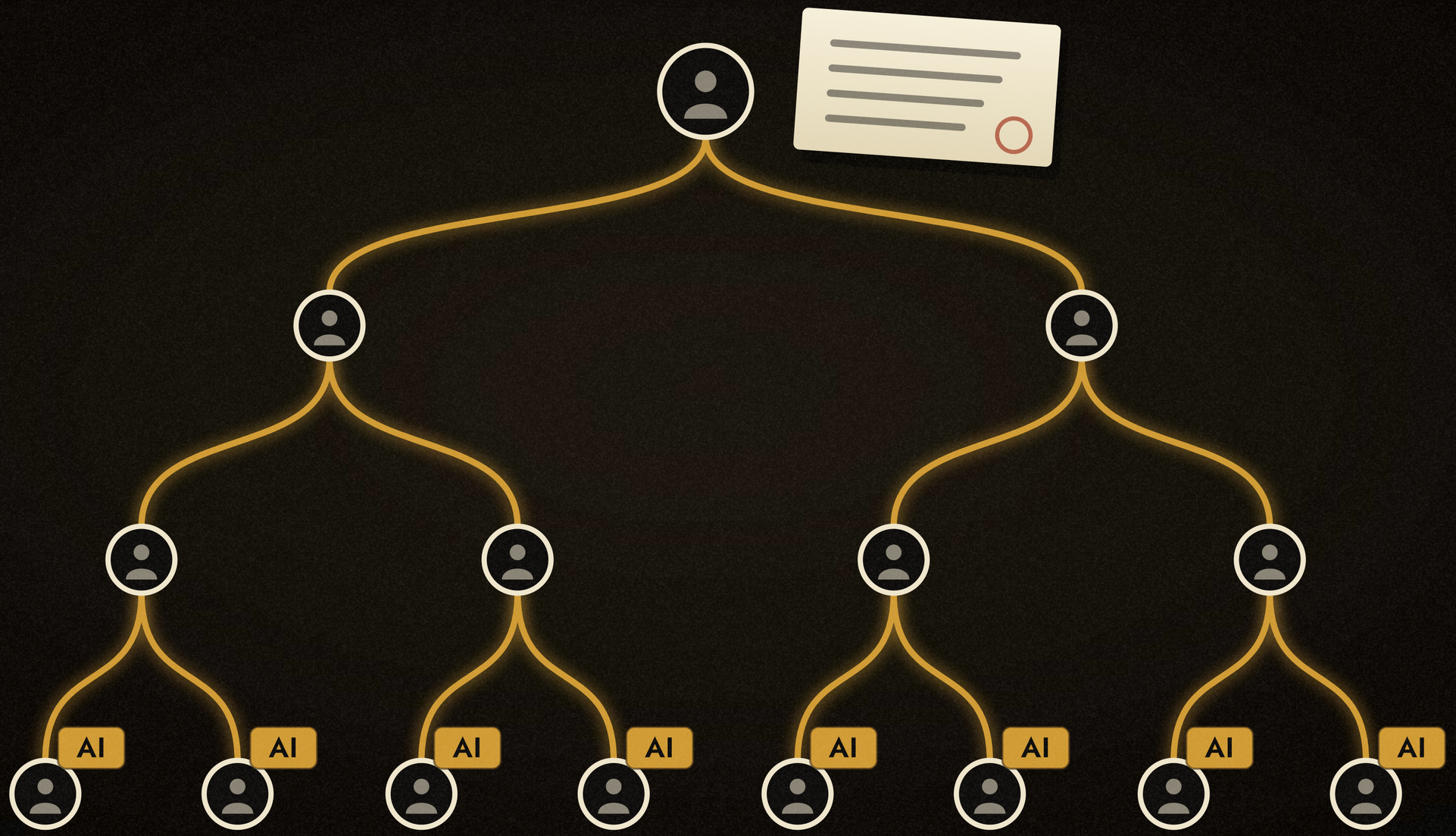
"Using AI is no longer optional."

Duolingo

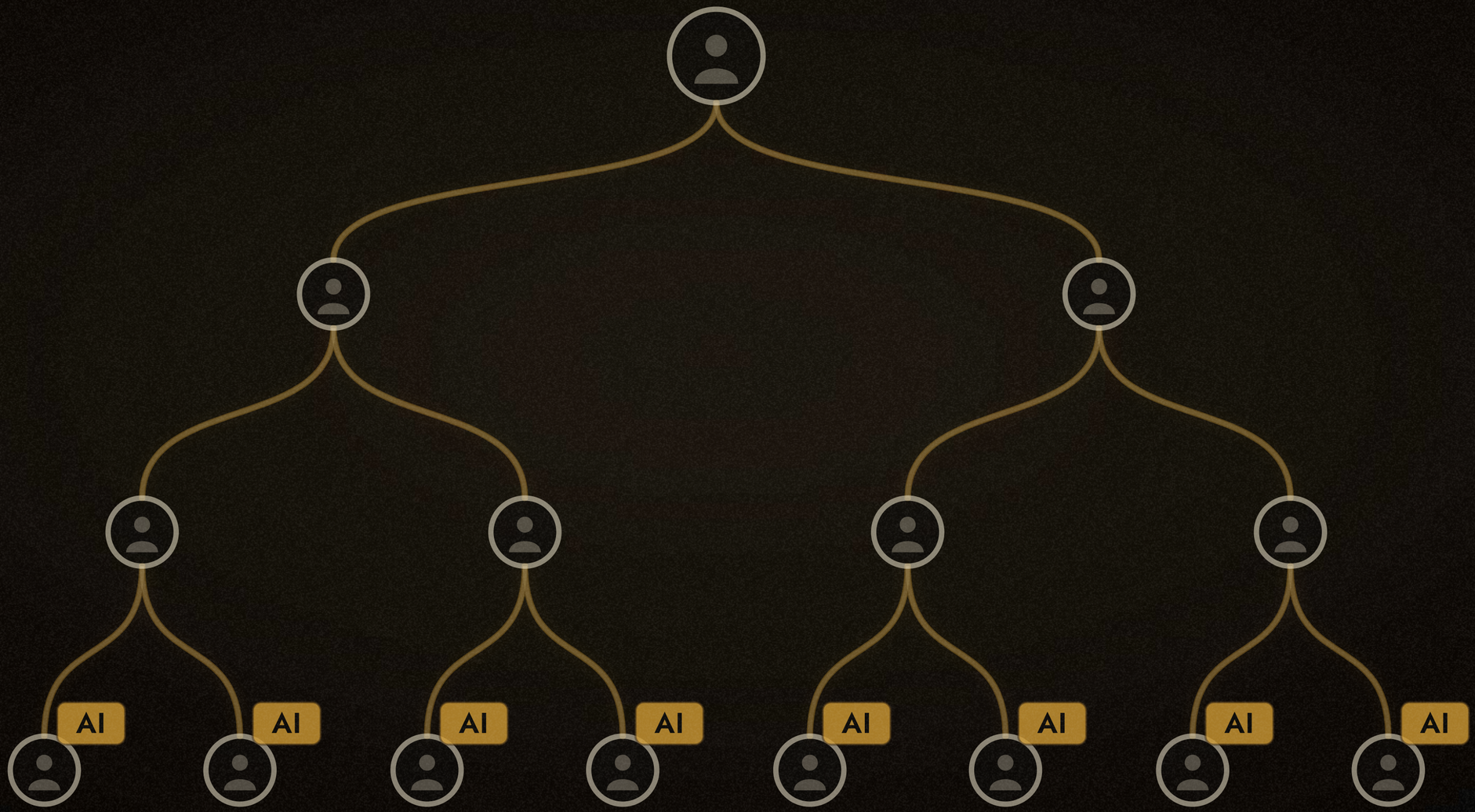
"Duolingo is going to be AI-first."

coinbase

Onboard by the end of the week.

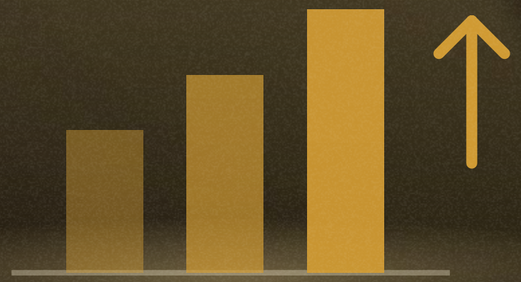

AI
AI
AI
AI
AI
AI
AI
AI

Licences **100%** Adoption **94%** Tokens ↑ **40×**



Hawthorne Works, 1924



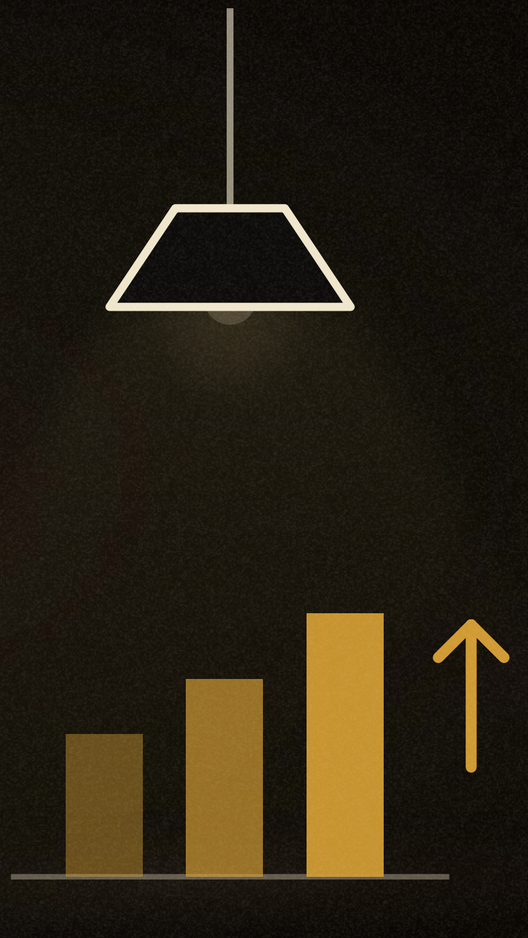


lights up · output up

Hawthorne Works, 1924



lights up · output up

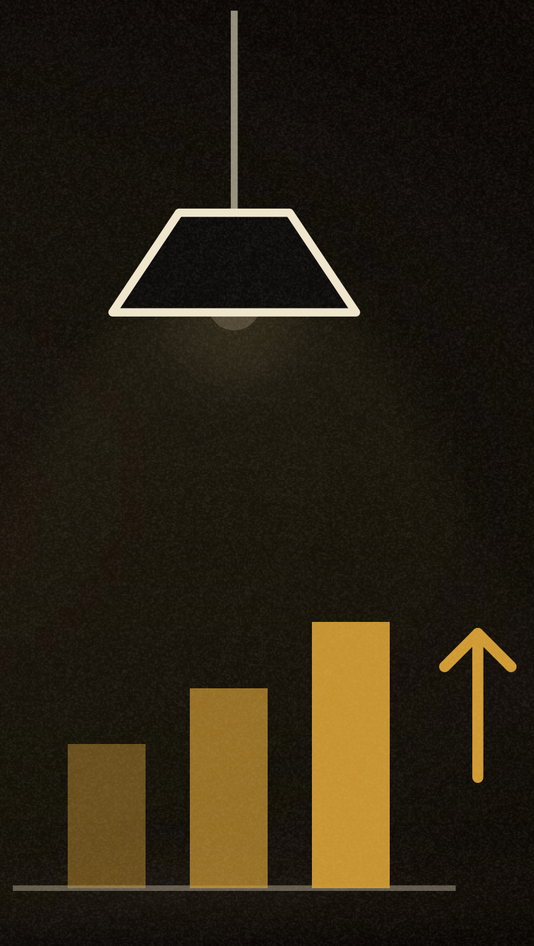


lights down · output up

Hawthorne Works, 1924



lights up · output up



lights down · output up

They were being watched.

Optimising for the wrong metric does not help.

Uber

The Verge · 26 May 2026

Uber president says AI spending is getting 'harder to justify'

"That link is not there yet." Annual AI budget reportedly gone in four months.

Top-down automation.



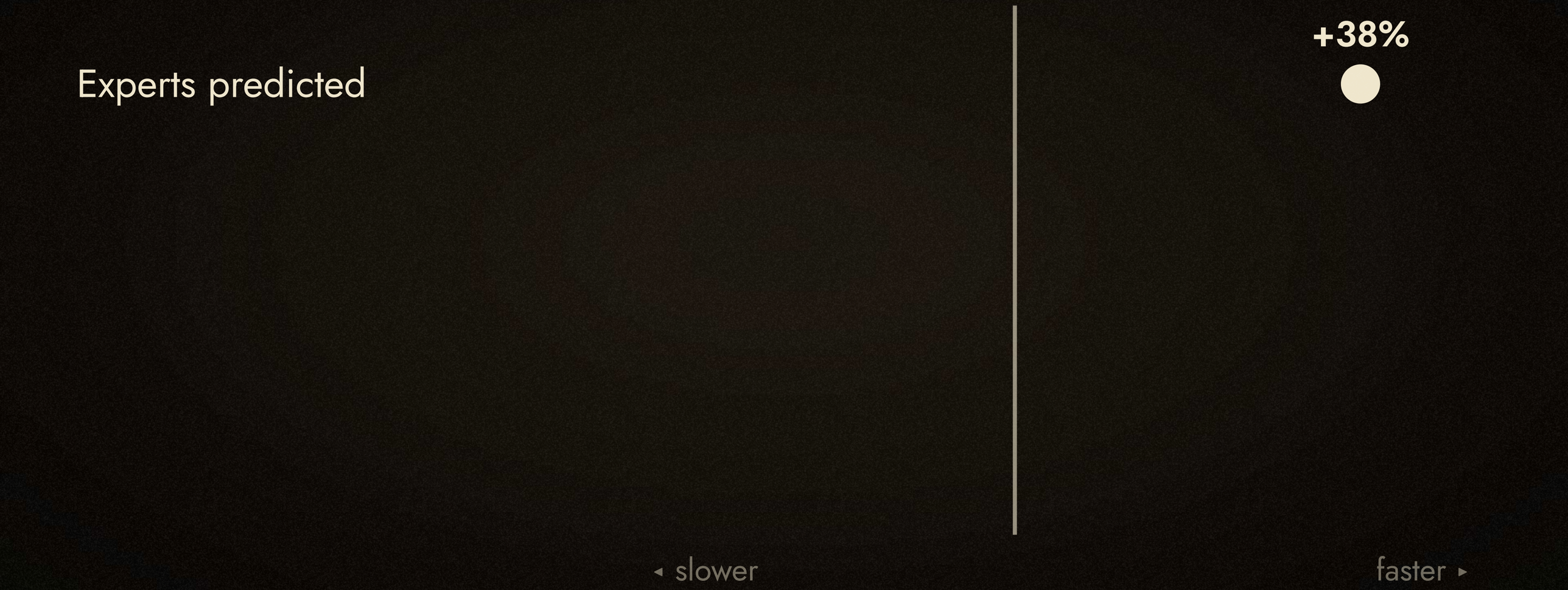
Buying robots.

It won’t.

“But my team is faster.”

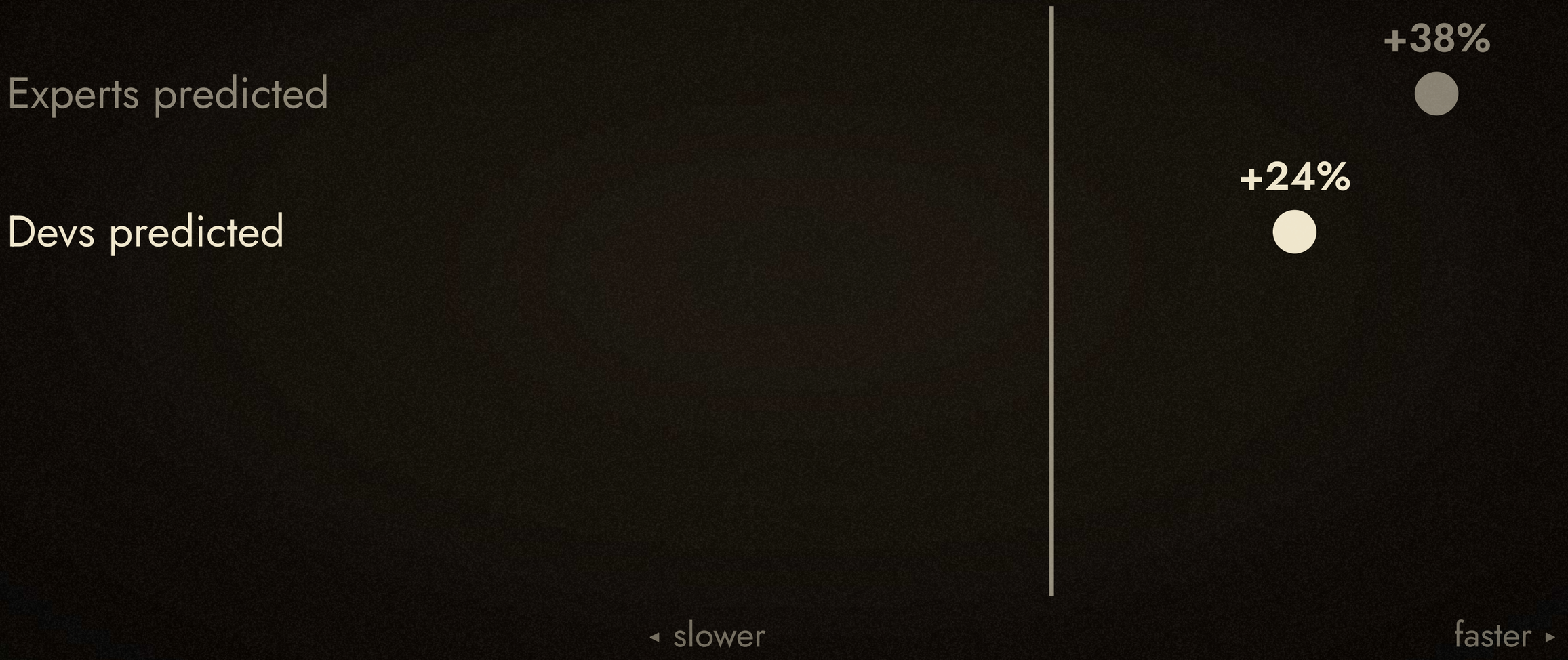
It only feels faster...

Felt vs measured



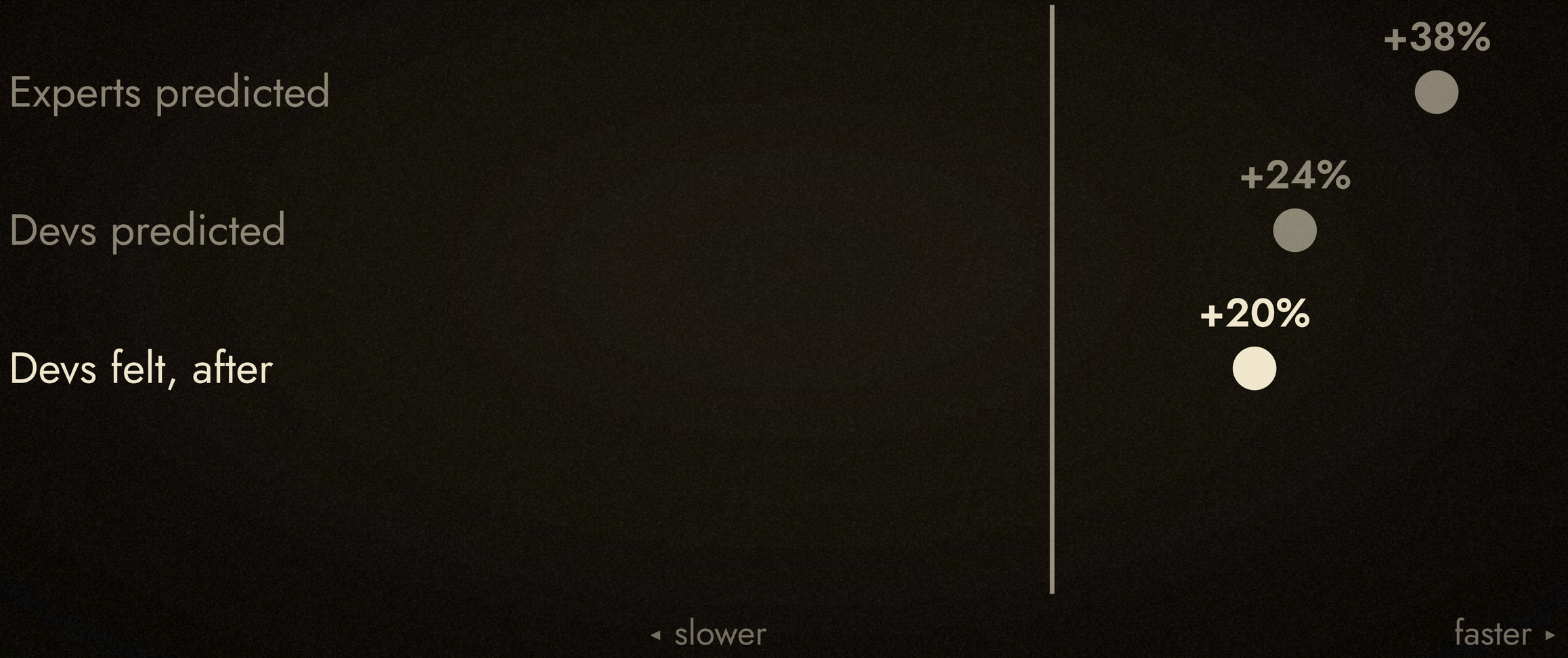
METR, July 2025 · 16 experienced OSS developers, 246 tasks

Felt vs measured



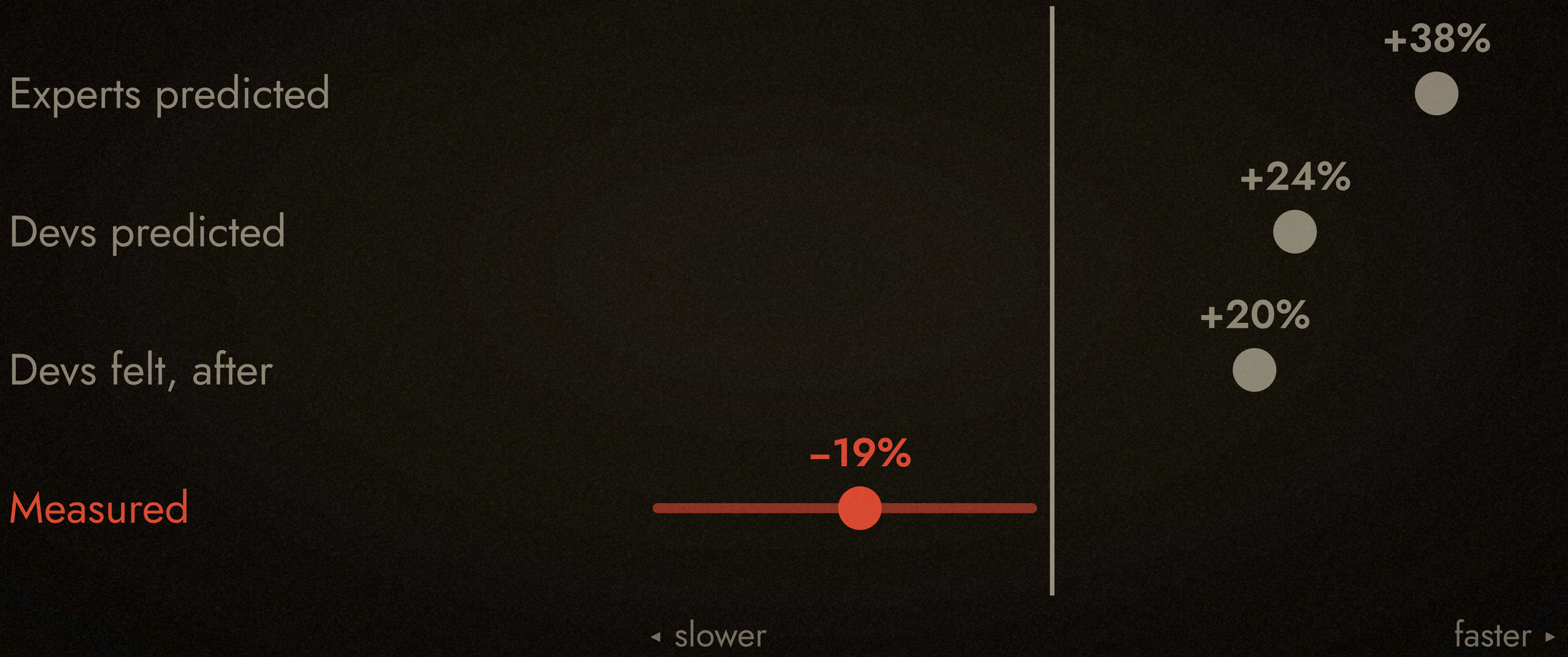
METR, July 2025 · 16 experienced OSS developers, 246 tasks

Felt vs measured



METR, July 2025 · 16 experienced OSS developers, 246 tasks

Felt vs measured



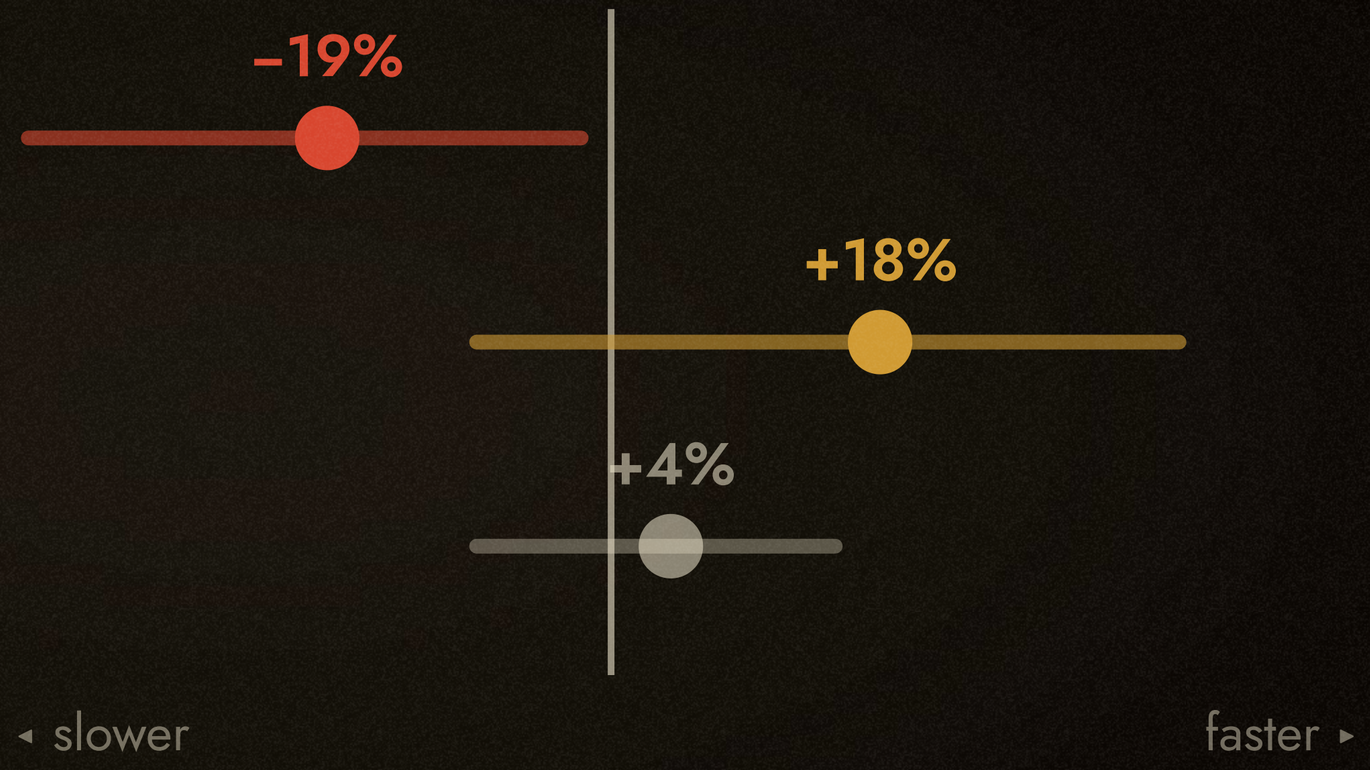
METR, July 2025 · 16 experienced OSS developers, 246 tasks

A year later

Early 2025

Late 2025, returning devs

Late 2025, new devs



METR, design-update post, Feb 2026 (late-2025 data)

−7.2%

delivery stability, for every 25% more AI adoption

DORA, 2024

“No significant correlation between AI adoption and improvements at the company level.”

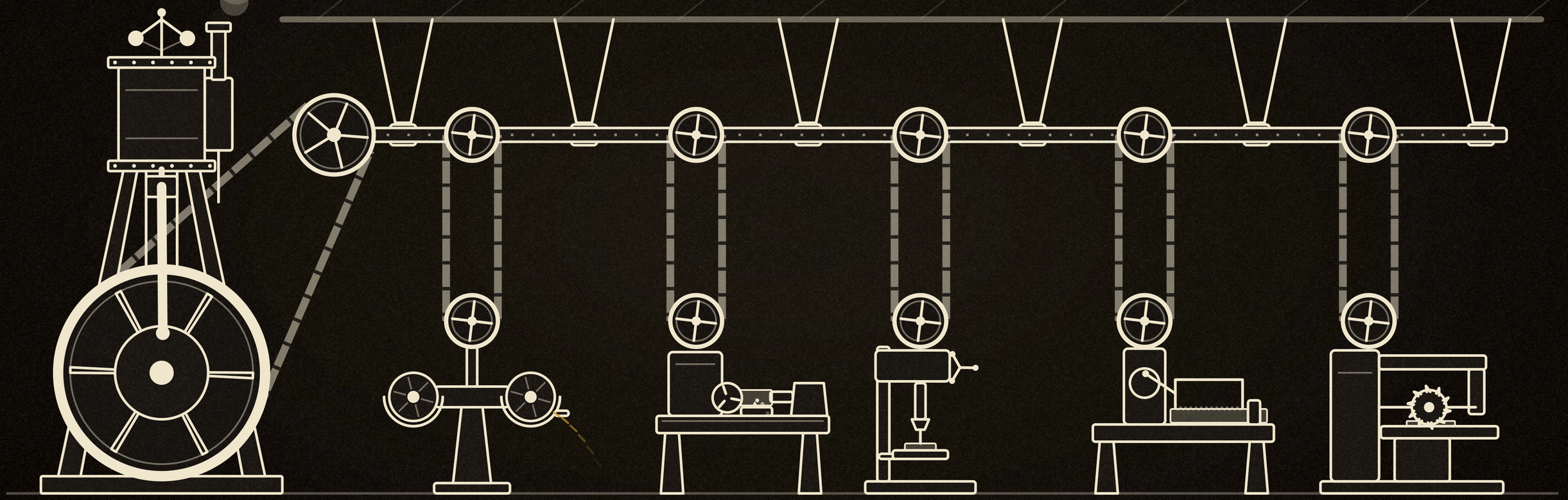
Faros AI, 10,000+ developers, 2025

“AI doesn’t fix a team; it amplifies what’s already there.”

DORA, 2025

1890s

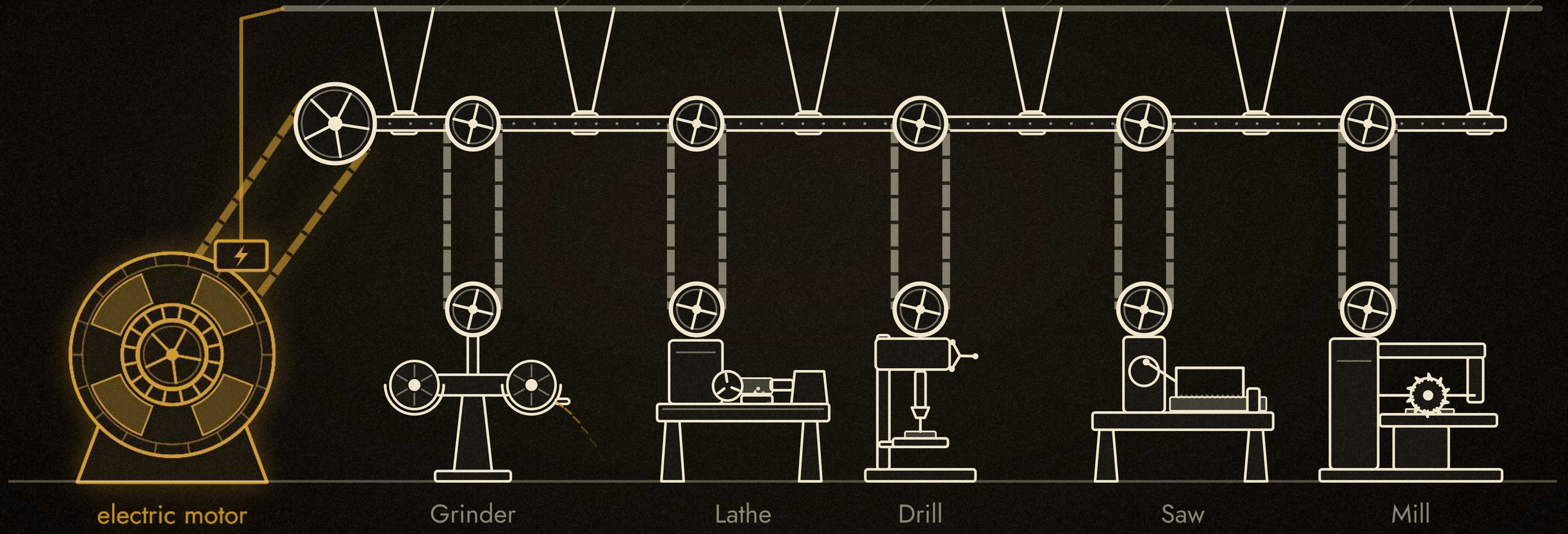
One shaft drives every machine, so the floor is laid out around the shaft



One engine. One shaft.

1900s

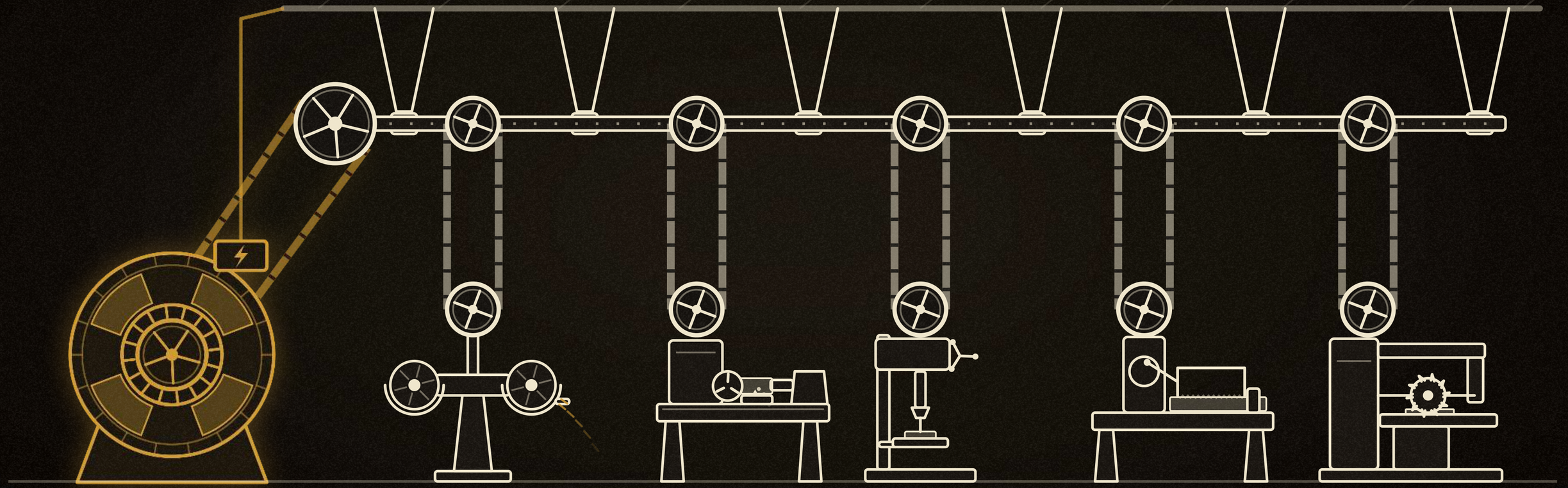
One shaft drives every machine, so the floor is laid out around the shaft



New motor. Same floor.

1910s

One shaft drives every machine, so the floor is laid out around the shaft

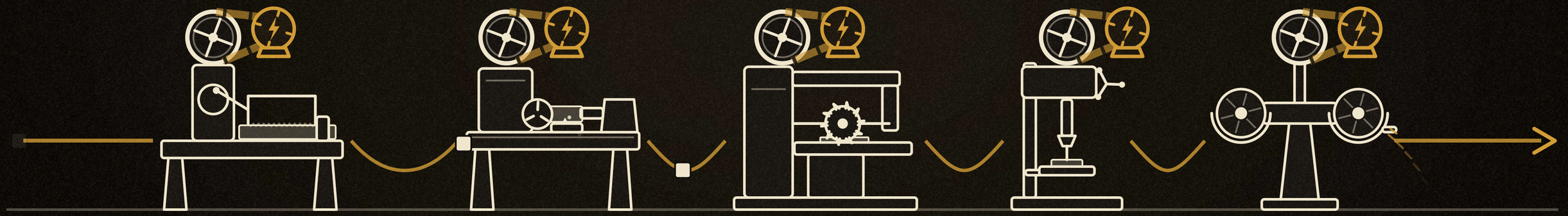


New motor. Same floor.

Productivity barely moved for decades.

1920s

A small motor in every machine; machines placed in the order the work flows

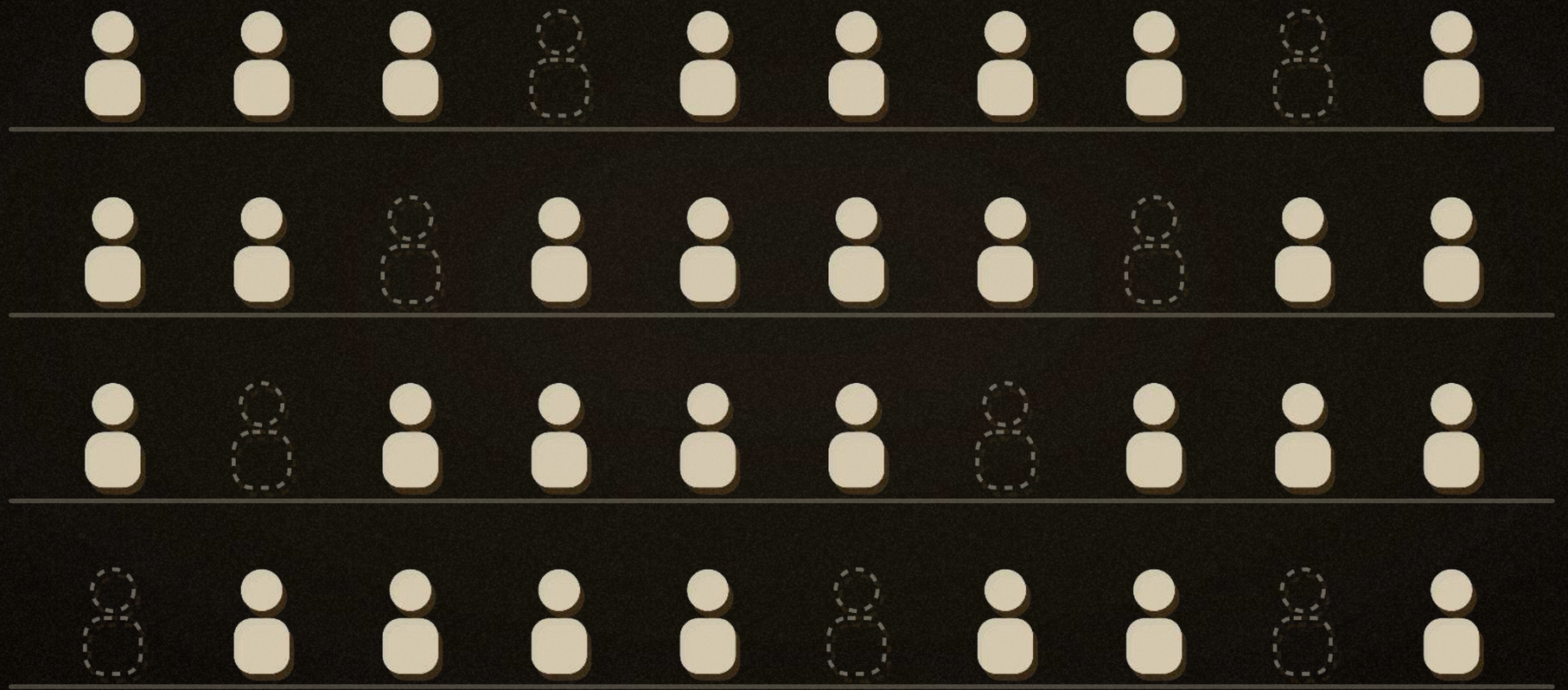


Then they redesigned the floor.


Fremont, California. 1982.

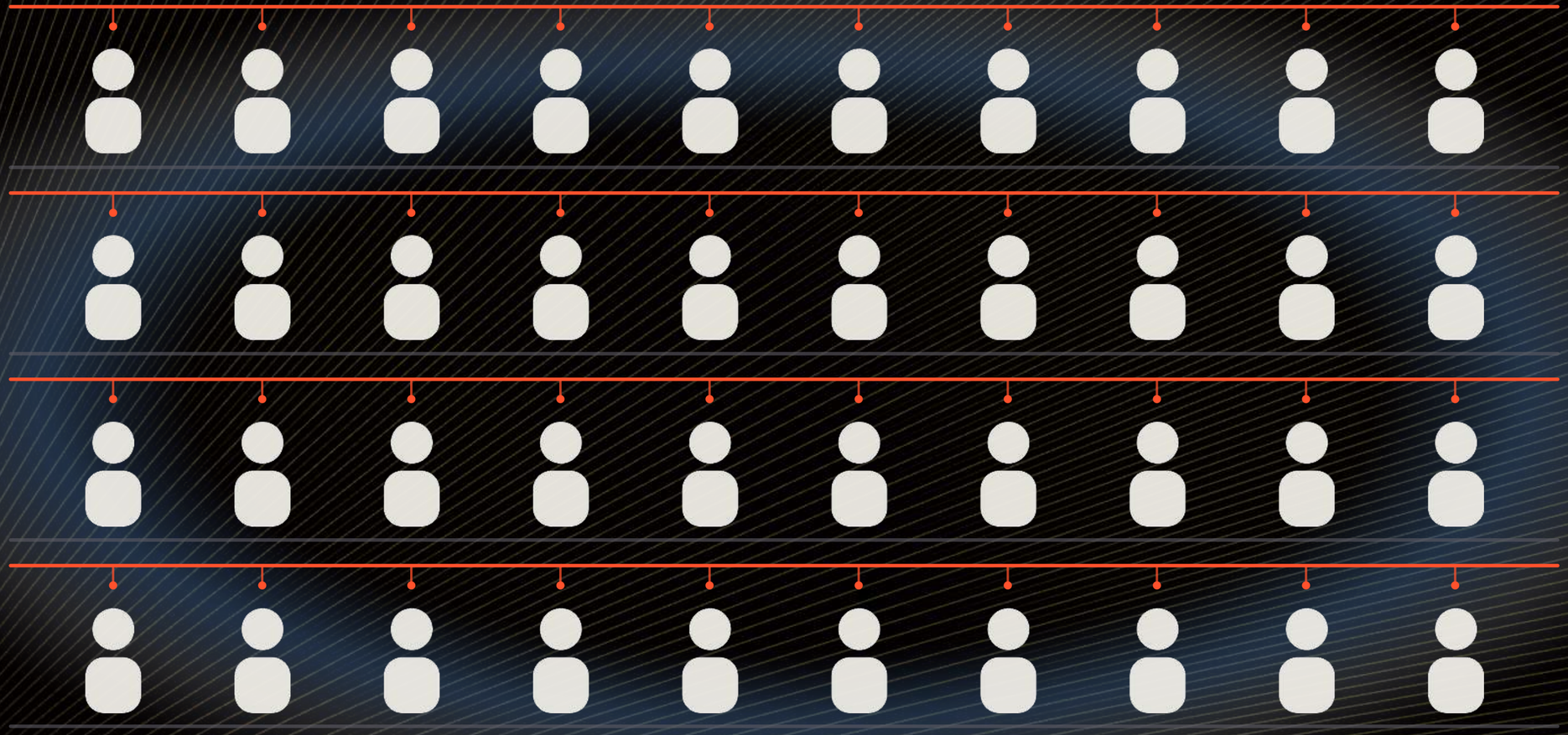
GM

1982



On a typical day, one in five stations is empty.

TOYOTA 1984



Same people. 85% of the workforce came from the old GM plant. Absenteeism: 2–3%.

	GM Framingham	TOYOTA Takaoka
hours per car	31	16
defects per 100 cars	135	45

MIT, 1987 plant data (Womack, Jones & Roos, 1990)

	GM Framingham	TOYOTA NUMMI Fremont	TOYOTA Takaoka
hours per car	31	19	16
defects per 100 cars	135	45	45

MIT, 1987 plant data (Womack, Jones & Roos, 1990)

It wasn’t the robots.

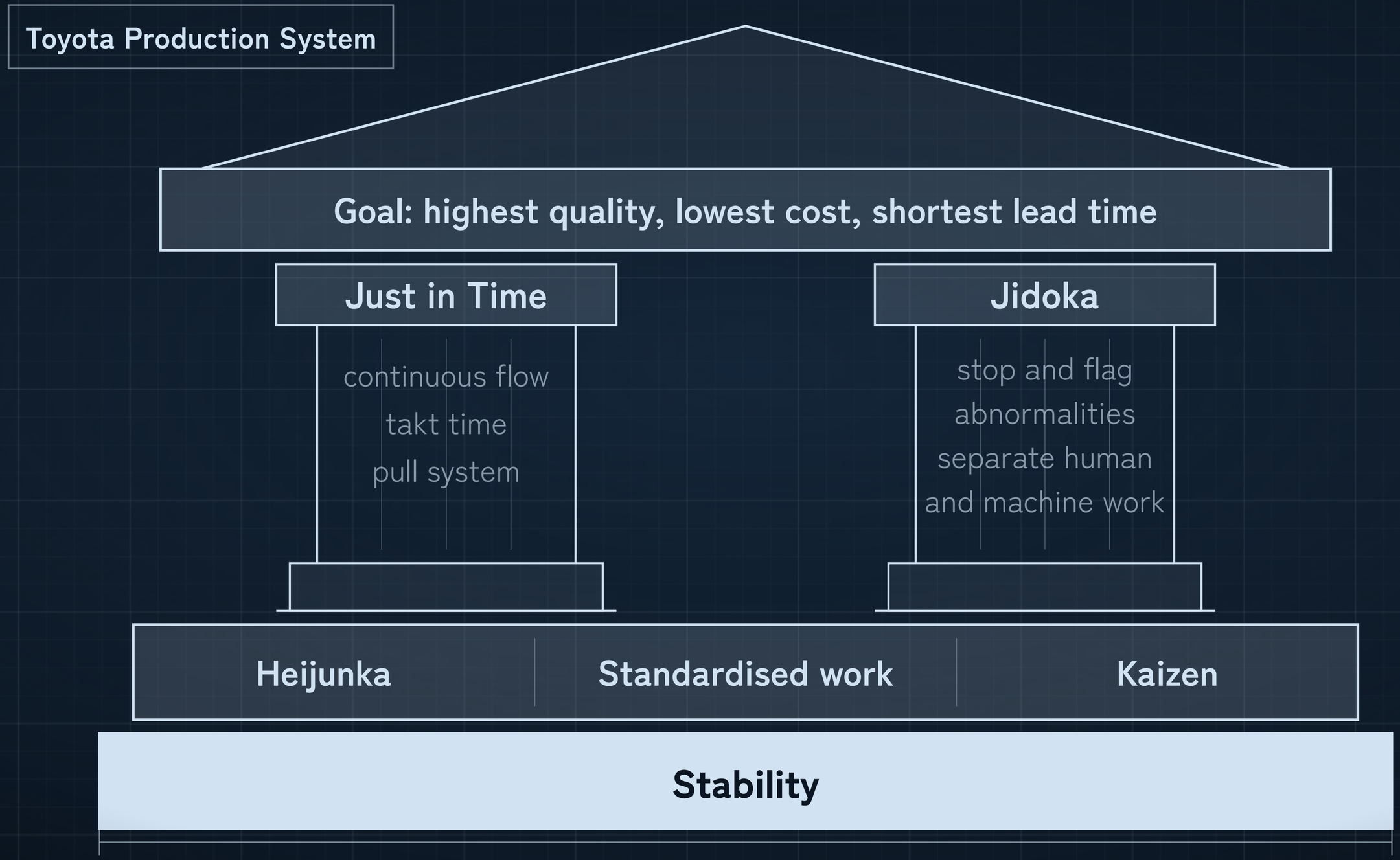
It wasn’t the robots.

It wasn’t the people.

It wasn’t the robots.

It wasn’t the people.

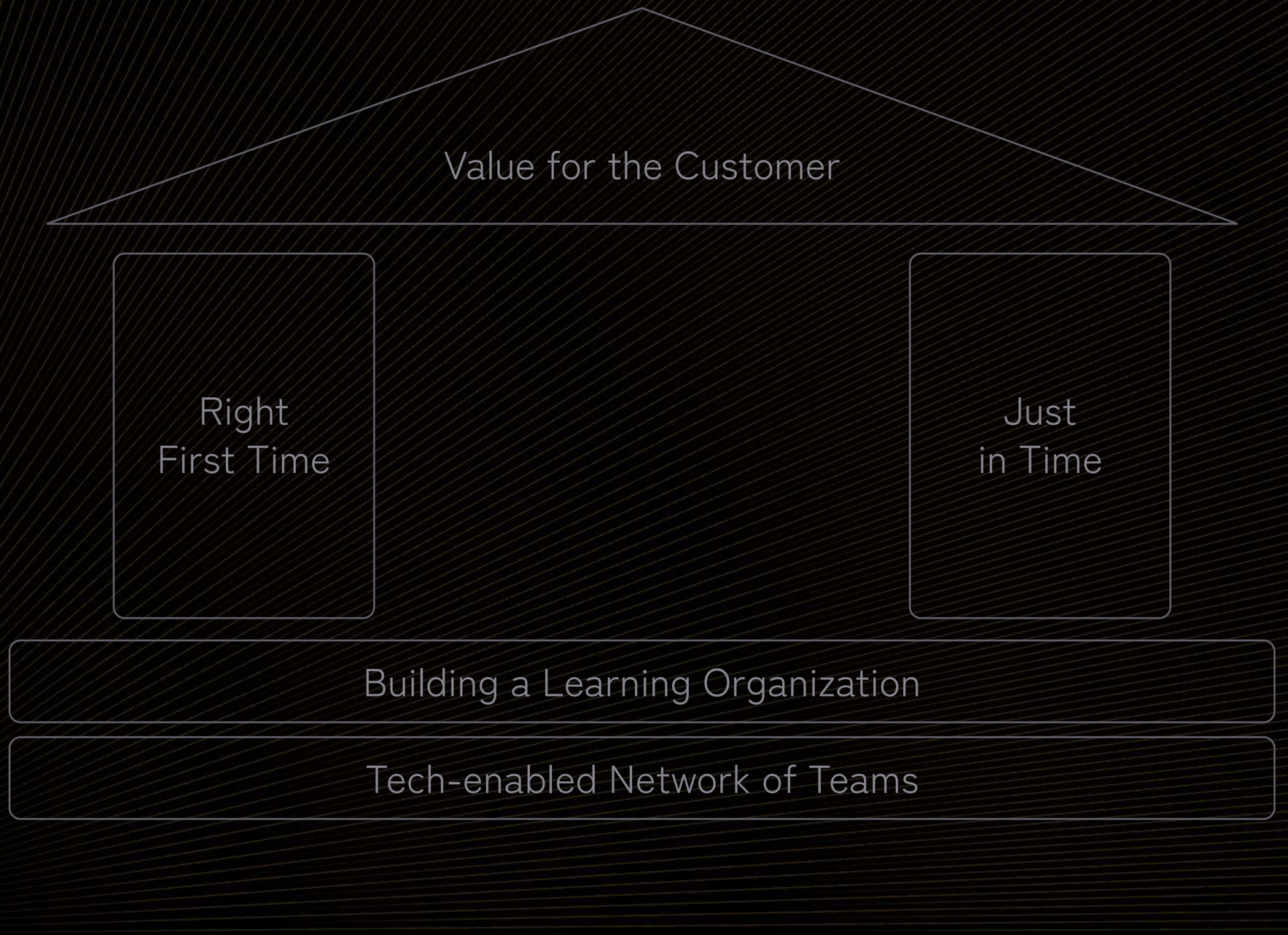
It was the system.

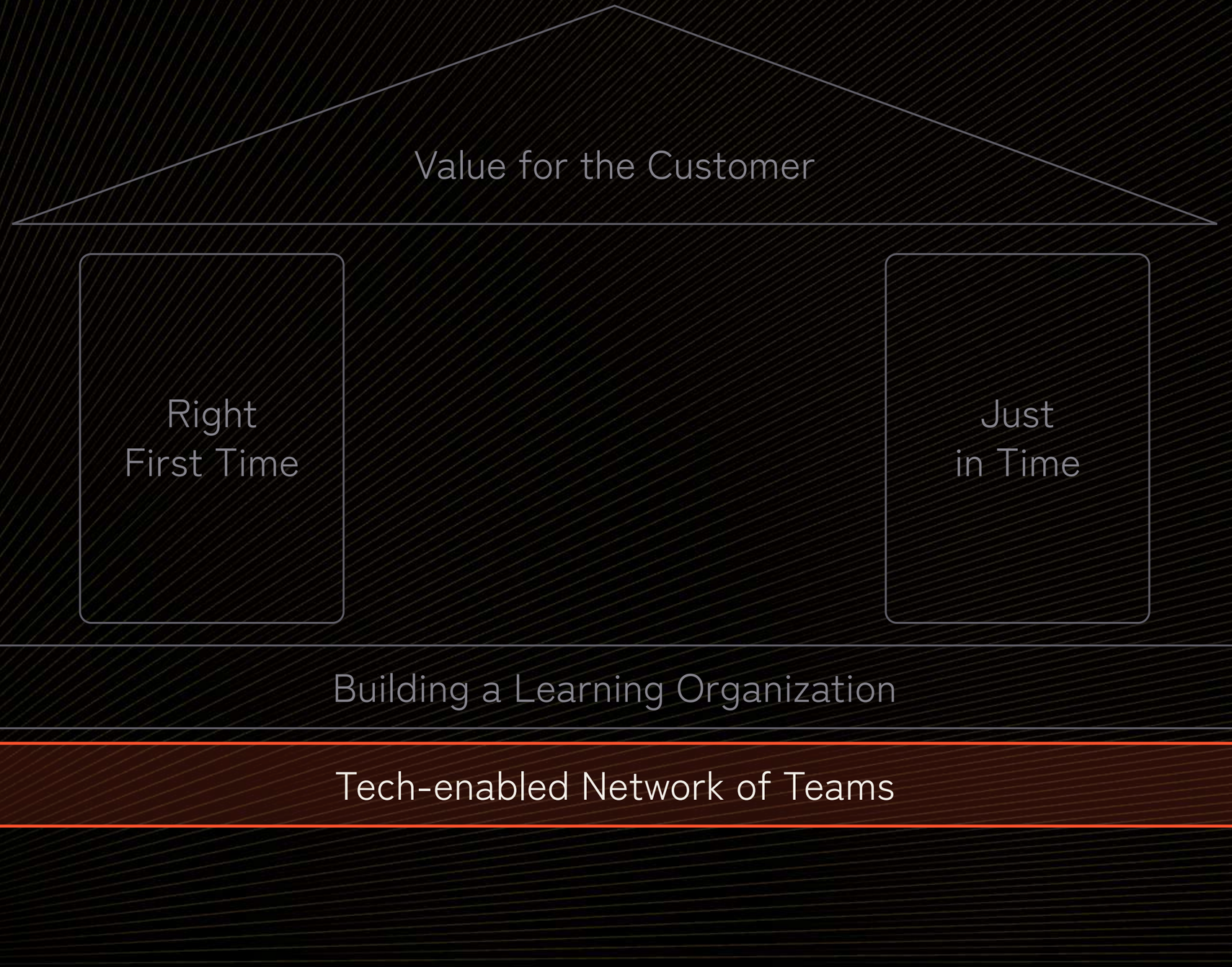

Toyota Production System
Goal: highest quality, lowest cost, shortest lead time
Just in Time
continuous flow
takt time
pull system
Jidoka
stop and flag
abnormalities
separate human
and machine work
Heijunka
Standardised work
Kaizen
Stability

How do we create the right system for software in the age of AI?



Toyota’s system, applied to tech teams.

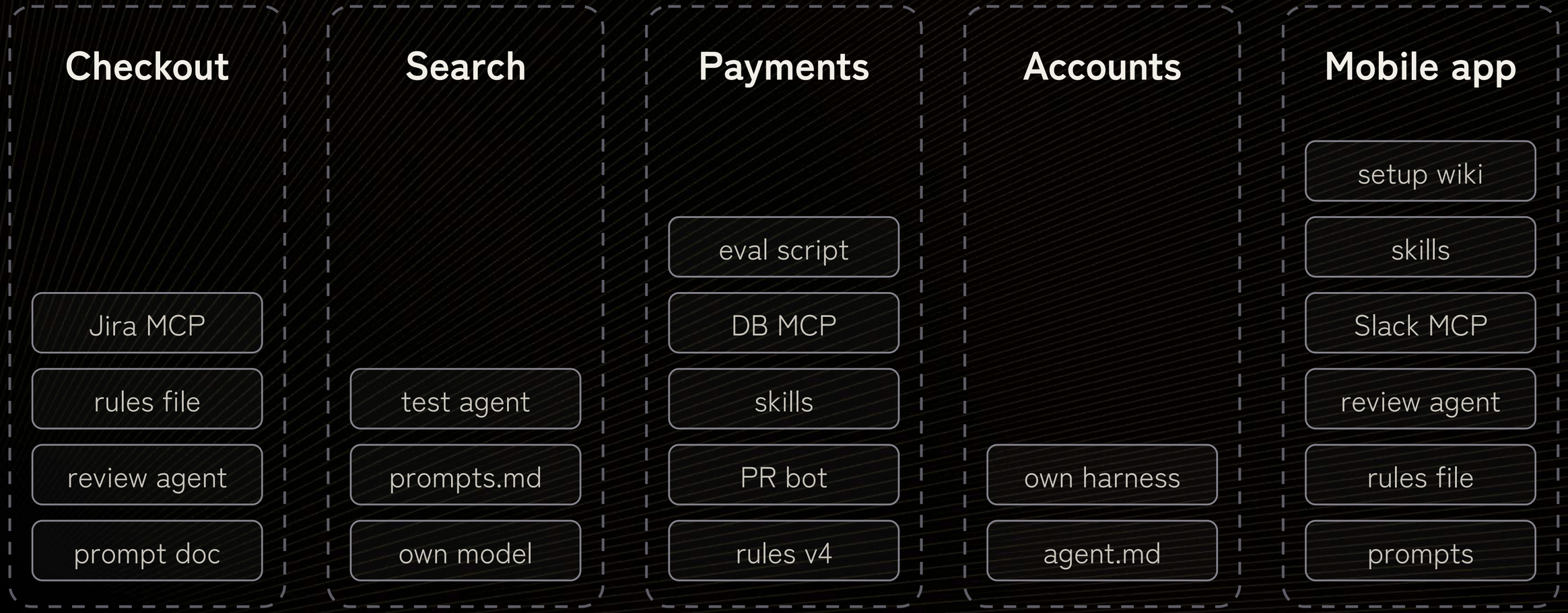

Value for the Customer
Right
First Time
Just
in Time
Building a Learning Organization
Tech-enabled Network of Teams


Value for the Customer
Right
First Time
Just
in Time
Building a Learning Organization
Tech-enabled Network of Teams

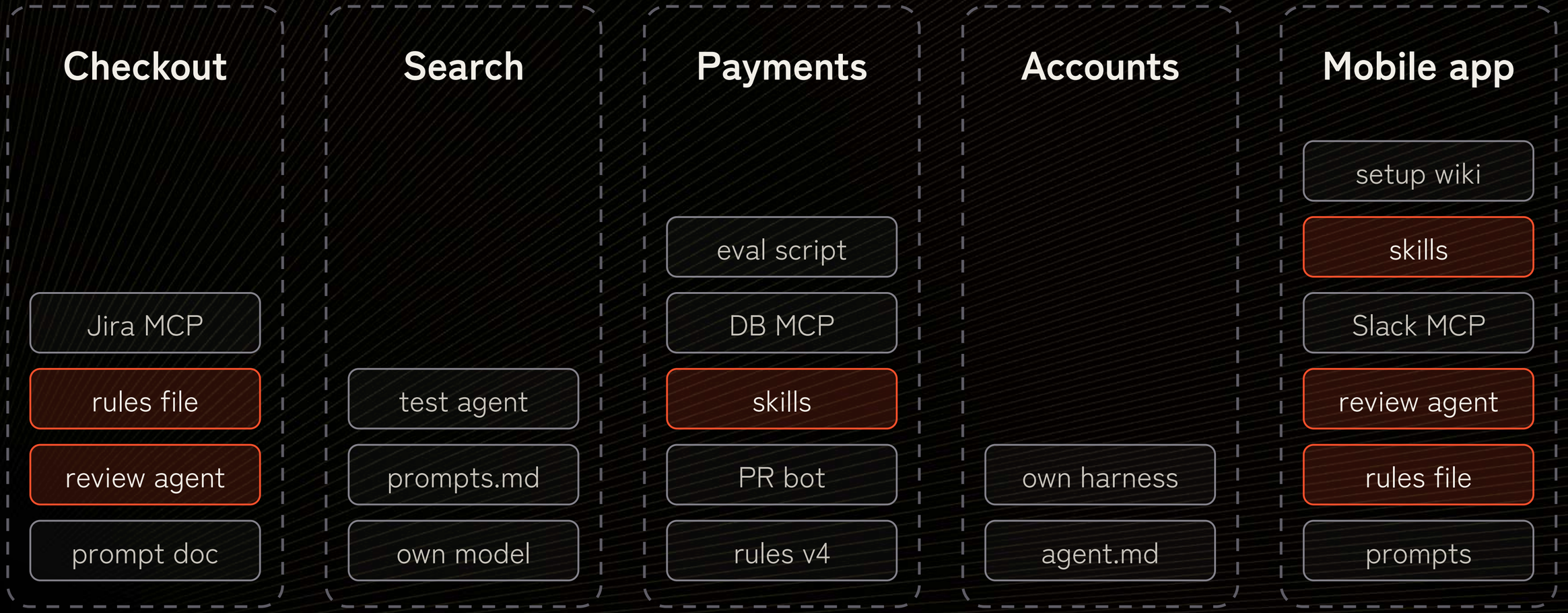
Every team gets AI.

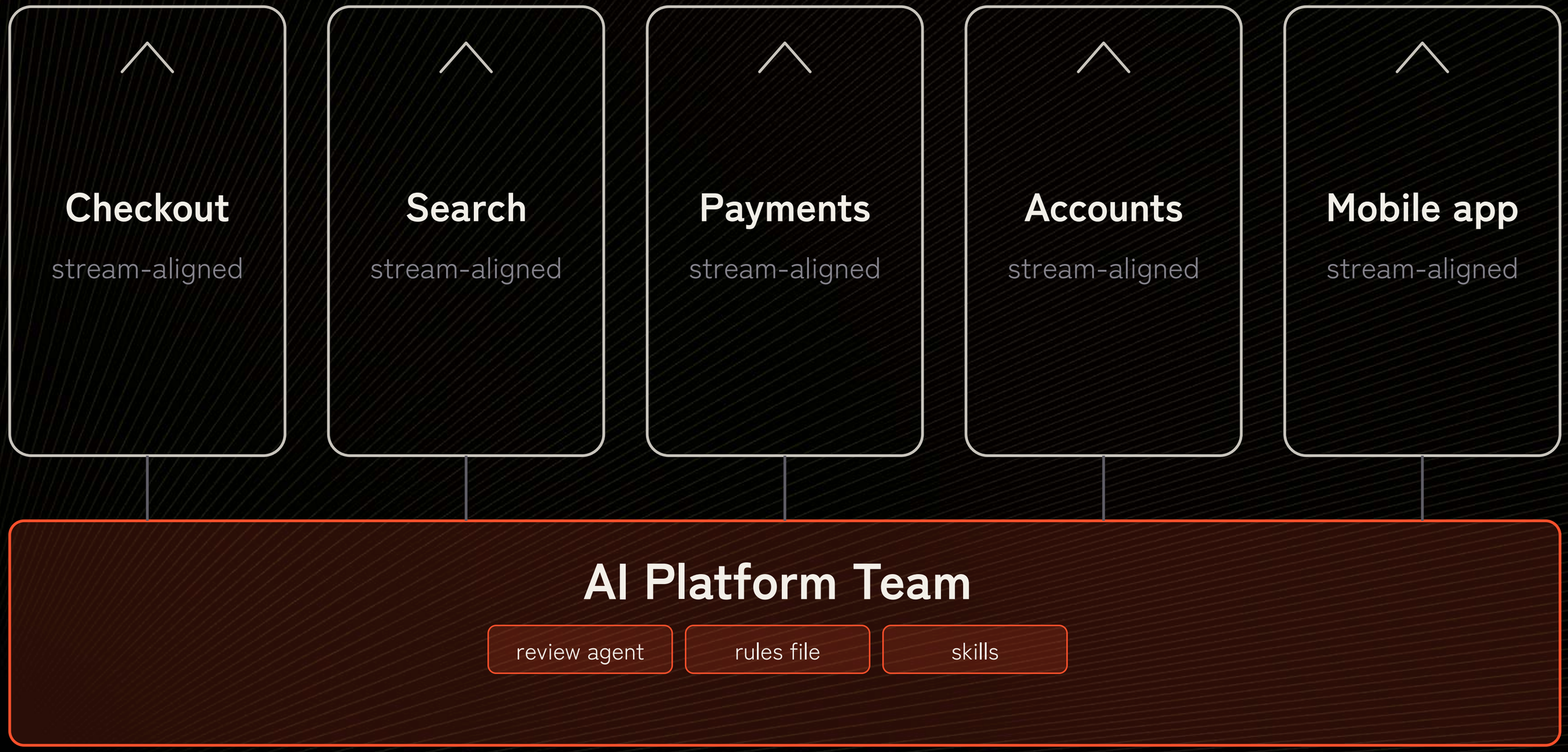


Every team gets AI.



Every team gets AI.




Checkout
stream-aligned
Search
stream-aligned
Payments
stream-aligned
Accounts
stream-aligned
Mobile app
stream-aligned
AI Platform Team
review agent
rules file
skills

AI Platform Team

- Model gateway
- Cost controls
- Audit logging
- Observability
- Harnesses and skills
- Review agent and guardrails

AI Platform Team

- Model gateway
- Cost controls
- Audit logging
- Observability
- Harnesses and skills
- Review agent and guardrails

What we measure

AI Platform Team

- Model gateway
- Cost controls
- Audit logging
- Observability
- Harnesses and skills
- Review agent and guardrails

What we measure

DORA

Lead time, deploy frequency, change failure rate, time to recover

AI Platform Team

- Model gateway
- Cost controls
- Audit logging
- Observability
- Harnesses and skills
- Review agent and guardrails

What we measure

DORA

Lead time, deploy frequency, change failure rate, time to recover

AI telemetry

Adoption, usage patterns, cost per ticket

AI Platform Team

- Model gateway
- Cost controls
- Audit logging
- Observability
- Harnesses and skills
- Review agent and guardrails

What we measure

DORA

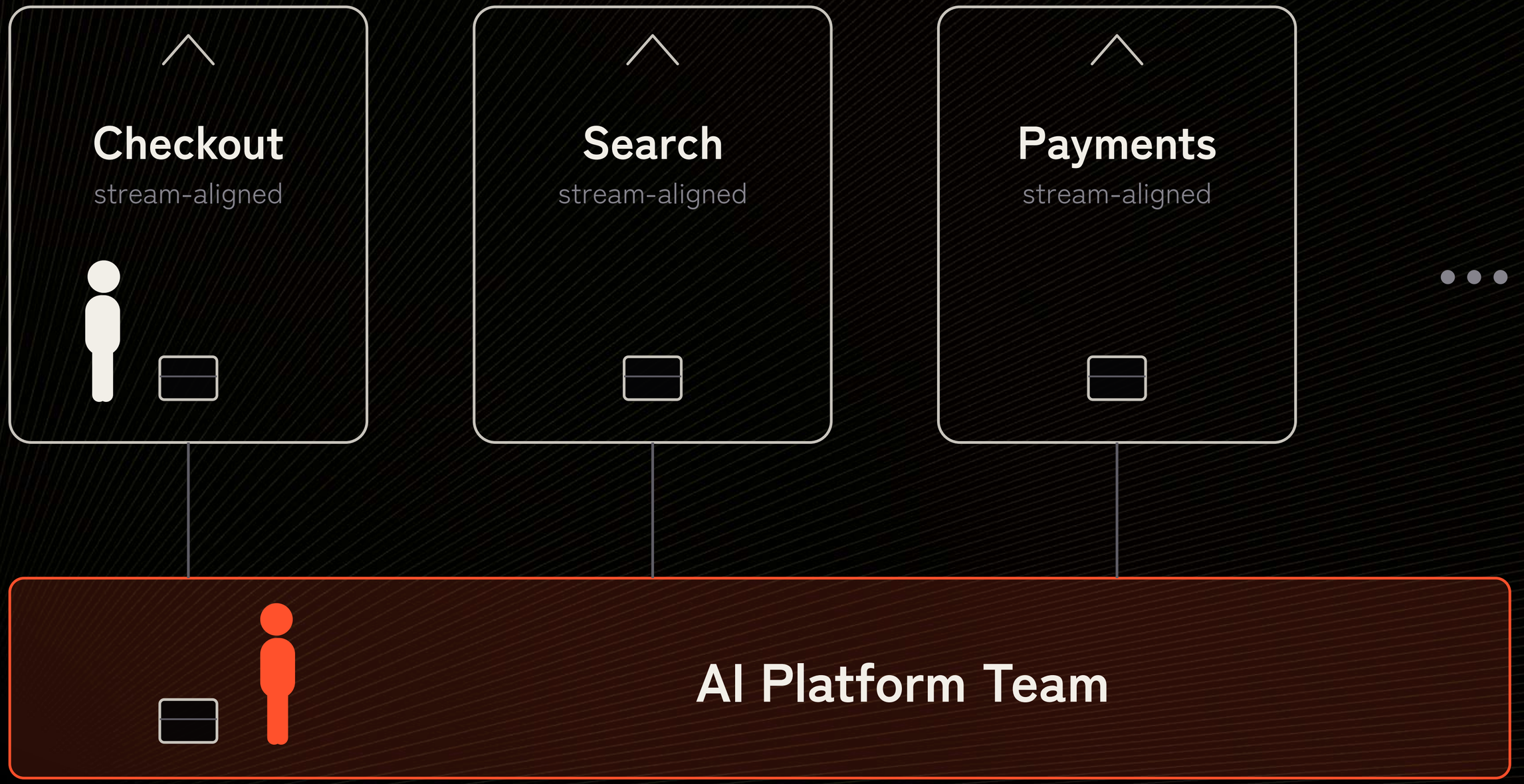
Lead time, deploy frequency, change failure rate, time to recover

AI telemetry

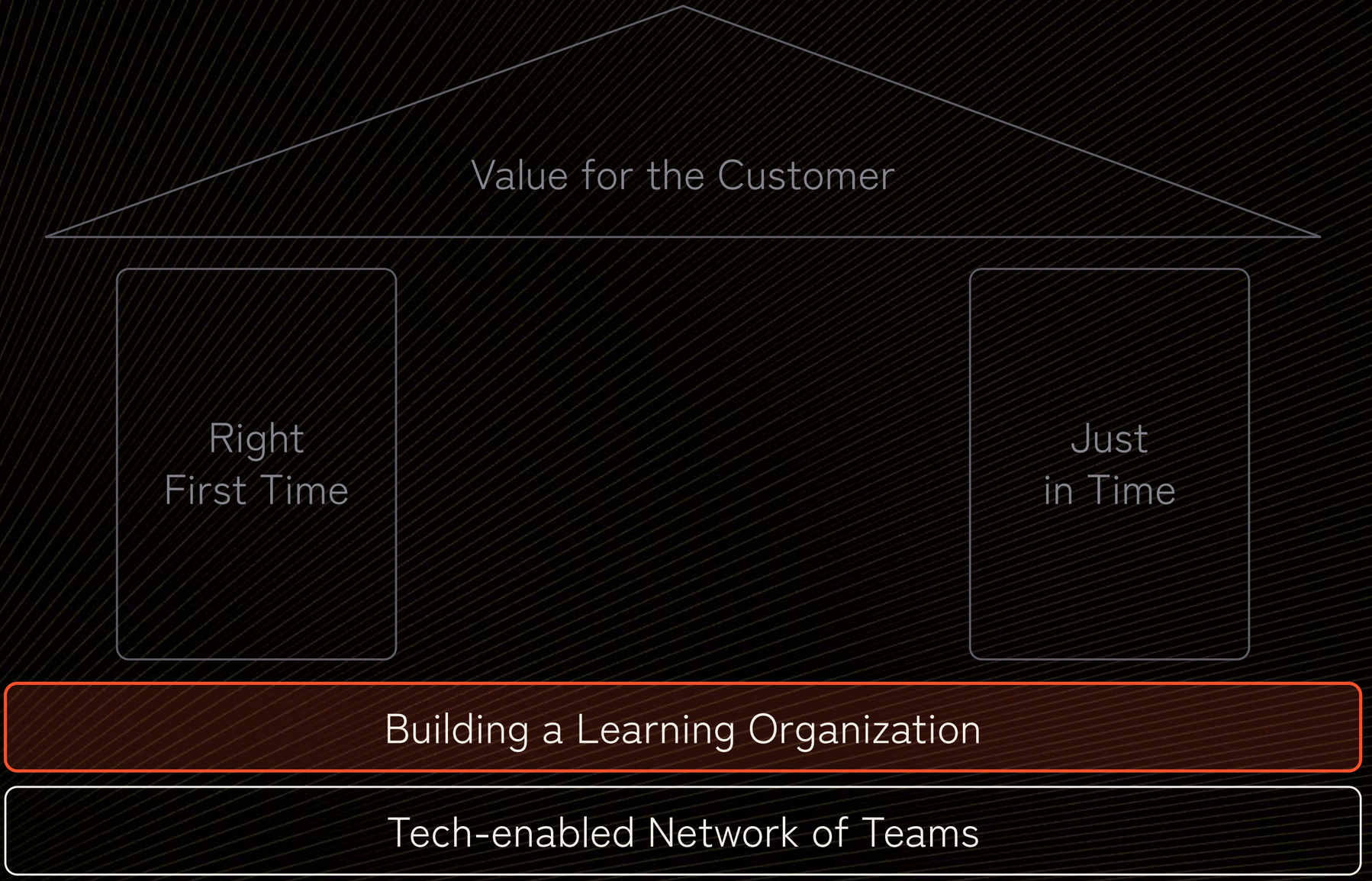
Adoption, usage patterns, cost per ticket

Right first time

Share of work shipped without rework



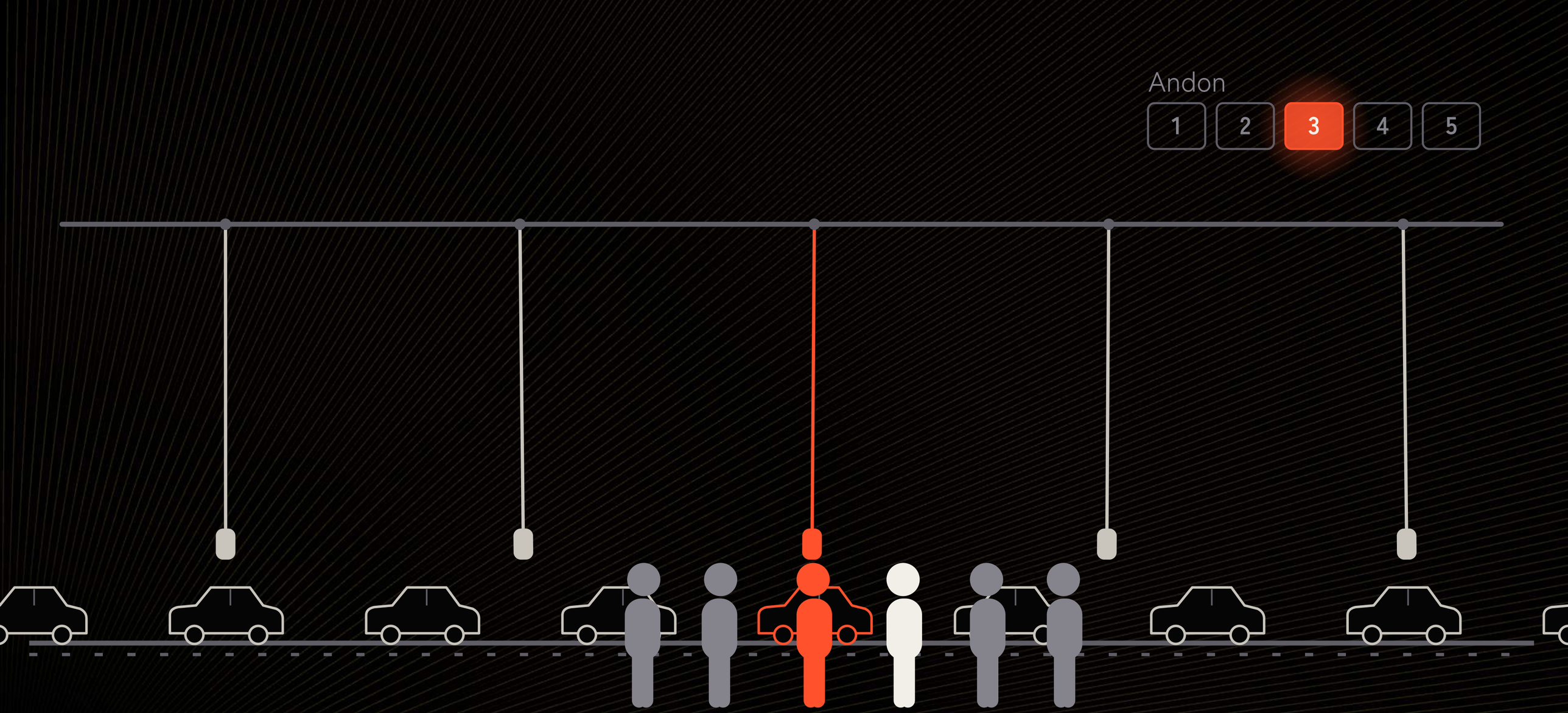
The Platform Team embeds and brings back learnings to the rest of the organisation.


Value for the Customer
Right
First Time
Just
in Time
Building a Learning Organization
Tech-enabled Network of Teams

改善

Kaizen

Change for the better.



The line stops. Everyone comes to learn.

The machine stopped.

Why? **An overload: the fuse blew.**

The machine stopped.

Why? An overload: the fuse blew.

Why? The bearing wasn’t lubricated.

The machine stopped.

Why? An overload: the fuse blew.

Why? The bearing wasn’t lubricated.

Why? The lubrication pump wasn’t pumping.

The machine stopped.

Why? An overload: the fuse blew.

Why? The bearing wasn’t lubricated.

Why? The lubrication pump wasn’t pumping.

Why? Its shaft was worn and rattling.

The machine stopped.

Why? An overload: the fuse blew.

Why? The bearing wasn’t lubricated.

Why? The lubrication pump wasn’t pumping.

Why? Its shaft was worn and rattling.

Why? No strainer. Metal scrap got in.

The machine stopped.

Why? An overload: the fuse blew.

Why? The bearing wasn’t lubricated.

Why? The lubrication pump wasn’t pumping.

Why? Its shaft was worn and rattling.

Why? No strainer. Metal scrap got in.

Fit a strainer.

Our agents were slow.

Why? **55% of their time went on validating. 36% on building.**

Our agents were slow.

Why? 55% of their time went on validating. 36% on building.

Why? Merge conflicts between parallel agents?

Our agents were slow.

Why? 55% of their time went on validating. 36% on building.

Why? ~~Merge conflicts between parallel agents?~~ Only 3%.

Our agents were slow.

Why? 55% of their time went on validating. 36% on building.

Why? ~~Merge conflicts between parallel agents?~~ Only 3%.

Why? Every fix re-ran validation from scratch.

Our agents were slow.

Why? 55% of their time went on validating. 36% on building.

Why? ~~Merge conflicts between parallel agents?~~ Only 3%.

Why? Every fix re-ran validation from scratch.

Why? Parity tests ran on every single ticket.

Our agents were slow.

Why? **55% of their time went on validating. 36% on building.**

Why? ~~**Merge conflicts between parallel agents?**~~ Only 3%.

Why? **Every fix re-ran validation from scratch.**

Why? **Parity tests ran on every single ticket.**

Why? **Tickets queued to be validated one at a time.**

45% of “parallel” waves ran a single agent.

Our agents were slow.

Why? **55% of their time went on validating. 36% on building.**

Why? ~~**Merge conflicts between parallel agents?**~~ Only 3%.

Why? **Every fix re-ran validation from scratch.**

Why? **Parity tests ran on every single ticket.**

Why? **Tickets queued to be validated one at a time.**

45% of “parallel” waves ran a single agent.

Validate once, per slice.

Kaizen isn’t just for the team.

Kaizen isn’t just for the team.

It’s how we coach engineers one-on-one.

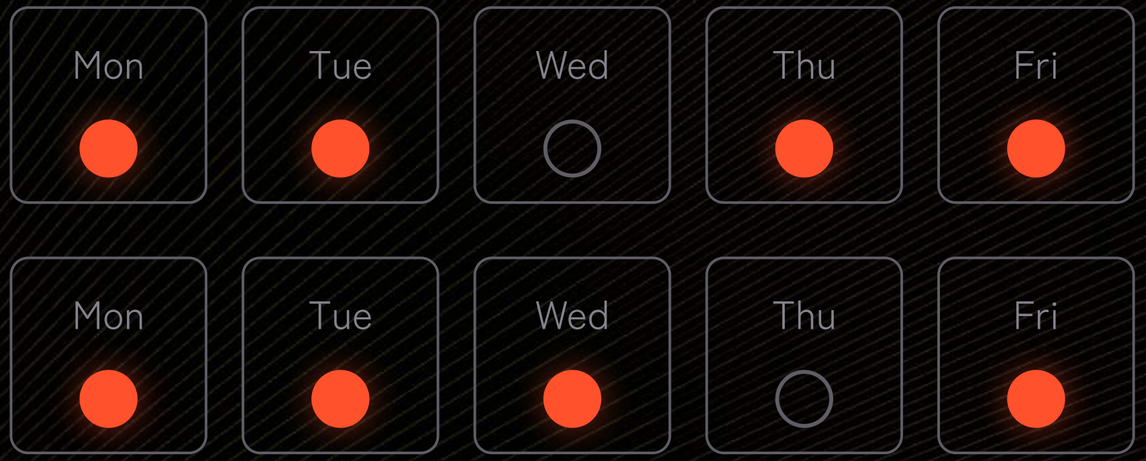
Kaizen becomes the 1-1.

```
› /kaizen-conv conversations/feat-free-shipping.txt

Reading the conversation…
3 friction points → 3 file changes

1  asked twice which test DB to use     → docs/testing.md
2  jOOQ type mismatch, fixed by hand    → docs/migrations.md
3  re-explained the PR template         → commands/pr.md

✓ feat-free-shipping-kaizen.md
```



It becomes part of the team culture.


London
Cape Town



28

experiments

daily problem solving, shared with the whole team



GitHub MCP vs CLI

Same flaky test, investigated two ways.

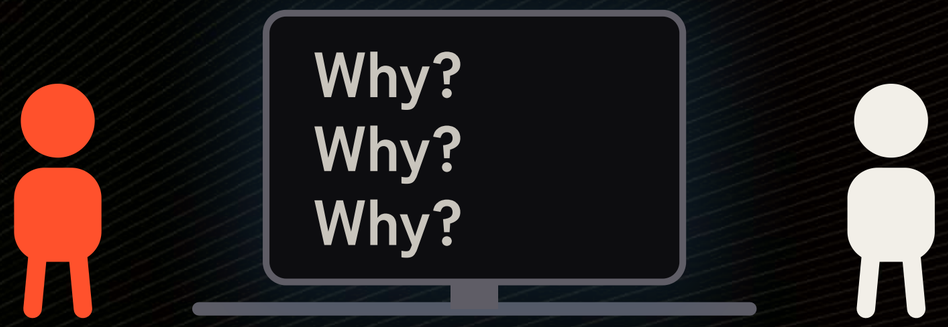
CLI	$1.49	Found the flaky test and suggested a fix.
MCP	$1.88	Got stuck on permissions.

“Overall, CLI wins.”

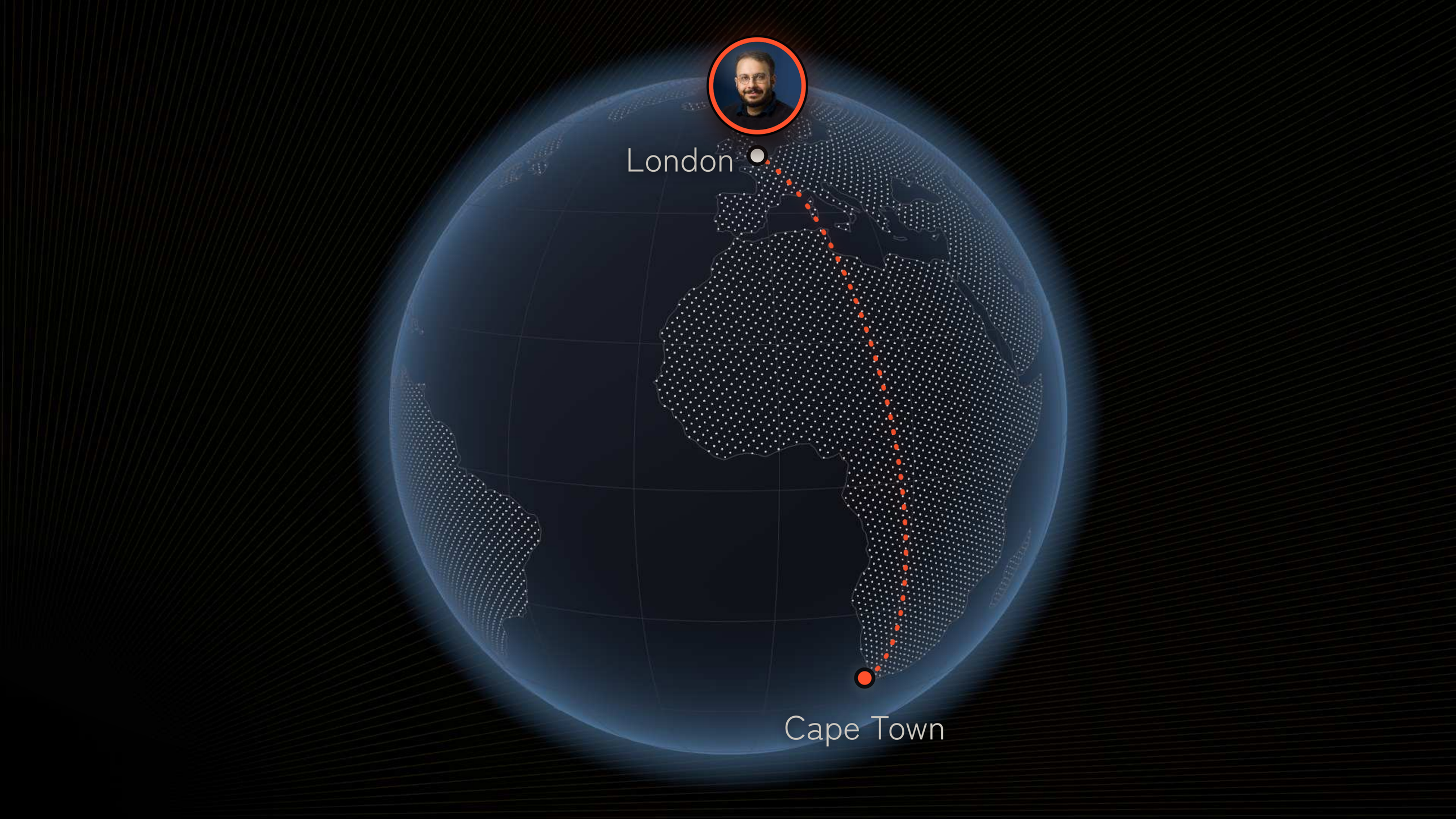
From the experiment wall, Cape Town

daily problem solving, shared with the whole team

Kaizen, 1-1



sitting with each engineer on their own problems


London
Cape Town

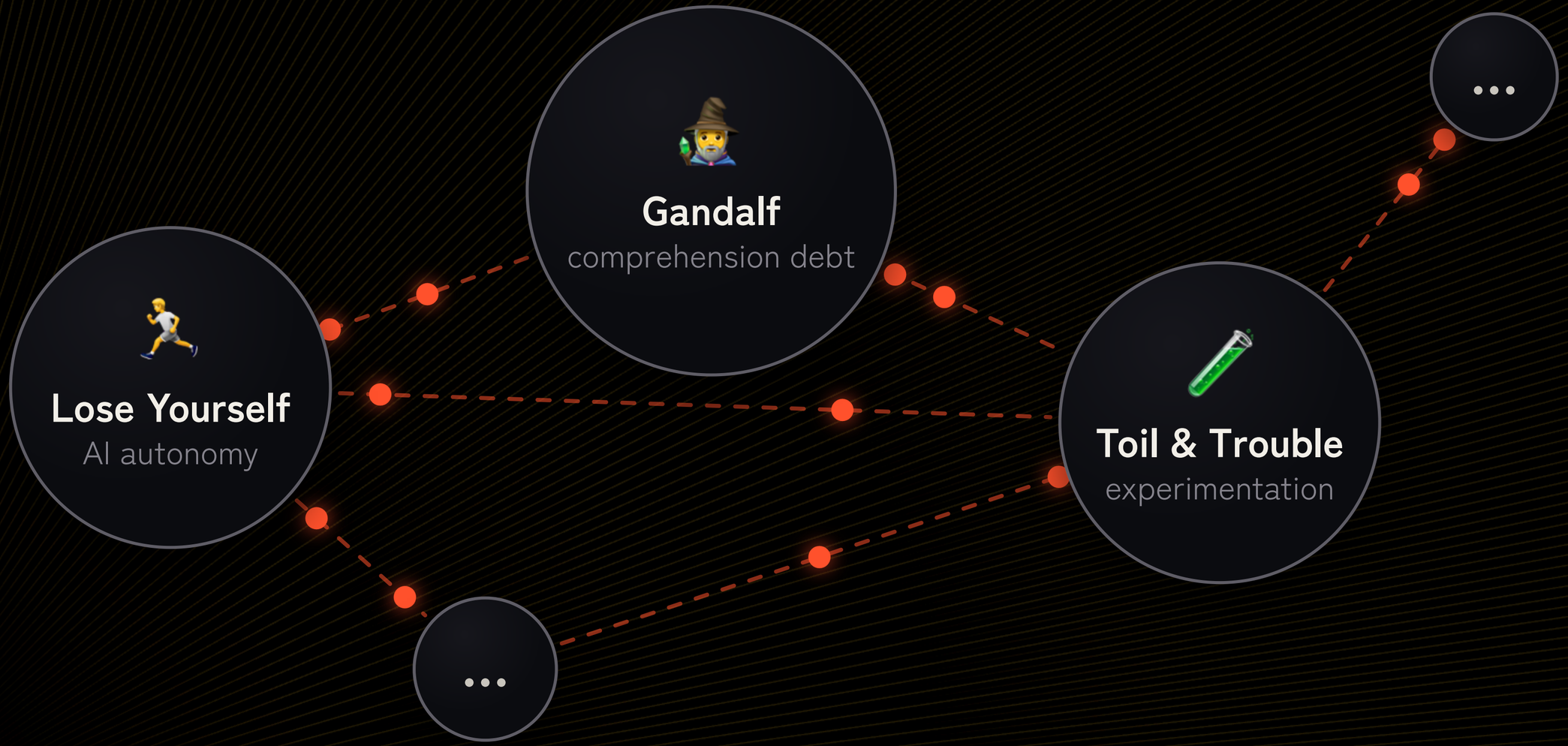
Cape Town, two weeks after I left



Dale

Should we start experiments again?

Guilds form on their own.



人間性尊重 Respect for people

Making things is making people.


Fremont, California. 2010s.



2010

Same plant. 2018.

“Excessive automation at Tesla was a mistake. Humans are underrated.”

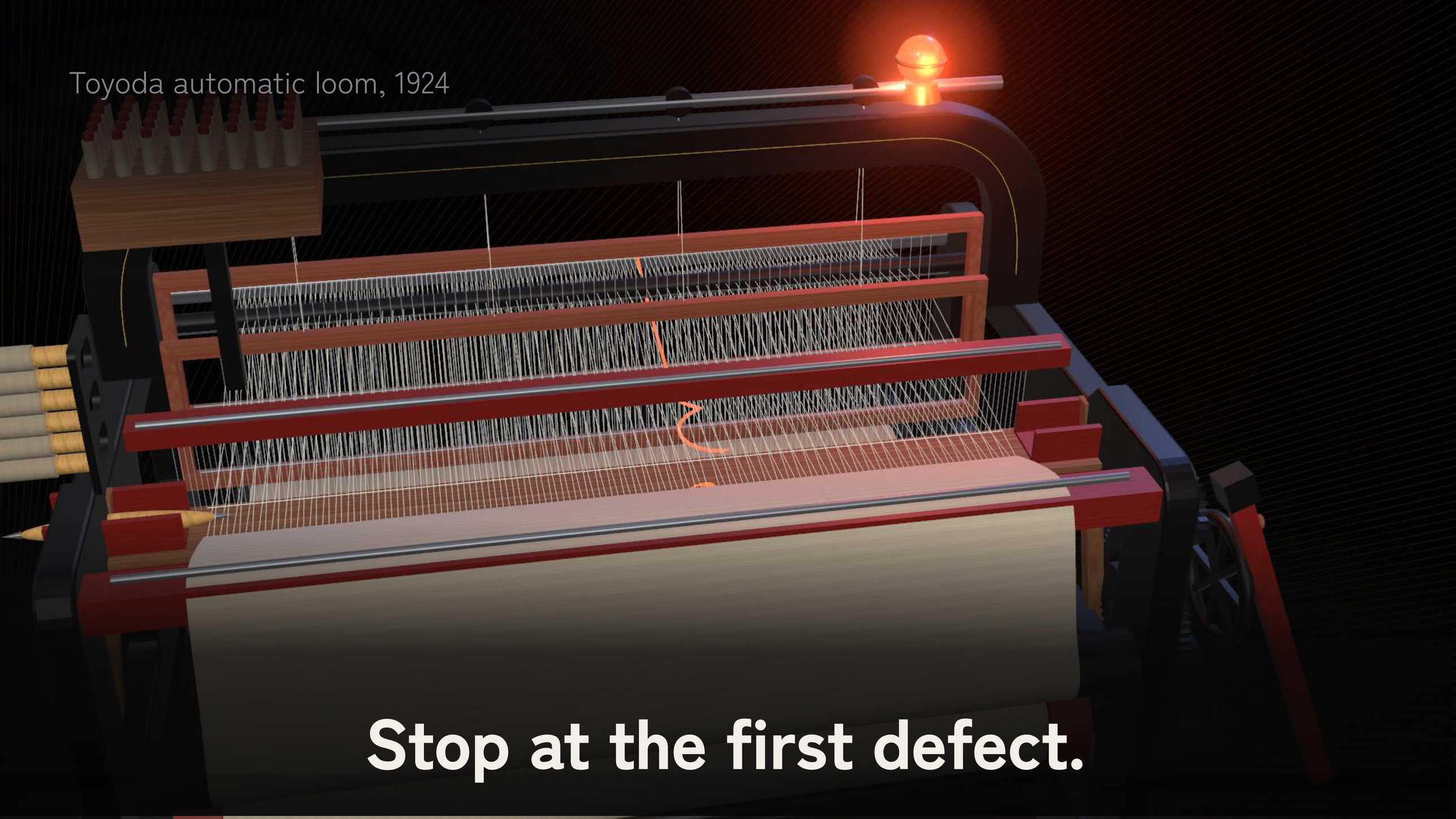
Elon Musk, April 2018 (shortened)


Value for the Customer
Right
First Time
Just
in Time
Building a Learning Organization
Tech-enabled Network of Teams

TOYOTA

Before Toyota made cars, it made looms.

Toyoda Automatic Loom Works, 1926


Toyoda automatic loom, 1924
Stop at the first defect.

Add free shipping over £50.

Done. Applied to the basket subtotal.

No, after discounts.

Updated.

And keep the existing tests passing.

Fixed. All tests pass.

3 prompts to get it right

Right first time = 1.

Agents can find multiple correct ways to do the same thing.

Agent A

```
client.GetItemAsync(key)
```

✓ compiles · ✓ tests pass

Agent B

```
ctx.LoadAsync<Order>(id)
```

✓ compiles · ✓ tests pass

Agent C

```
new OrderStore(db).Find(id)
```

✓ compiles · ✓ tests pass

Agents can find multiple correct ways to do the same thing.

Agent A

```
client.GetItemAsync(key)
```

✓ compiles · ✓ tests pass

Agent B

```
ctx.LoadAsync<Order>(id)
```

✓ compiles · ✓ tests pass

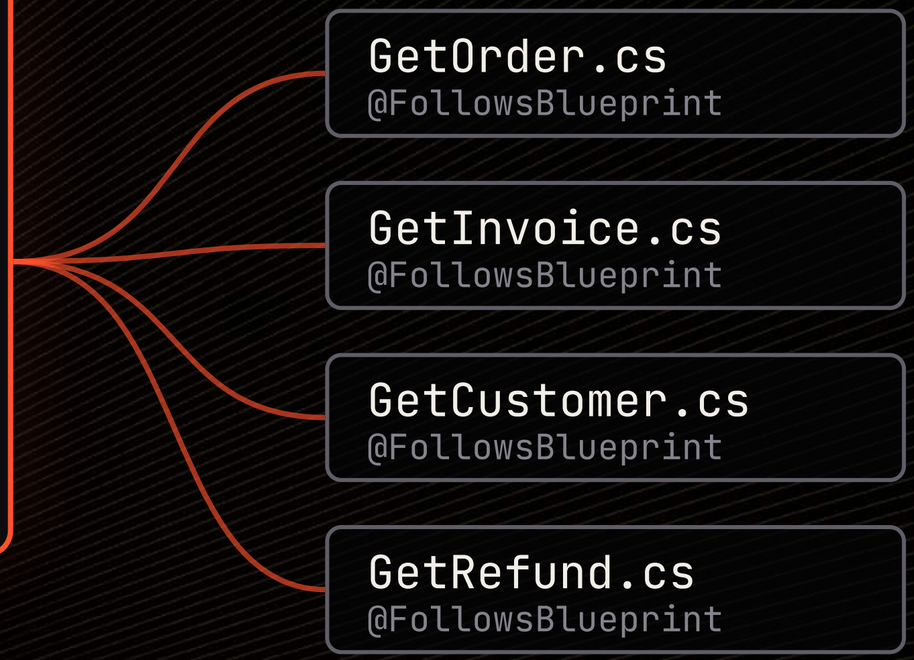
Agent C

```
new OrderStore(db).Find(id)
```

✓ compiles · ✓ tests pass

```
/// @Blueprint ddb-read-by-id
/// use: single-item reads

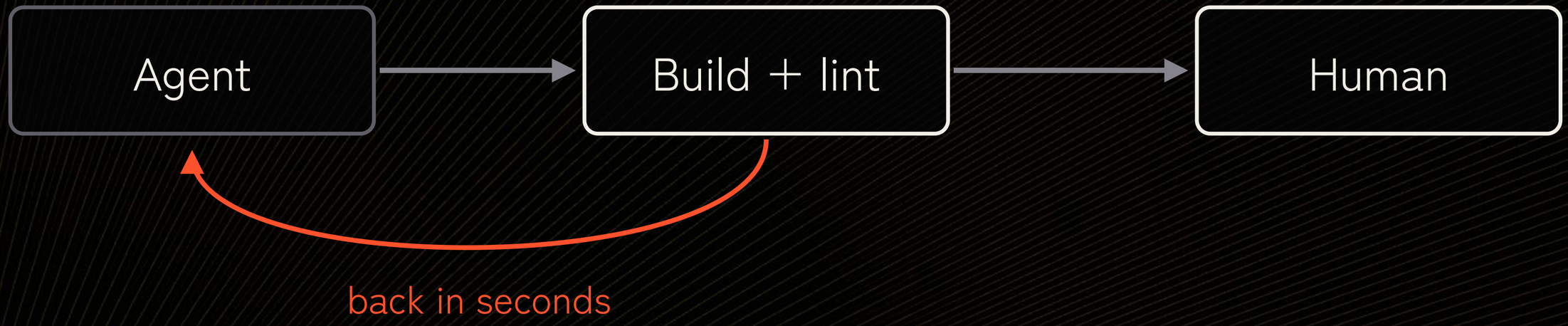
var order = await ctx
    .LoadAsync(id);
```



Show the agent. Don’t tell it.

The canonical example, in the code, with every copy pointing back to it

The build catches it. Not the reviewer.



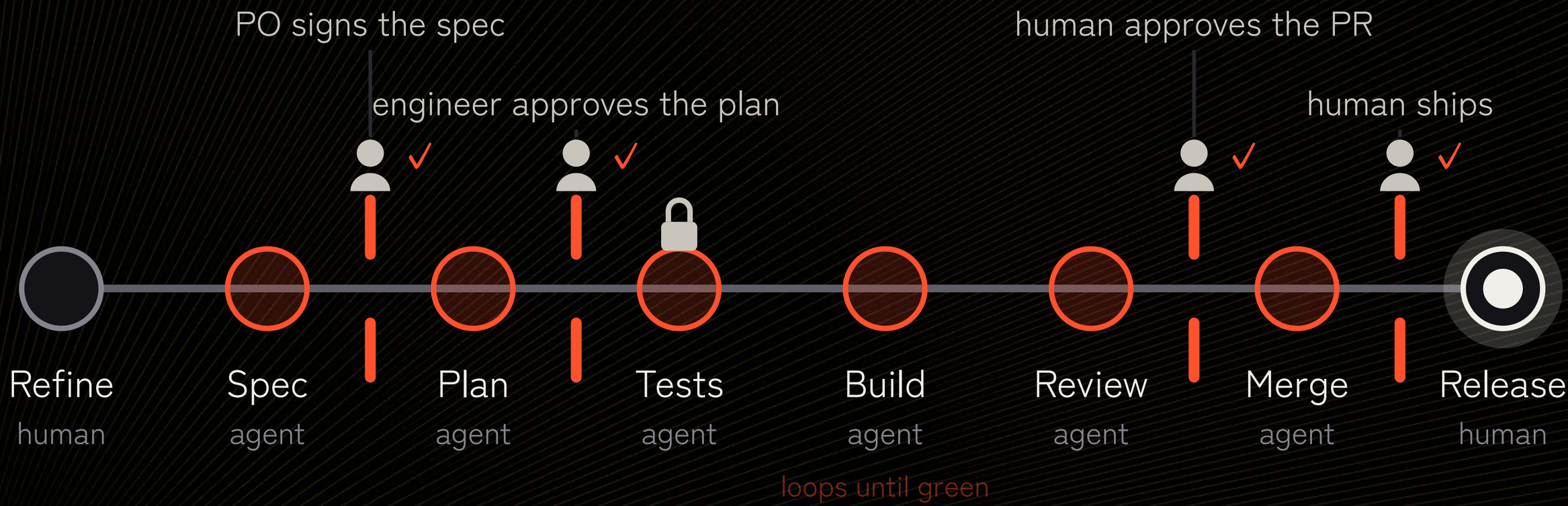

```
$ dotnet build
error TH0101: read bypasses blueprint 'ddb-read-by-id'
  → use DynamoDBContext.LoadAsync<T>
Build succeeded. 0 errors.
```

ポカ　ヨケ

Poka-yoke

Make the mistake impossible.

Autonomy between the gates.



Not thinking through your gates can have serious consequences.



ai-reviewer bot commented on src/api/client.ts

P1 Close the discarded client to avoid a resource leak.

Suggested change

```
    const res = await sharedClient.request(req);
+   sharedClient.close();
```

Commit suggestion



ai-reviewer bot commented on src/api/client.ts

P1 Close the discarded client to avoid a resource leak.

Suggested change

```
    const res = await sharedClient.request(req);
+   sharedClient.close();
```

✓ **Committed** · merged to main

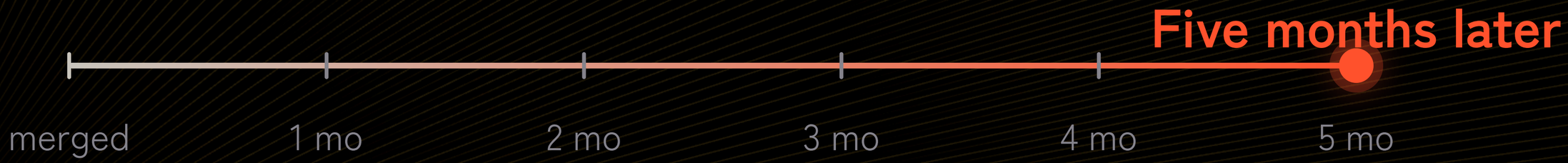
🤖 **ai-reviewer** bot commented on src/api/client.ts

P1 Close the discarded client to avoid a resource leak.

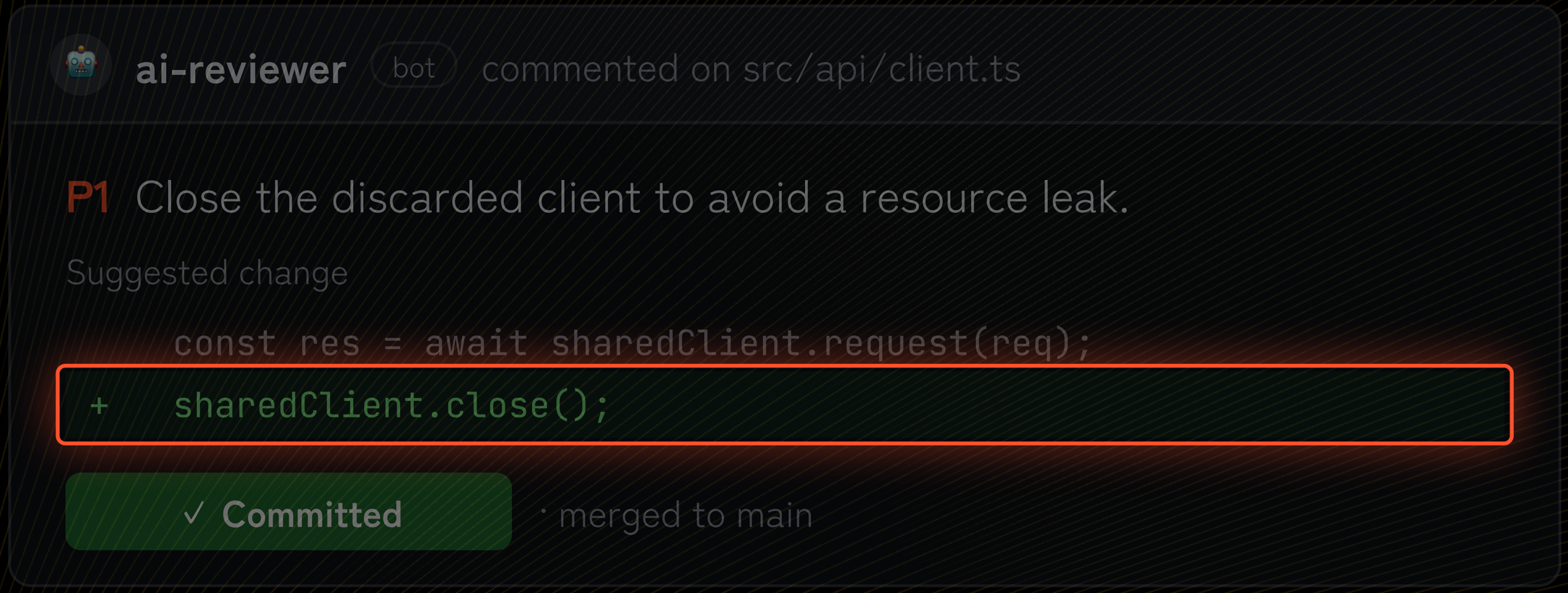
Suggested change

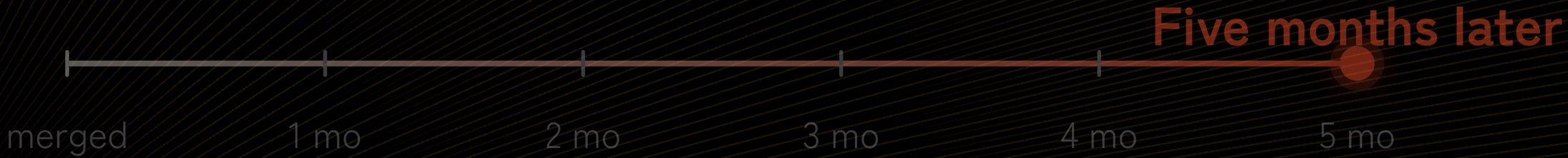
```
    const res = await sharedClient.request(req);
+   sharedClient.close();
```

✓ **Committed** · merged to main



Concurrent requests failing in production.



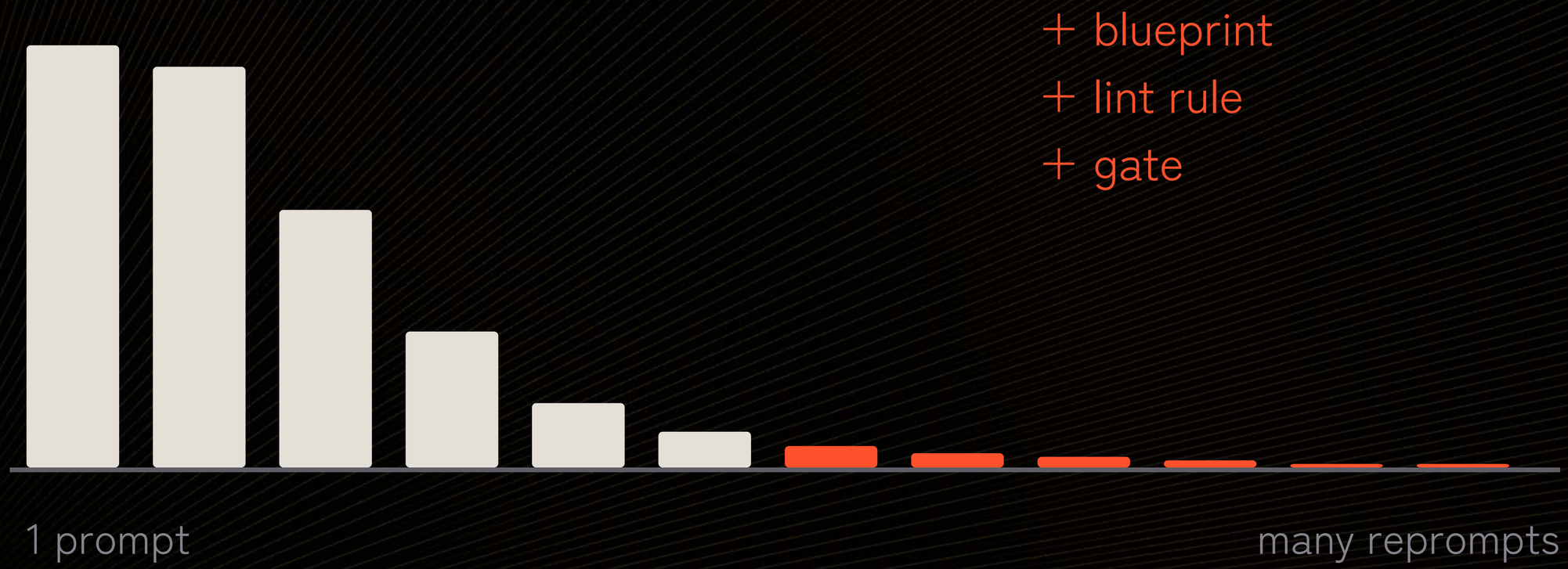


Concurrent requests failing in production.

The fix was the bug.

Reprompts per ticket

from the AI gateway, every session tagged with its ticket




Value for the Customer
Right
First Time
Just
in Time
Building a Learning Organization
Tech-enabled Network of Teams

Where does the time go?

Scrum: one two-week sprint



Refinement

Sprint planning

Development

Code review

QA

Demo

Retro

Rearrange the stations.

Build got cheap: one week, rebalanced

retro → becomes kaizen



Refine Plan Build Review Verify Demo

Rearrange the stations.

Build got cheap: one week, rebalanced



Add new ones.

Mon

Product, design and tech in one room: refine the epic, slice the week



Tue

Two competing plans, merged, then build



Wed

Two competing plans, merged, then build



Thu

Two competing plans, merged, then build



Fri

Kaizen from an evidence brief



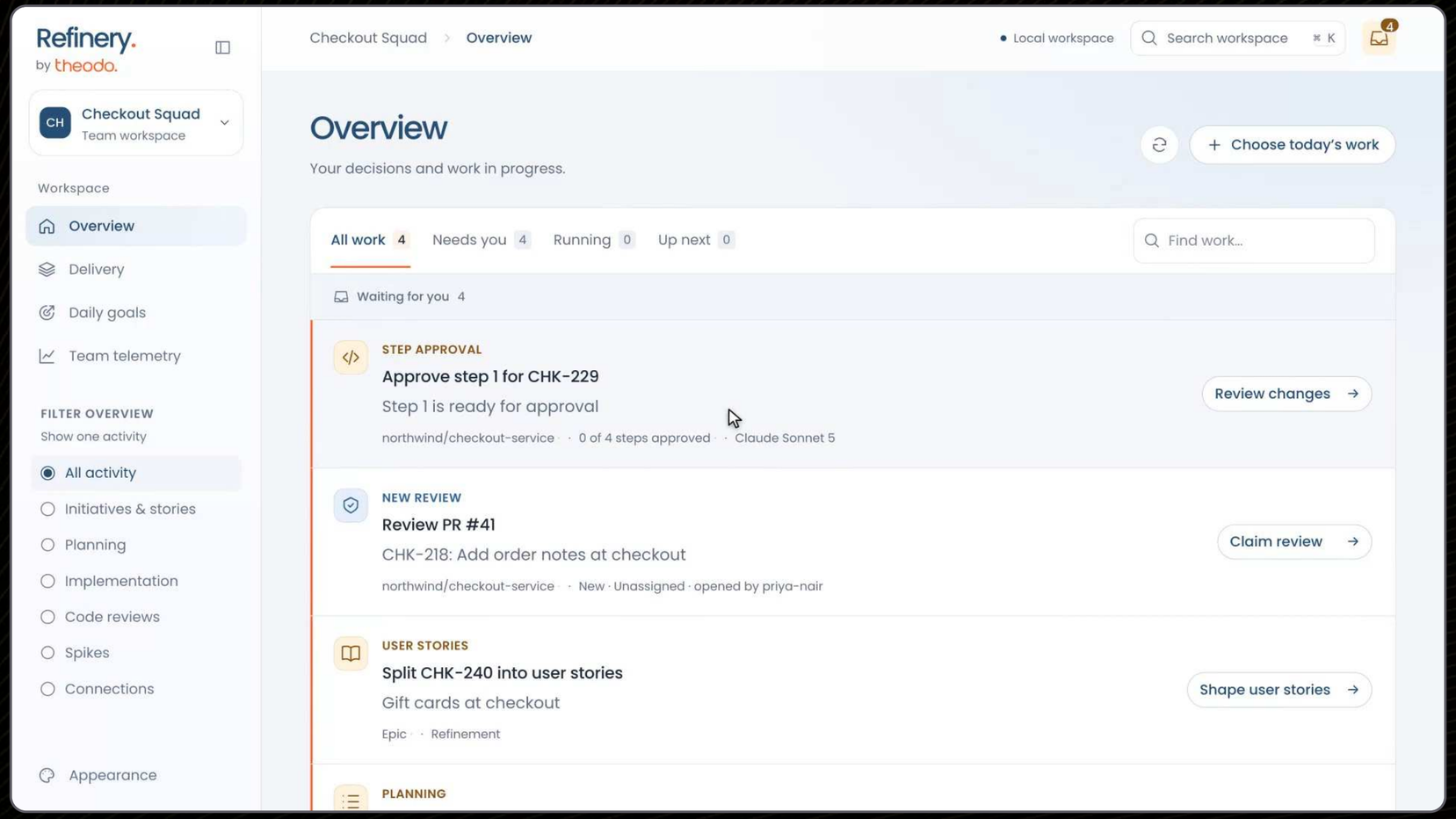
👨‍💼 product owner · 👨‍🎨 designer · 👨‍💻 engineer · 🤖 agent

So we built the tools around the process...

So we built the tools around the process...

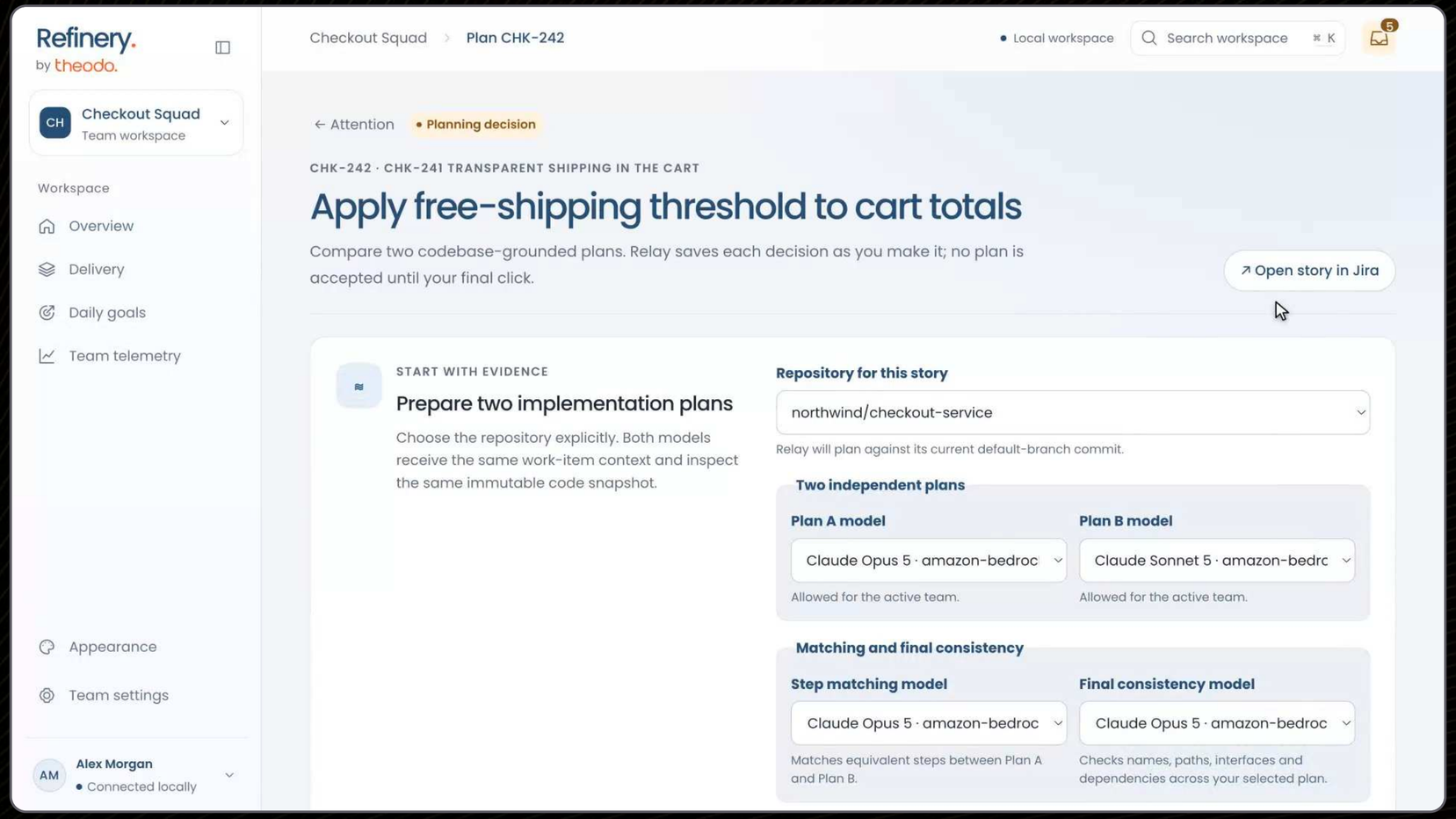
...not the process around the tools.

One connected platform, for product, design and engineering.



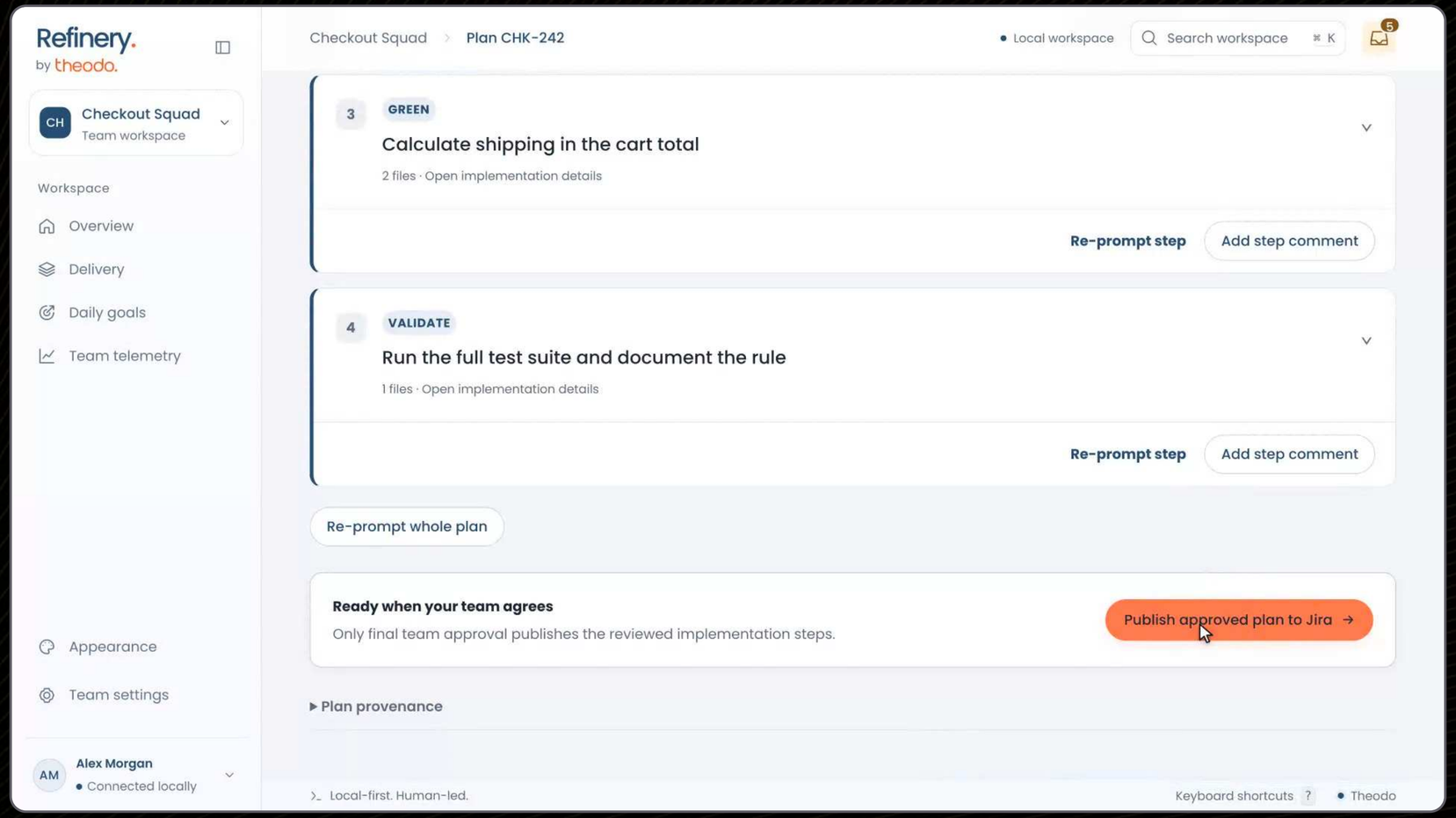
Connected Plan Build Gates Telemetry

The engineer shapes the plan, and stays close to the code.



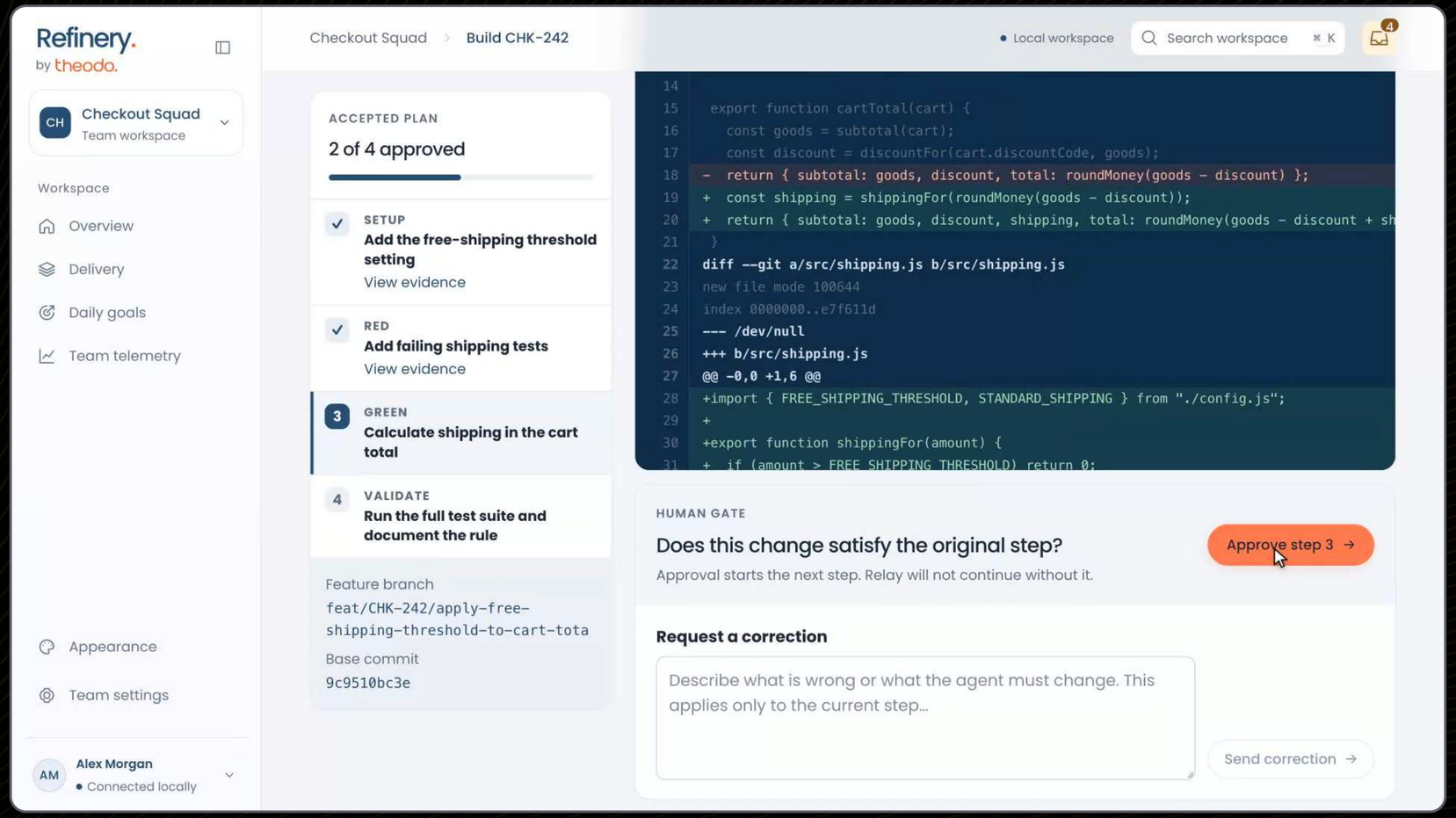
Connected Plan Build Gates Telemetry

Built test-first, one approved step at a time.



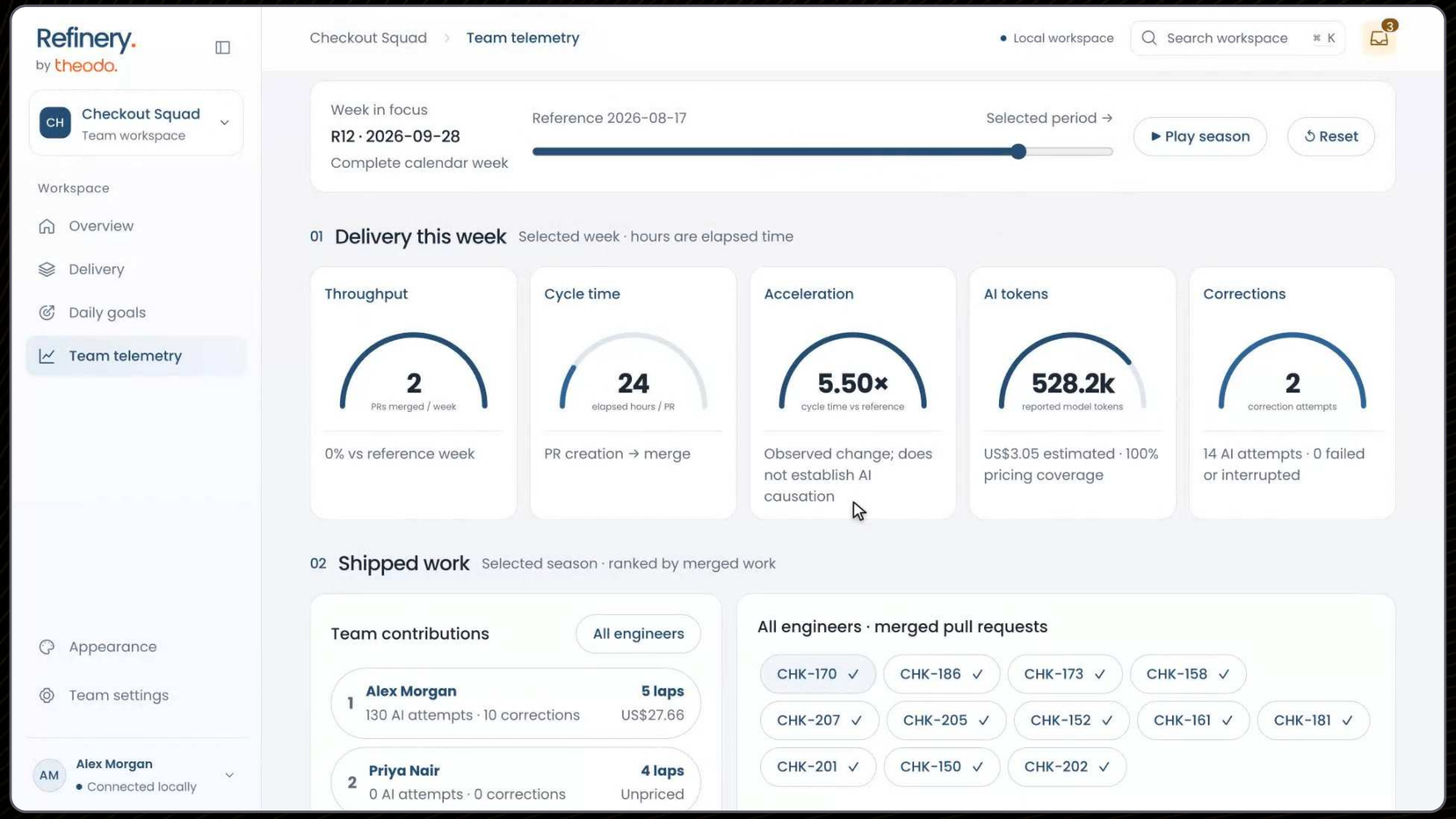
Connected Plan Build Gates Telemetry

Humans in the loop, where regulation requires it.

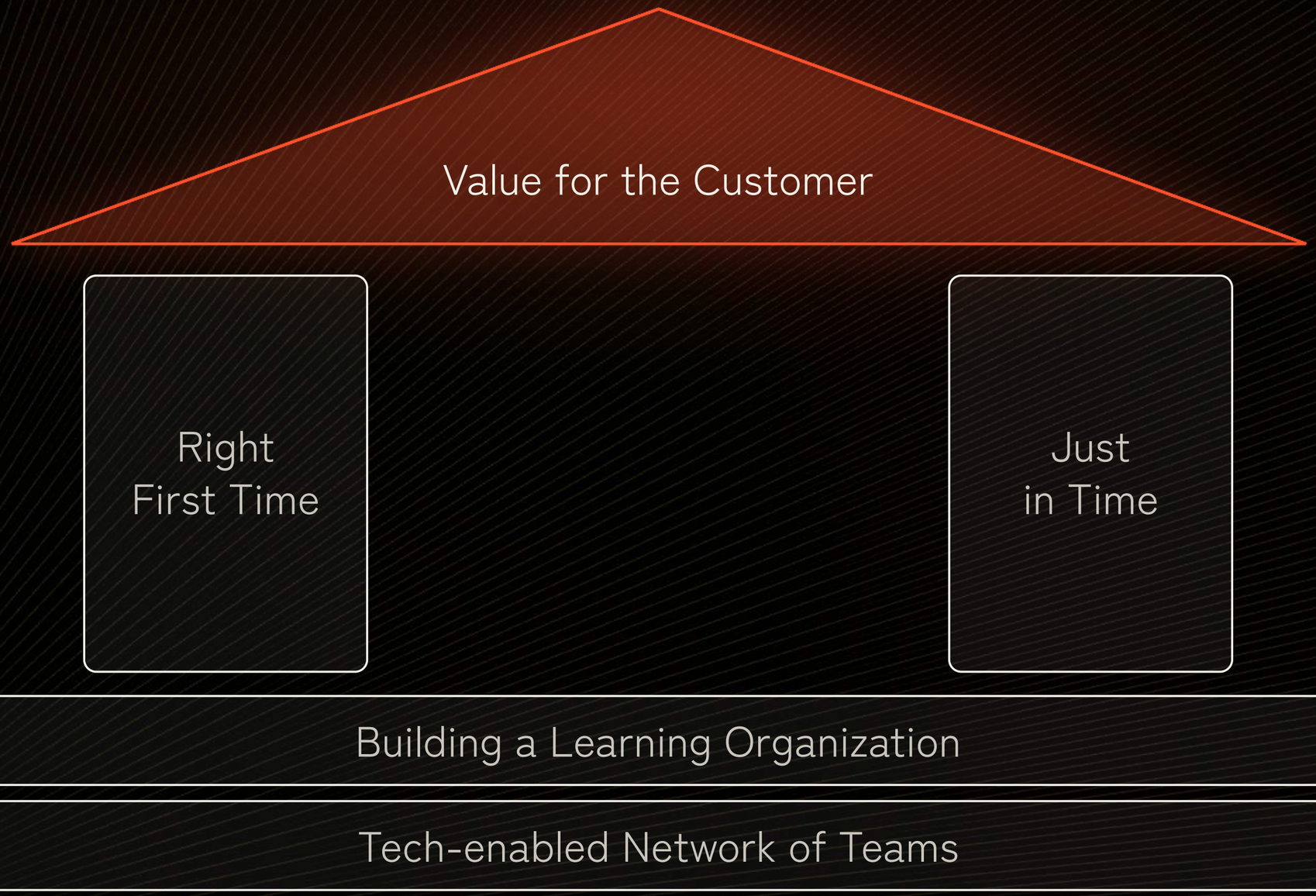


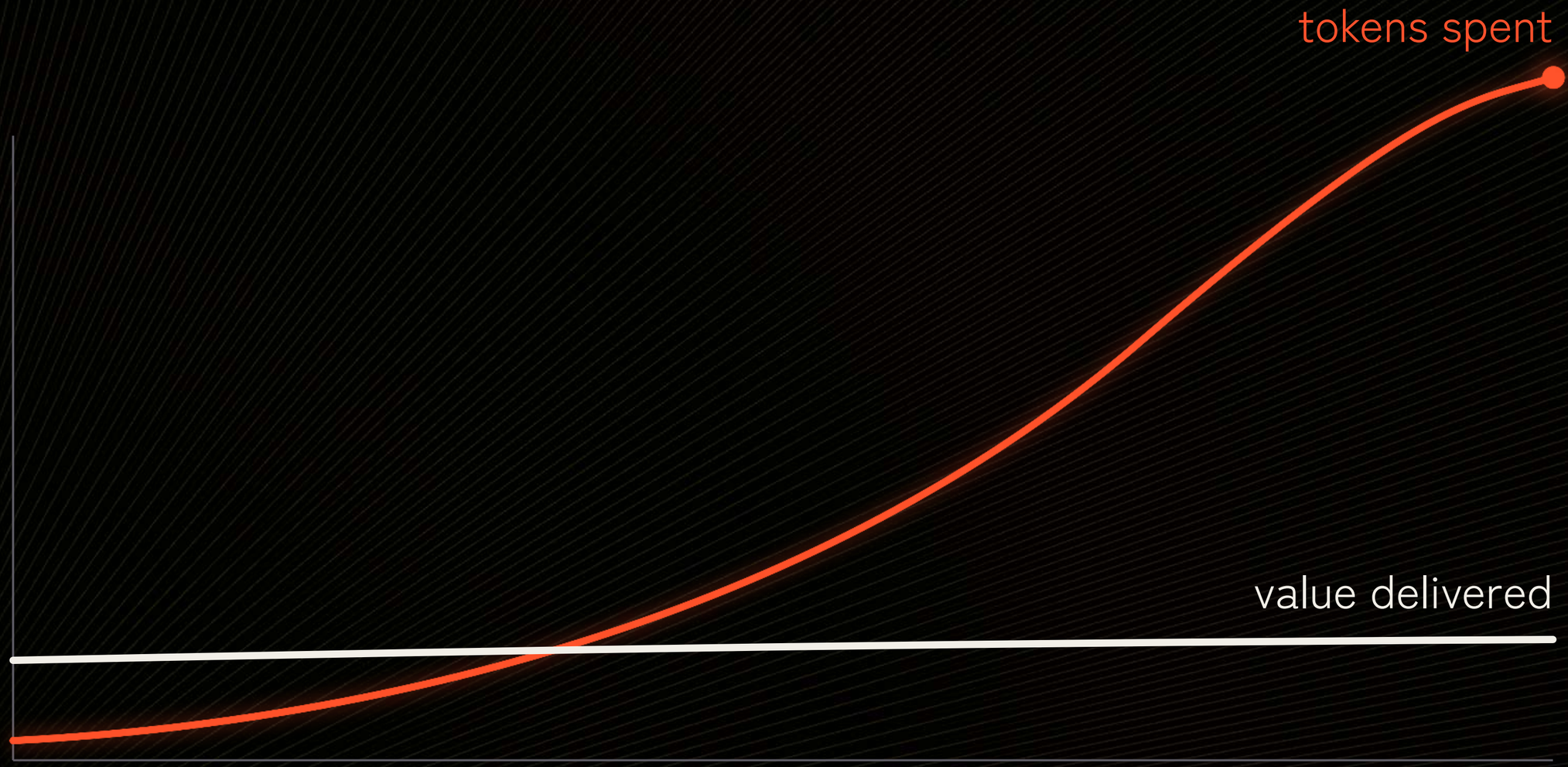
Connected Plan Build Gates Telemetry

Every run measured, per ticket and per engineer.



Connected Plan Build Gates Telemetry


Value for the Customer
Right
First Time
Just
in Time
Building a Learning Organization
Tech-enabled Network of Teams



Token spend isn’t value.

Uber

The Verge · 26 May 2026

Uber president says AI spending is getting ‘harder to justify’

“That link is not there yet.” Annual AI budget reportedly gone in four months.

What we measure instead.



Flow (DORA)

Lead time ·
deploy frequency

What we measure instead.

Flow (DORA)

Lead time · deploy frequency

Quality (DORA)

Change failure rate · time to recover

What we measure instead.

Flow (DORA)

Lead time · deploy frequency

Quality (DORA)

Change failure rate · time to recover

Right first time

Reprompts per ticket

What we measure instead.

Flow (DORA)

Lead time · deploy frequency

Quality (DORA)

Change failure rate · time to recover

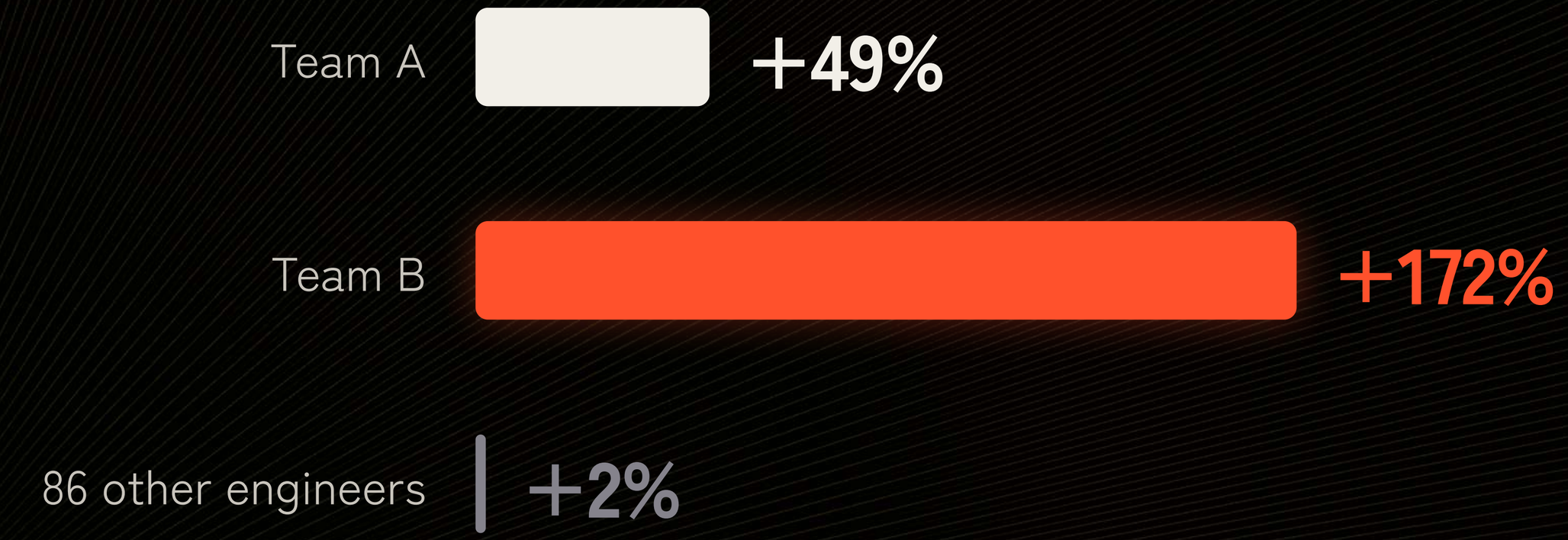
Right first time

Reprompts per ticket

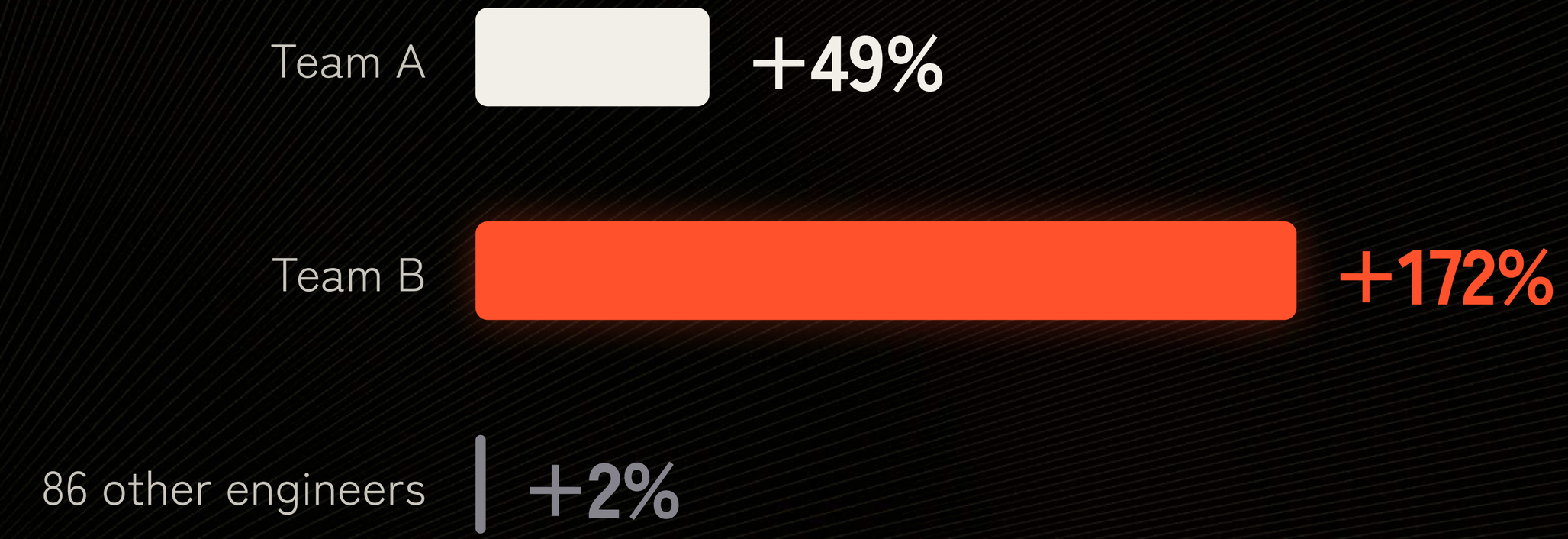
Rework

Fix and rework PRs

Merged output per developer, three weeks, same codebases



Merged output per developer, three weeks, same codebases



And on one team, fix and rework PRs halved.

Build the right thing.
Build it right.
Speed follows.

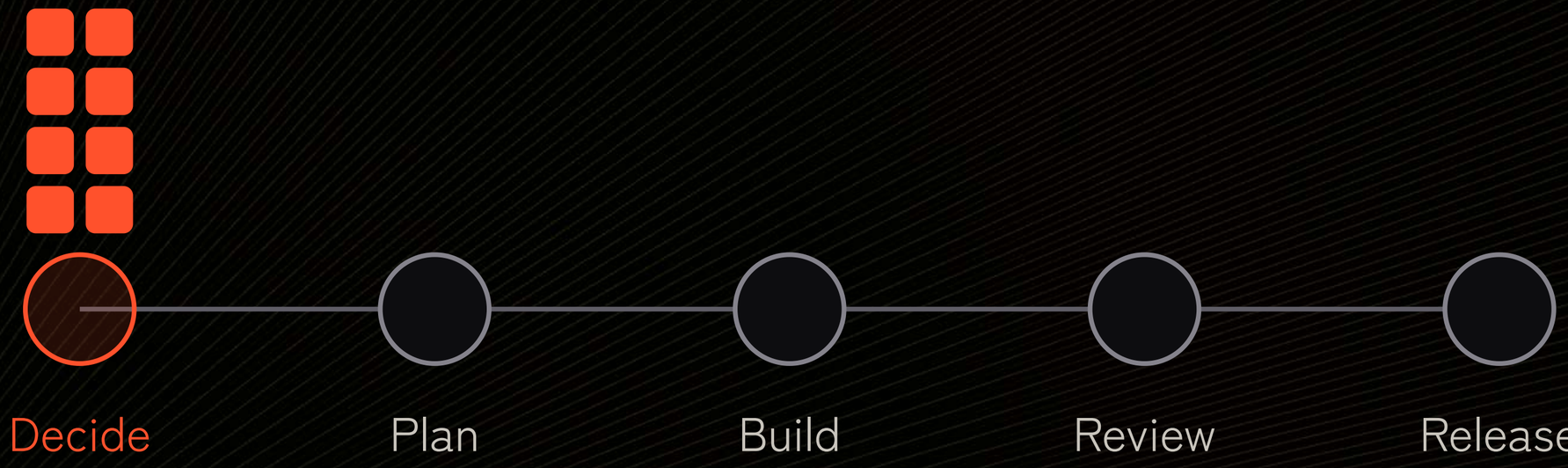
Build the right thing.

- ✓ Build it right.
- ✓ Speed follows.

Build the right thing.

✓ Build it right.

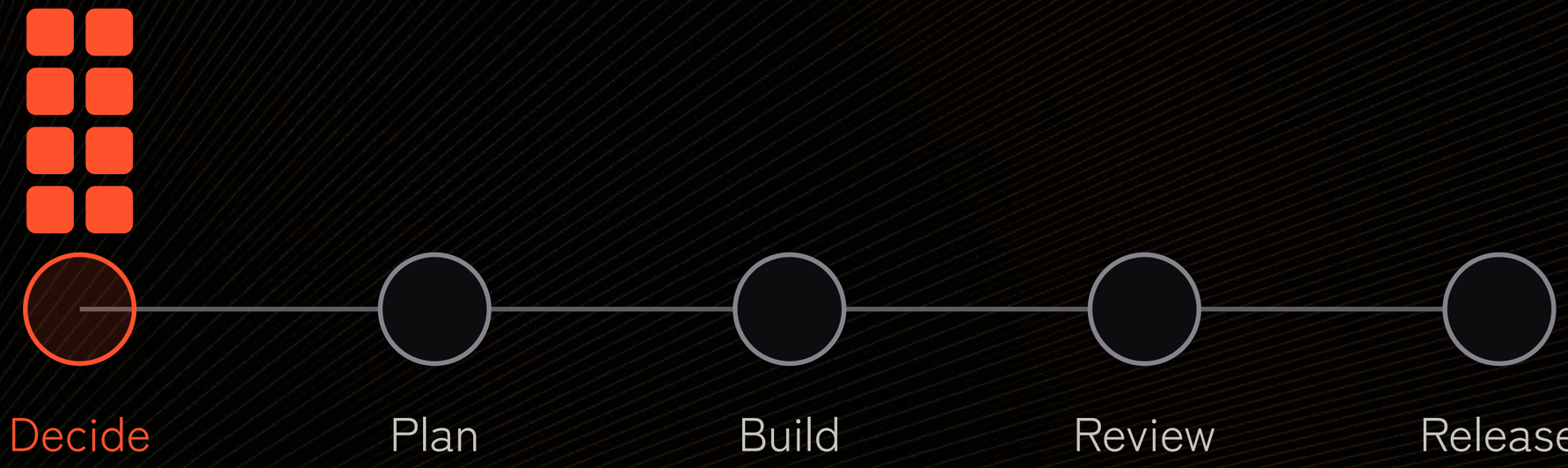
✓ Speed follows.



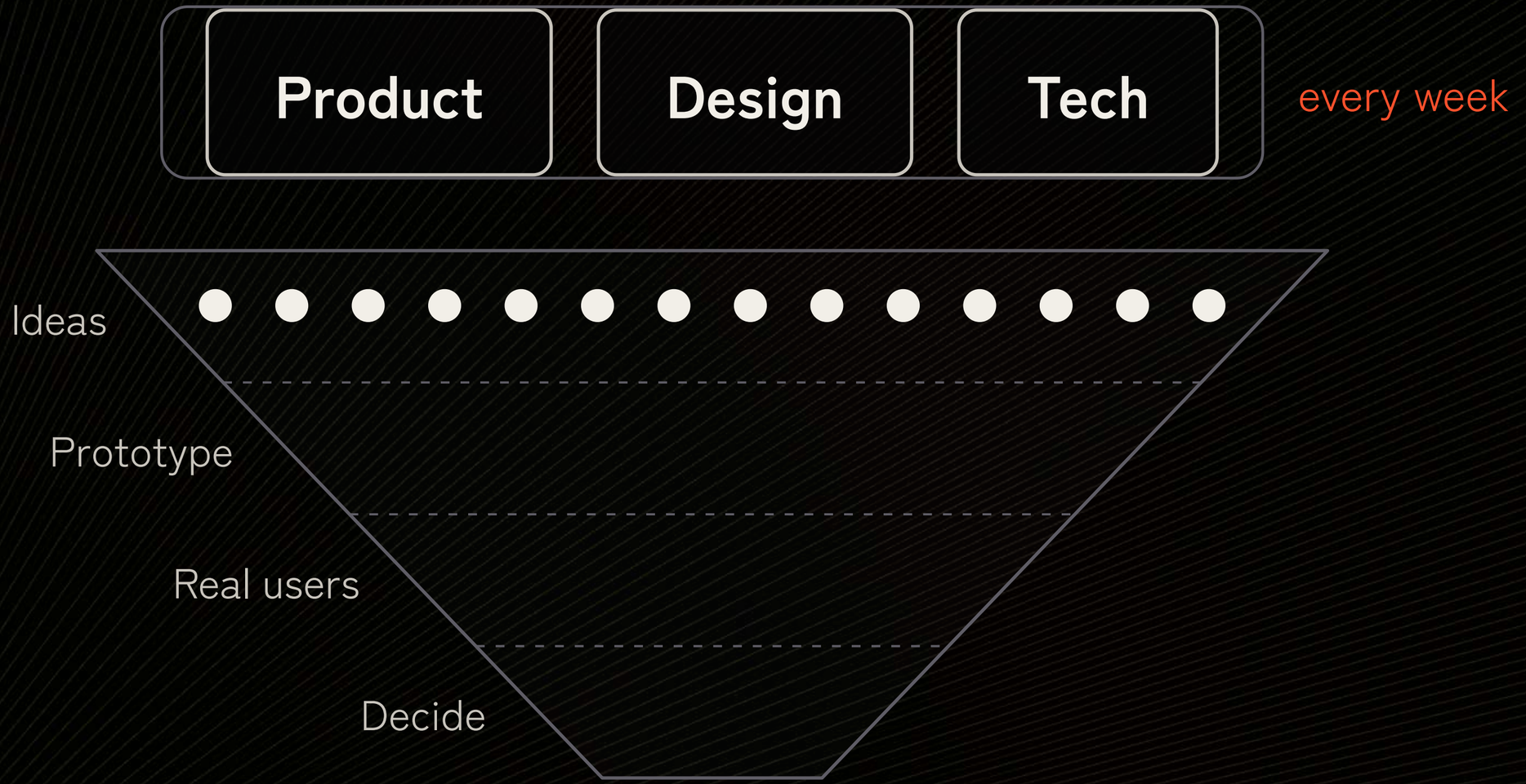
Build the right thing.

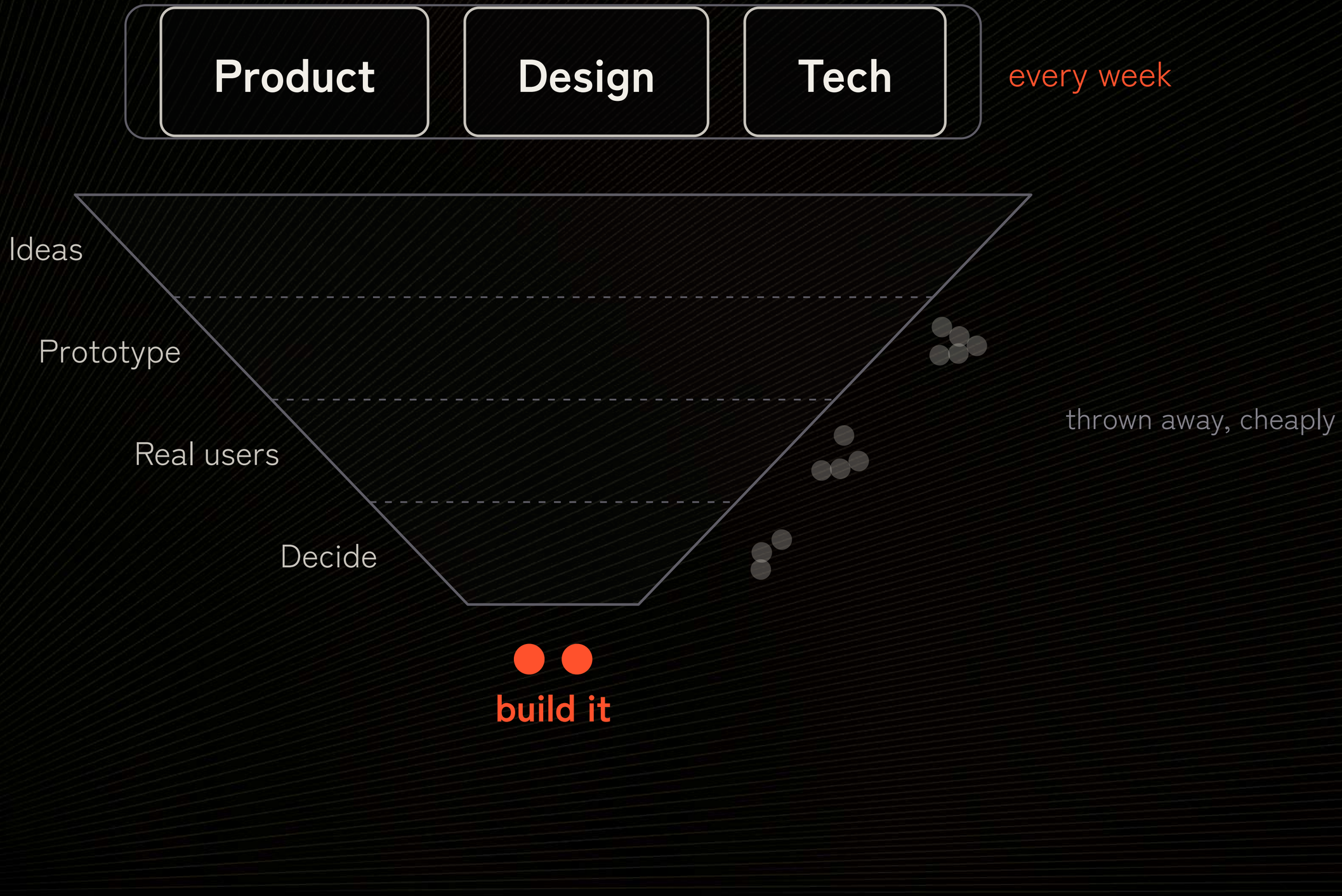
✓ Build it right.

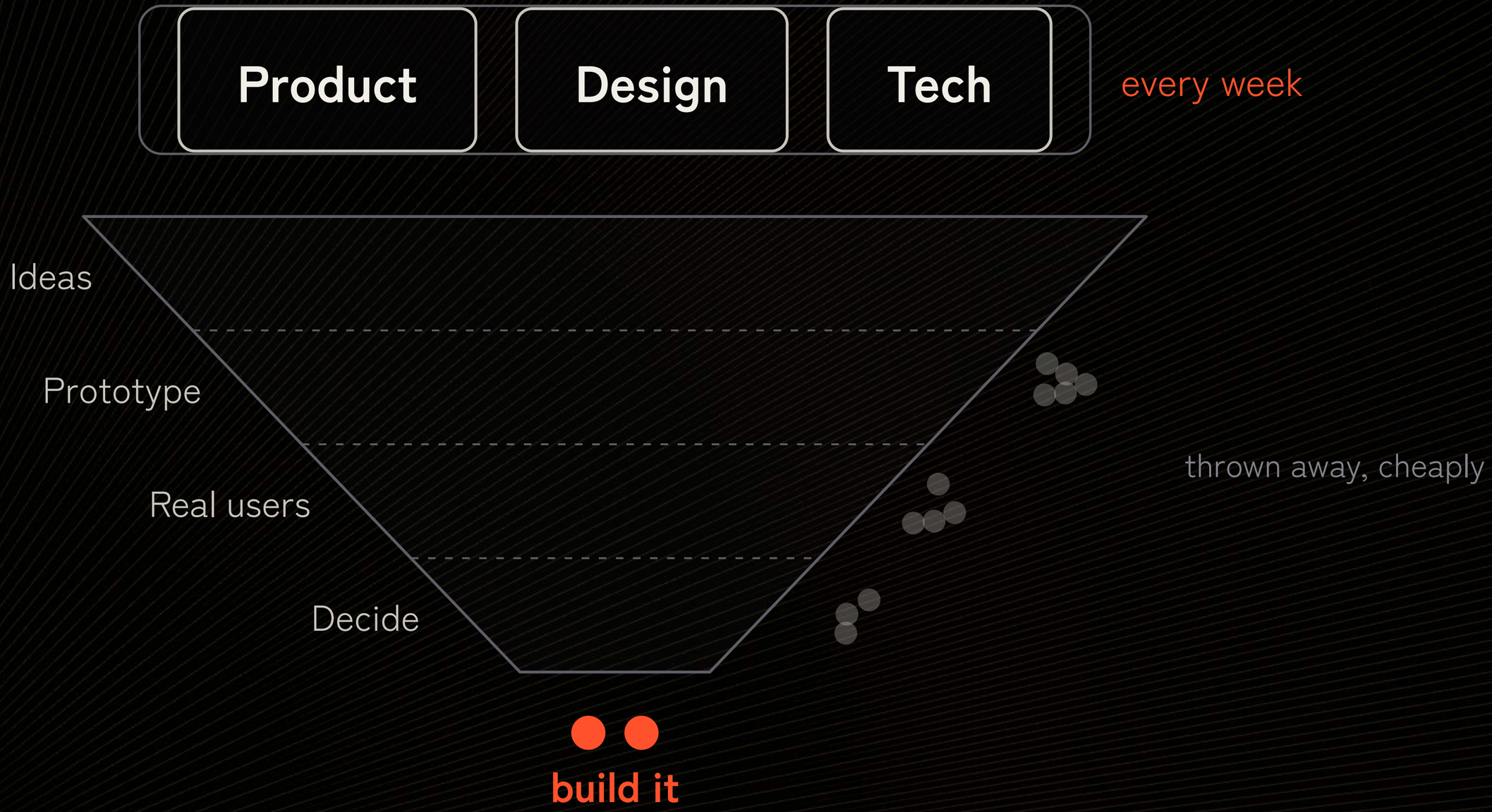
✓ Speed follows.



With speed solved, value for the customer is the work.


Product
Design
Tech
every week
Ideas
Prototype
Real users
Decide


Product
Design
Tech
every week
Ideas
Prototype
Real users
Decide
thrown away, cheaply
build it



Building the right thing is the job now.

Same platform. Different cars.

C-HR



same platform underneath

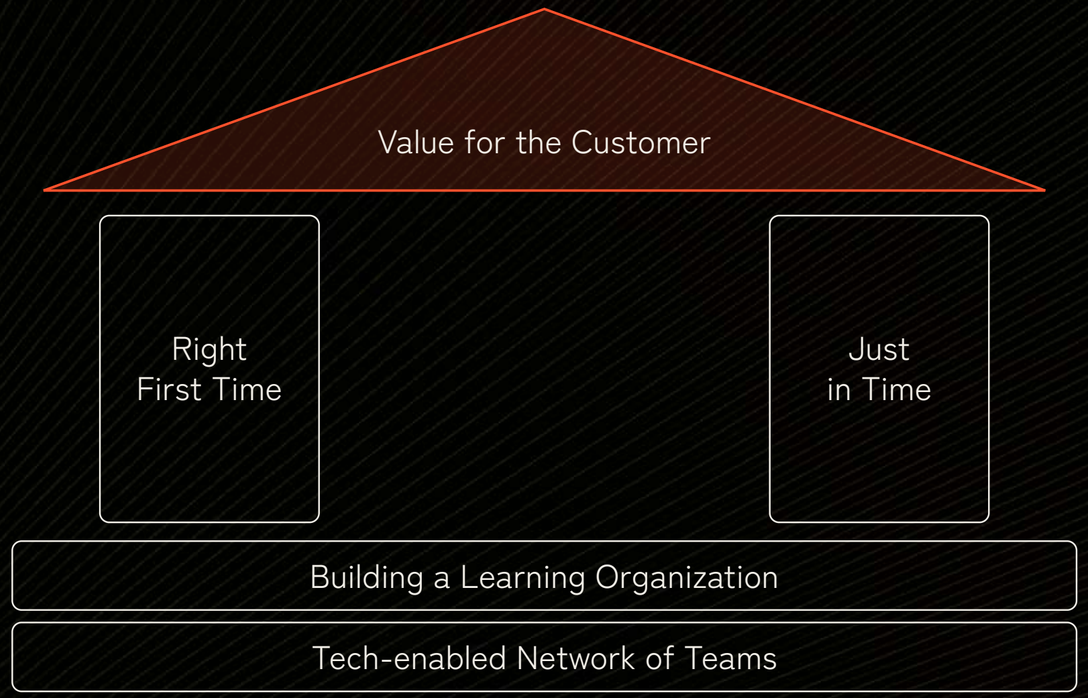
Same platform. Different cars.

C-HR



same platform underneath

One line. A product for every user.



So let’s wrap up...

Audit your production line.

idea | spec | plan | build | verify | review | release | outcome



Focus on the right metrics.

Focus on the right metrics.

Flow.

Focus on the right metrics.

Flow. Outcomes.

Focus on the right metrics.

Flow. Outcomes. Not seats.

Focus on the right metrics.

Flow. Outcomes. Not seats.

We don’t measure AI by the code it writes,
but by the bottlenecks it clears.

Make sure this isn’t your 1978.



Thank you.

Connect on LinkedIn