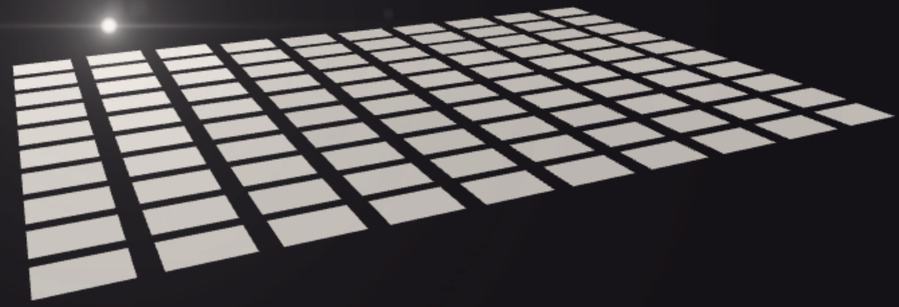
LeadDev Berlin 2026

CodeRabbit

After the Agents Write the Code

A Field Report From 100 Million Code Reviews

Every square: 1 million code reviews



David Loker
VP of Applied AI, CodeRabbit



We judge coding agents by
how much they can do on their own.

We judge coding agents by
how much they can do on their own.

The number that matters is how much of their work your company can trust.

Today’s questions

What does it take to move from human-centered to fully agentic software development?



Today's questions

What does it take to move from human-centered to fully agentic software development?

And what does it take to ship dramatically faster without sacrificing quality or security?



Today's questions

What does it take to move from human-centered to fully agentic software development?

And what does it take to ship dramatically faster without sacrificing quality or security?

01
What changed

02
What we saw from the outside

03
What we lived on the inside

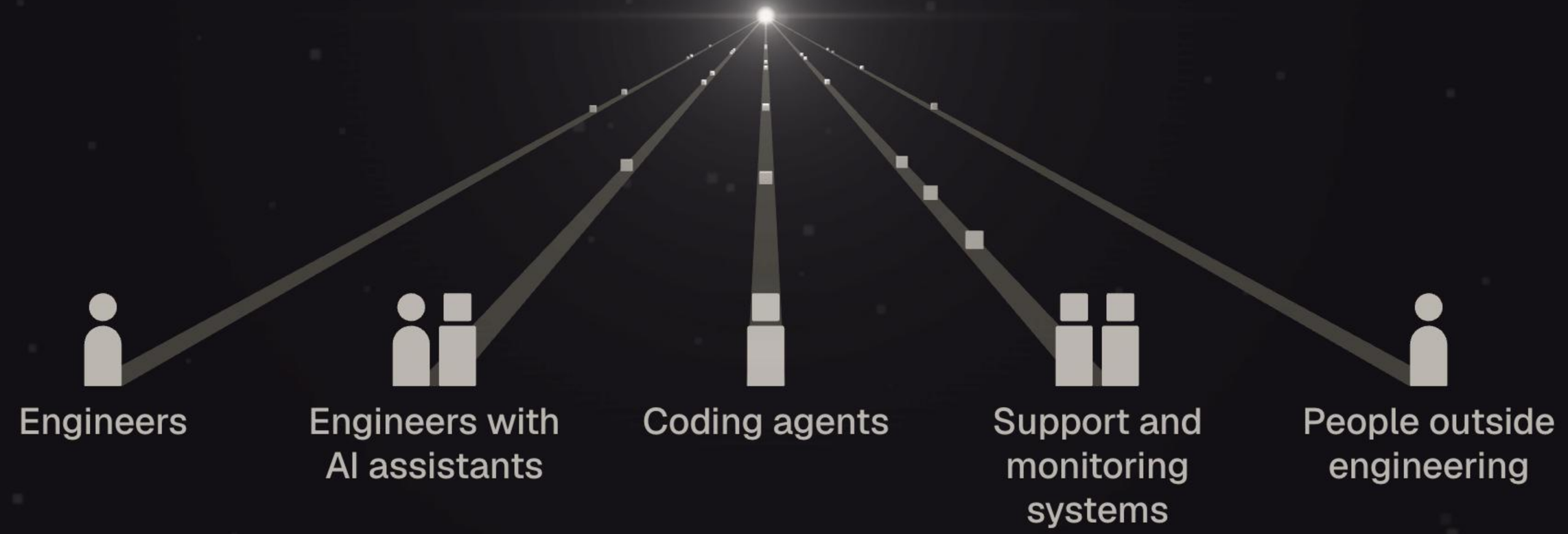
04
What it takes

What changed



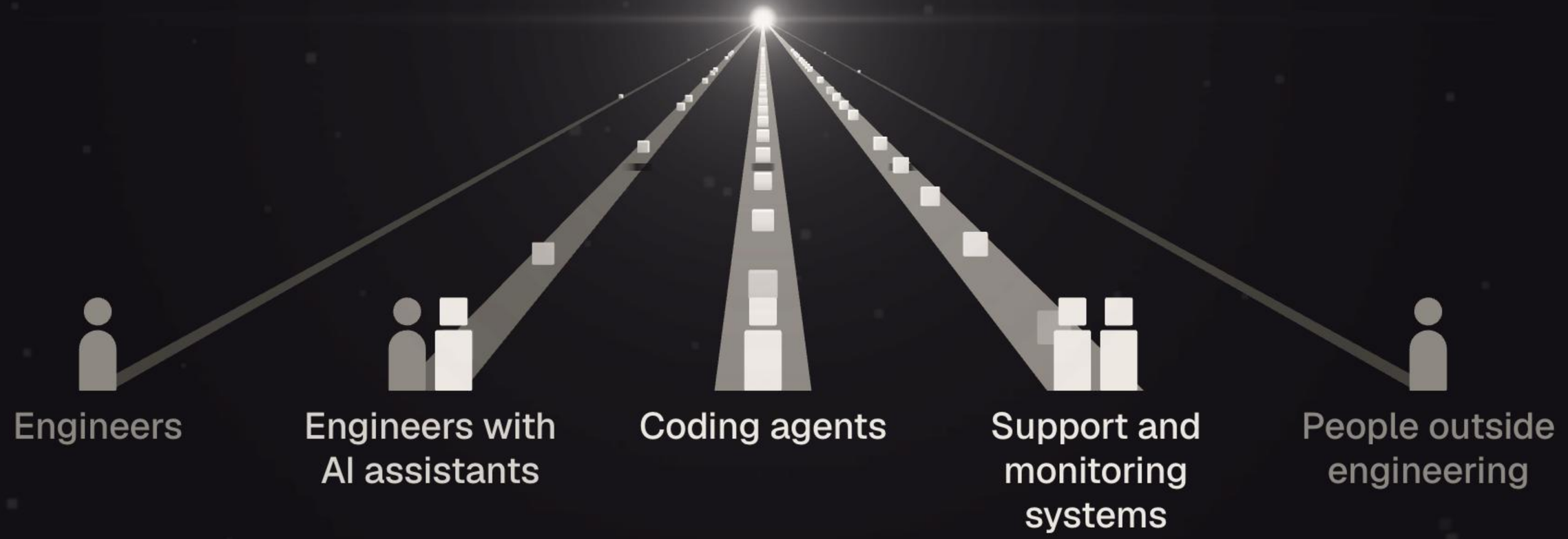
01

Who writes the changes we review?



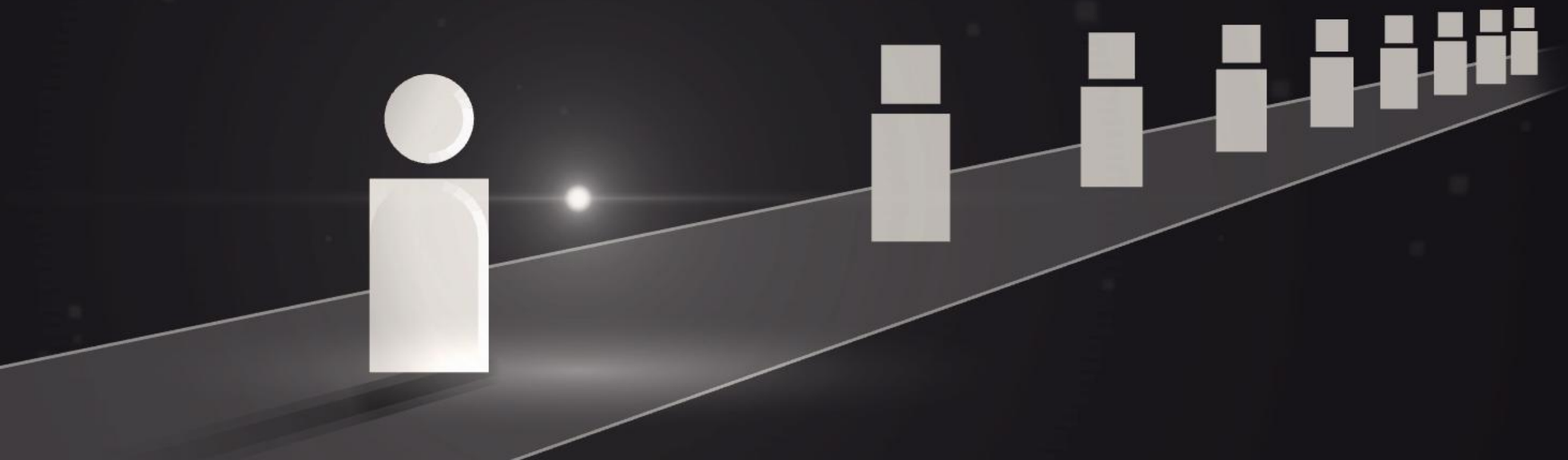
Who writes the changes we review?

More and more often, the author is an agent.

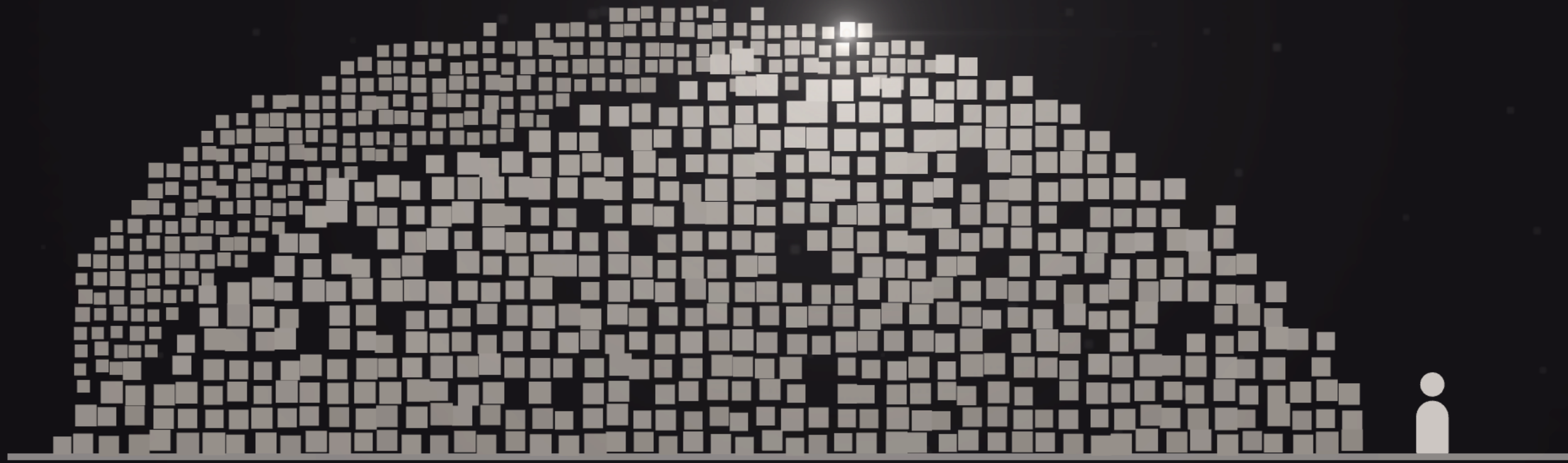


The opportunity right now

What a team can build no longer has to depend on hiring.

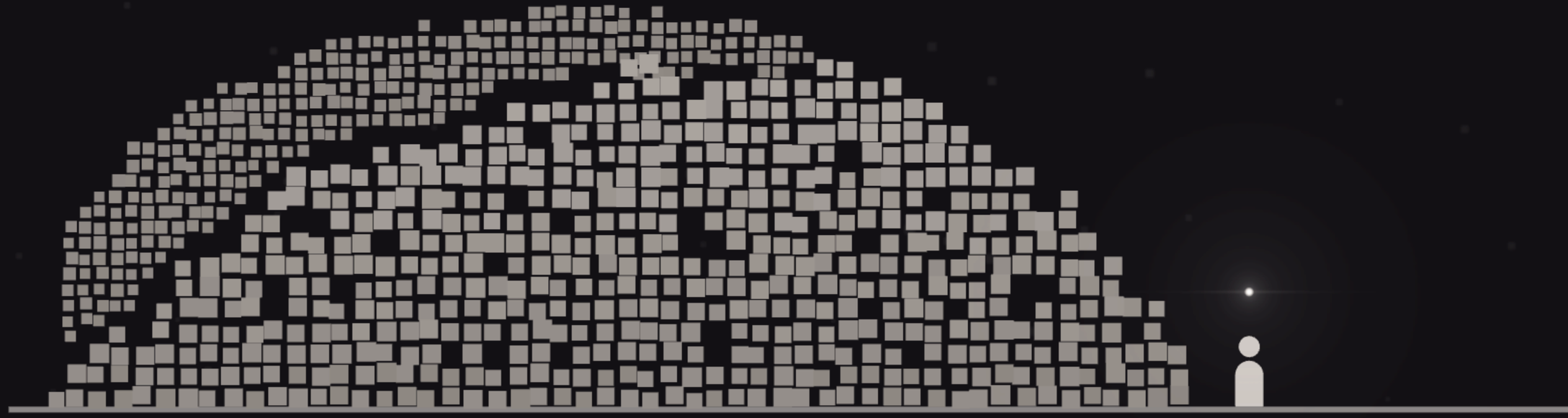


AI made software creation abundant.

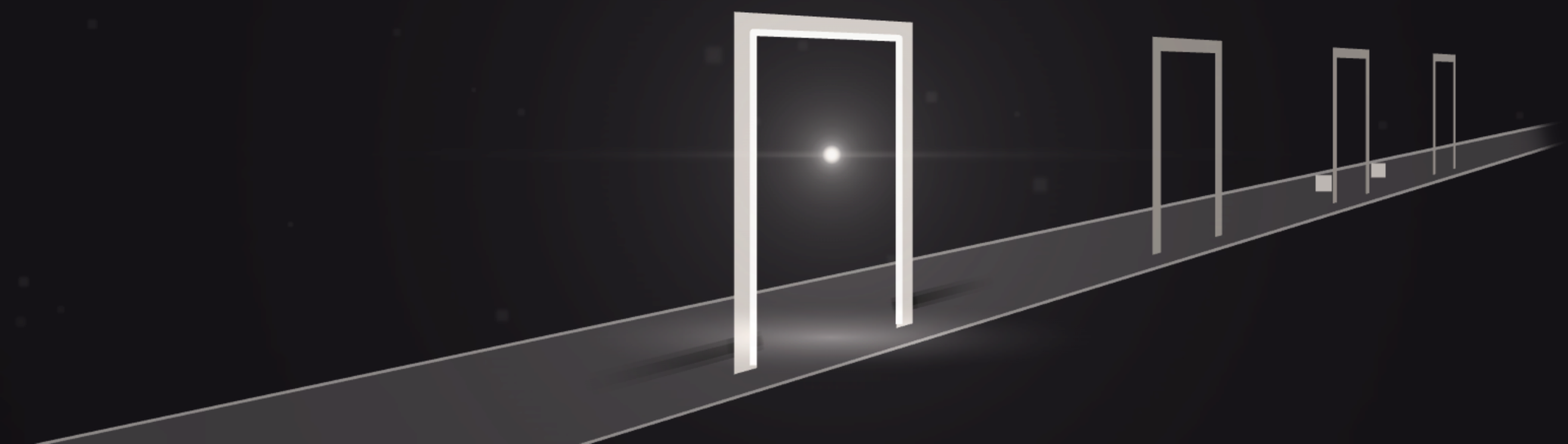


AI made software creation abundant.

But it broke how software is trusted.

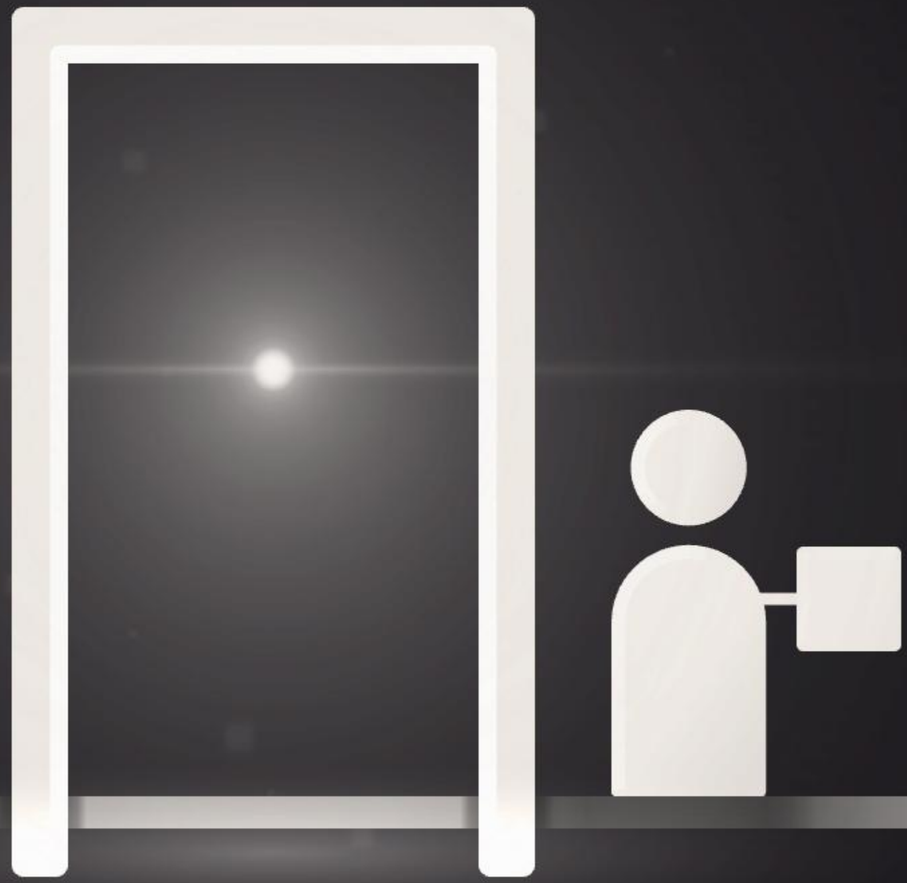


Most of what kept our software trustworthy was a side effect of how expensive code used to be.



When code was expensive,
we got four things for free.

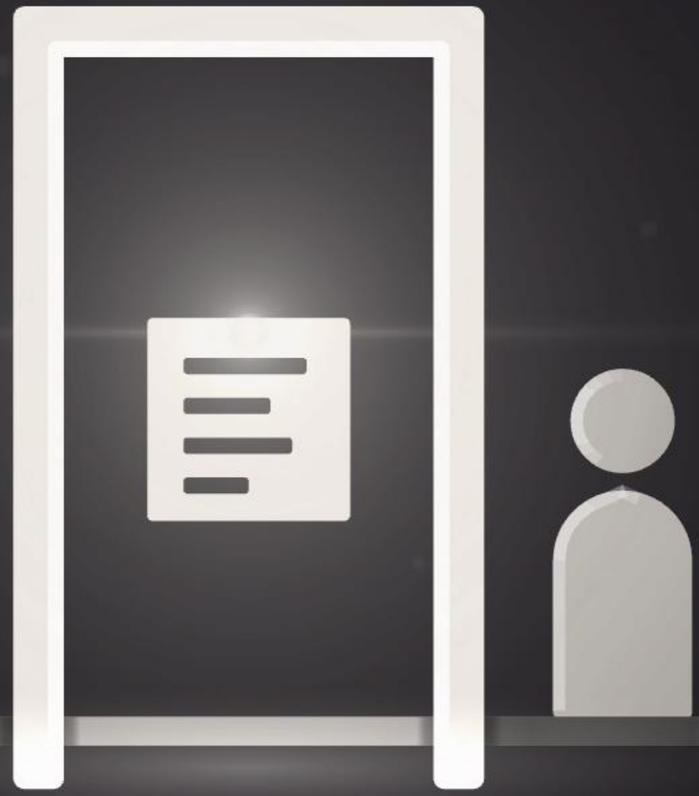
We knew who wrote
every change.



When code was expensive,
we got four things for free.

We knew who wrote every change.

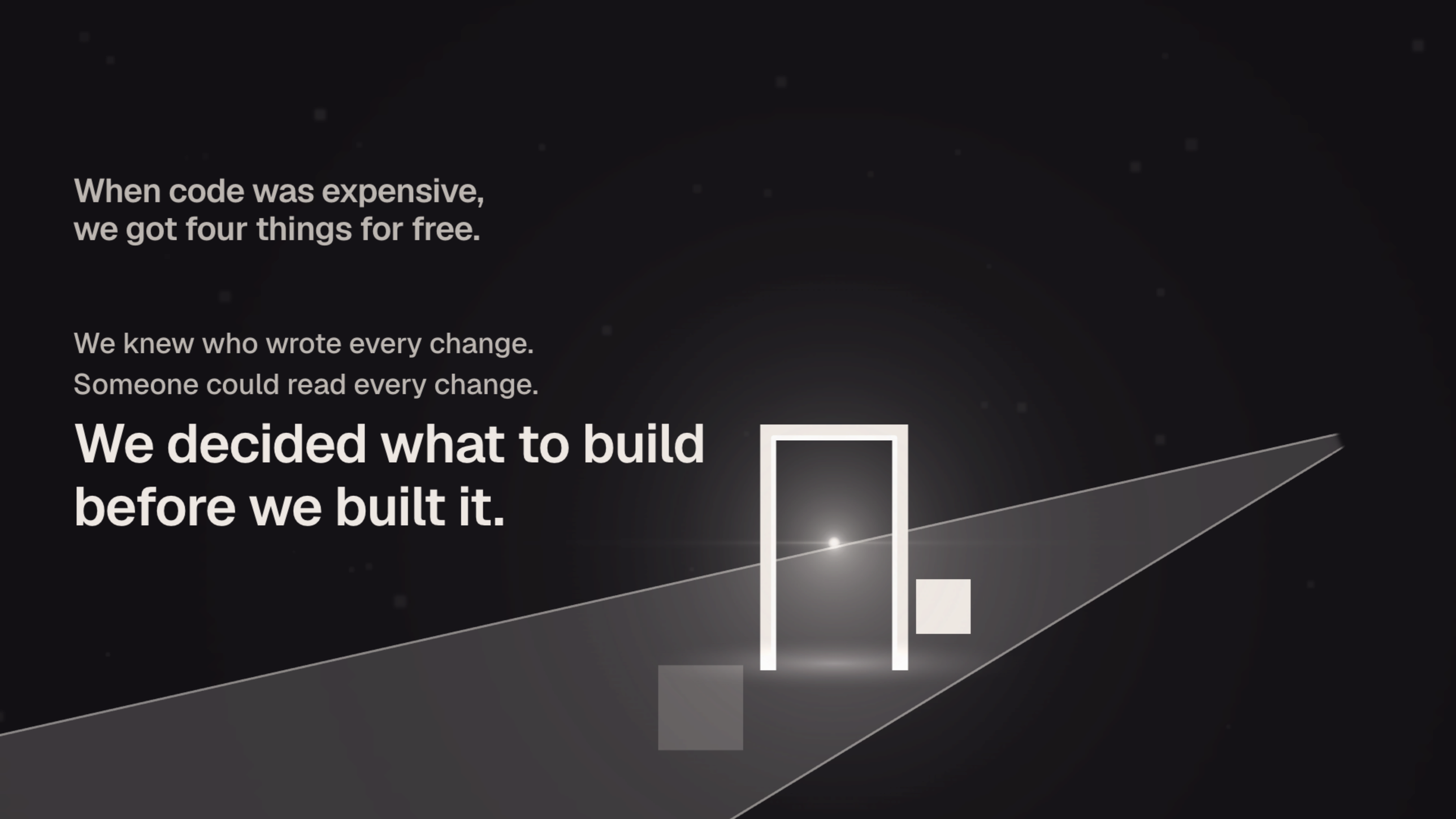
Someone could read every change.



When code was expensive,
we got four things for free.

We knew who wrote every change.
Someone could read every change.

We decided what to build before we built it.



When code was expensive,
we got four things for free.

We knew who wrote every change.
Someone could read every change.
We decided what to build before we built it.

Code changed slowly, so what we checked stayed safe.



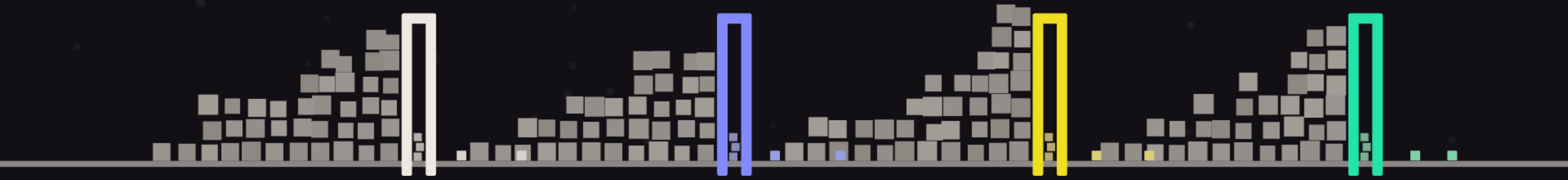
Cheap code took all four away at once.

We knew who wrote every change.

Someone could read every change.

We decided what to build before we built it.

Code changed slowly, so what we checked stayed safe.



Review

Understanding

Prioritization

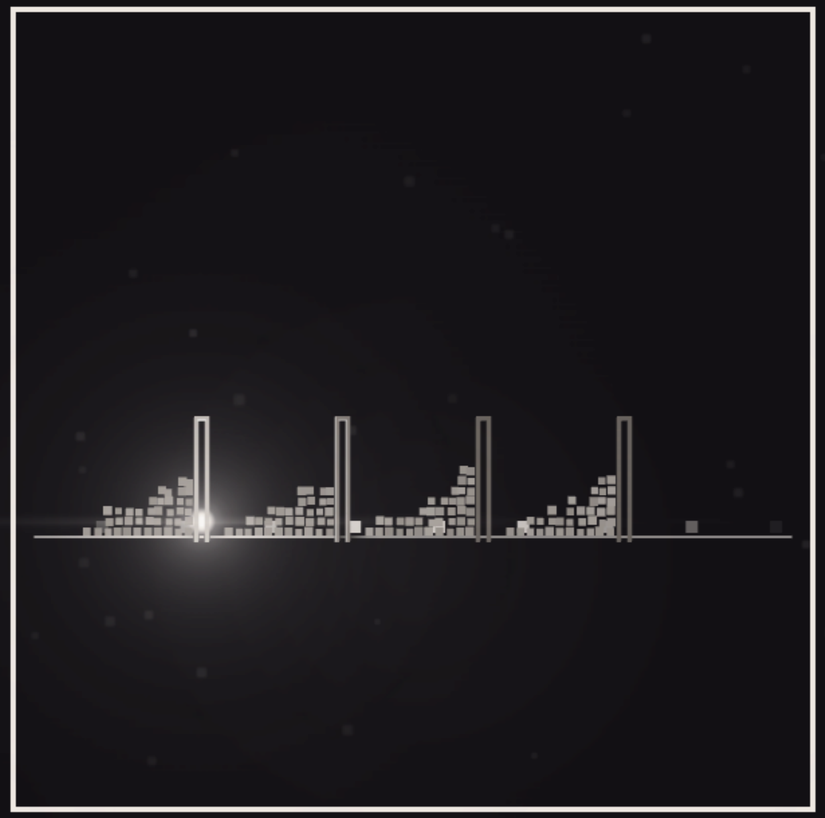
Security

What we saw from the outside

02



One company sees its own agents.



One company

We see companies of every size, all at once.

100 million

code reviews



One company

Many teams get stuck in the same four places.

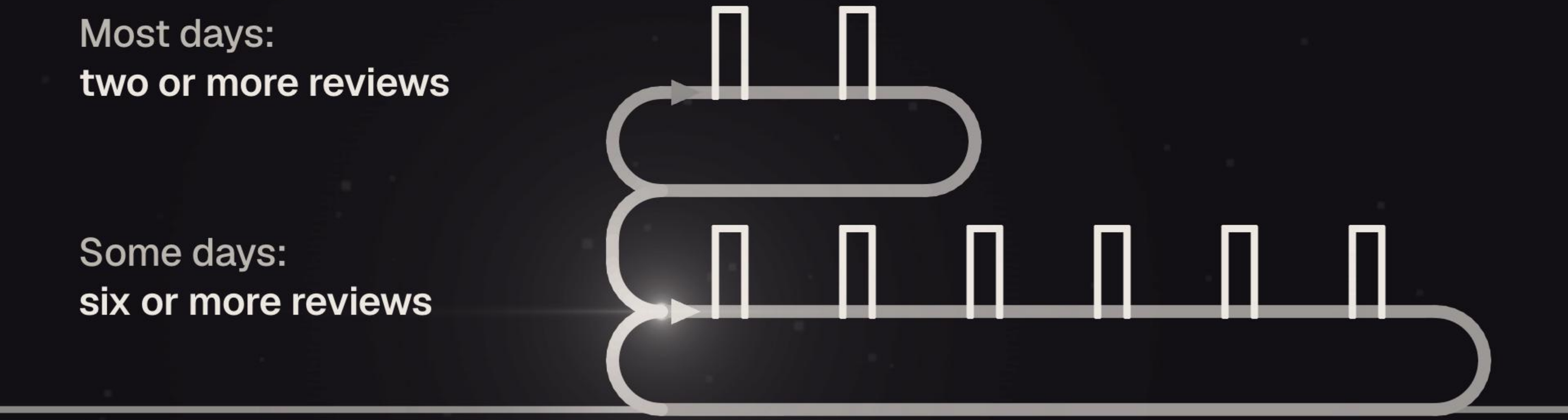


The agents already work in a loop.

One agent, one repository, one day

Most days:
two or more reviews

Some days:
six or more reviews



The agents are all about the same.

Each bar: one coding agent



On about

7 in 10

changes, our first review left at least one comment.

Larger organizations often run more than one agent.

Share running two or more coding agents



Approving a change
no longer means understanding it.



Review

Understanding

Prioritization

Security

A reviewer is accountable for a change they didn’t write.



Nothing says which changes matter.



Understanding

Prioritization

Security

Just 10% of pull requests got half of all review comments.



So you don’t need to read everything.

You need to find that 10%.



Someone asks whether any of this is safe.



Up to 2.74x

Cross-site scripting, the biggest gap

the security issues in pull requests
with an AI co-author
compared with pull requests without one



Without one

With an AI co-author

CodeRabbit study, 470 open-source pull requests, 2025

AI models now help find weaknesses in old code.

Old code



27-year-old code

An AI lab's report, 2026

A security check you ran once goes out of date.

One week in October

Oct 21

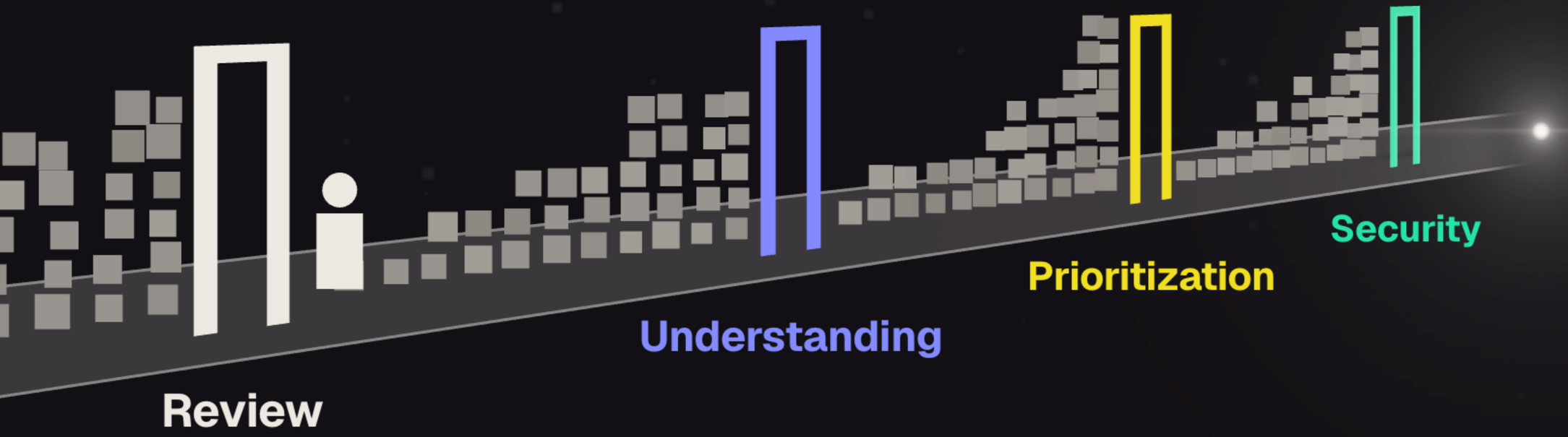
PHP AI helped find a decade-old flaw

LiteLLM Hijacked again, shipped a password stealer

Microsoft 365 Hackers' AI agents targeted hundreds of organizations

Sources: PHP Foundation, Wiz, LiteLLM and Microsoft, October 2026

Teams are solving agentic coding, but not yet agentic shipping.



What we lived on the inside



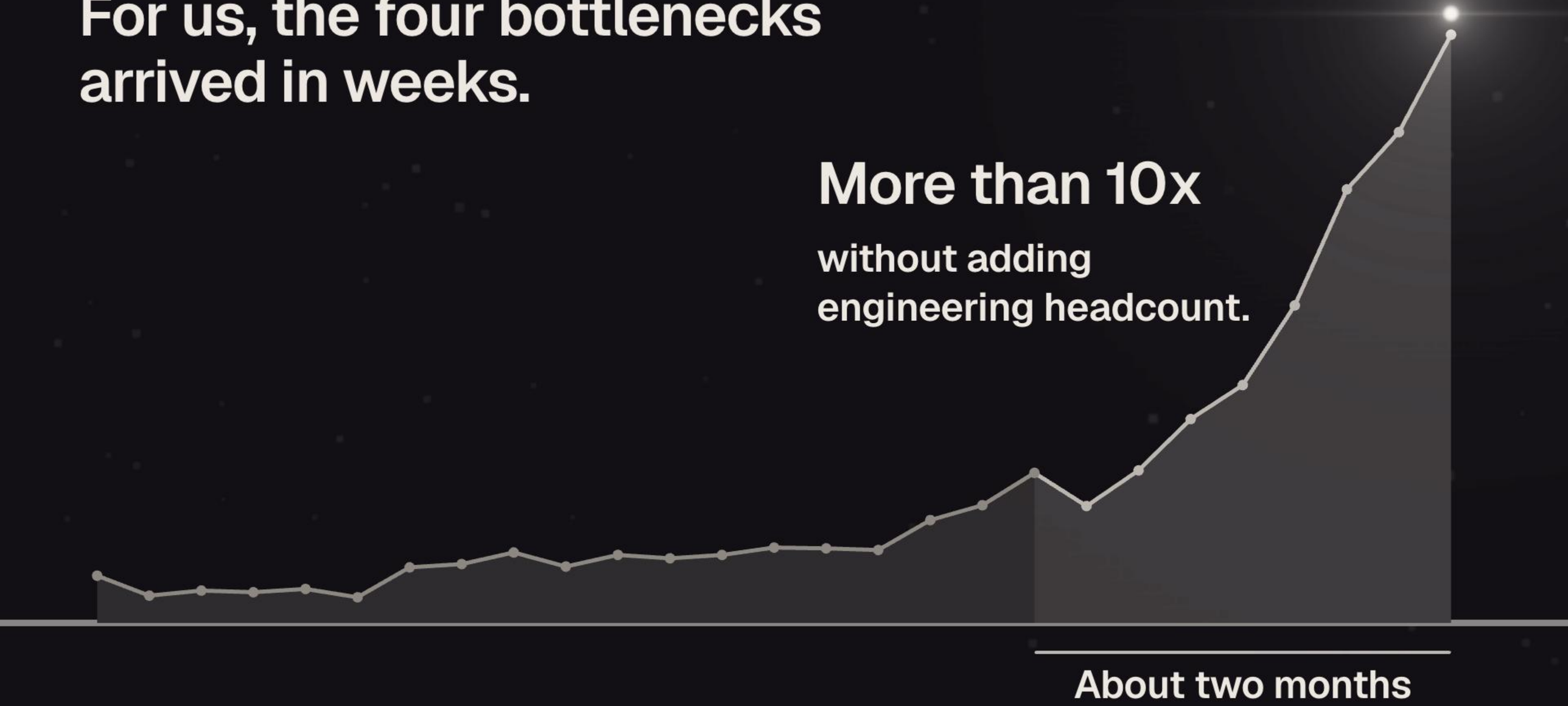
03

Our first climb took about four months.

Merged pull requests a week, across our code

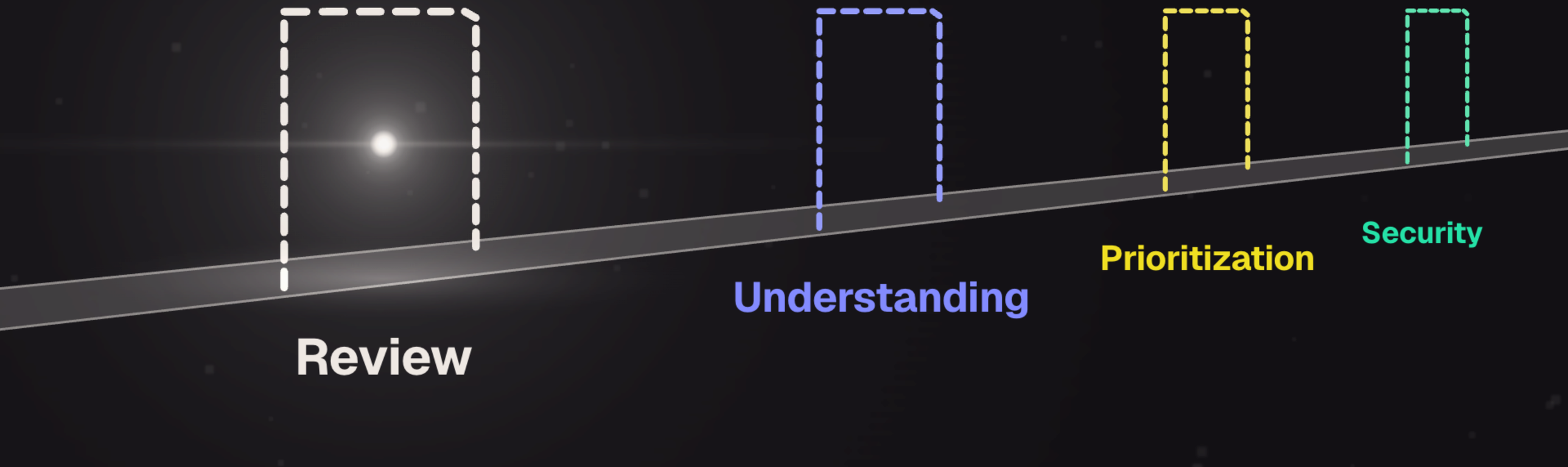


For us, the four bottlenecks arrived in weeks.



Each of the four things we got for free
now has to be built

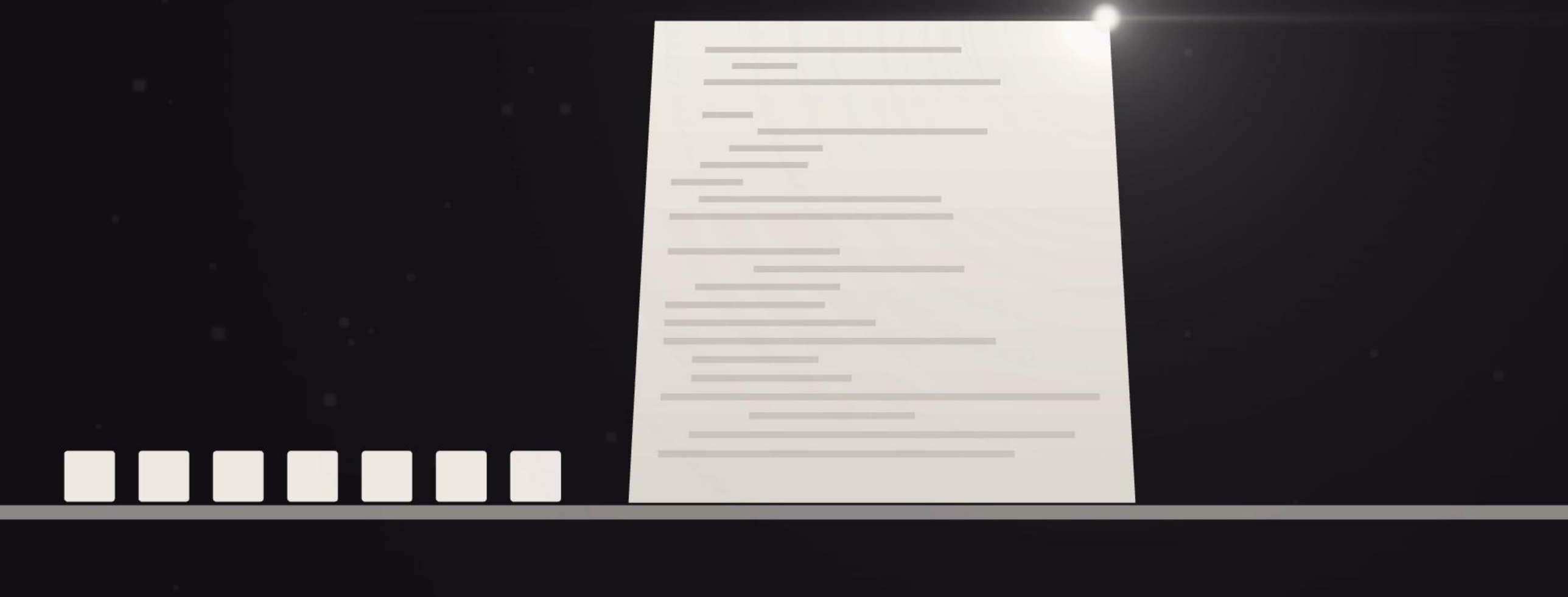
on purpose.



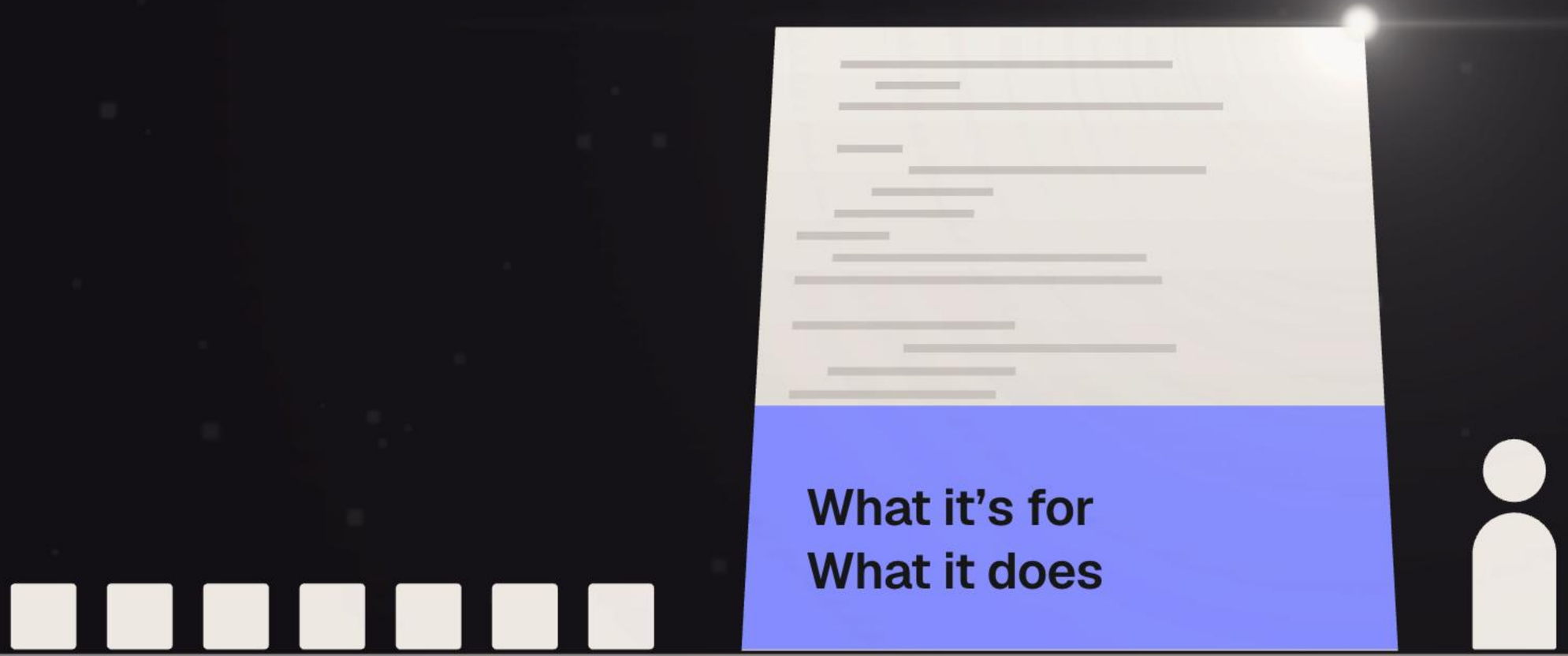
The best code comes from an adversarial review loop.



About 1 in 8 of our merged changes is 1,000 lines or more.



Changes a person reviews come with an explanation.



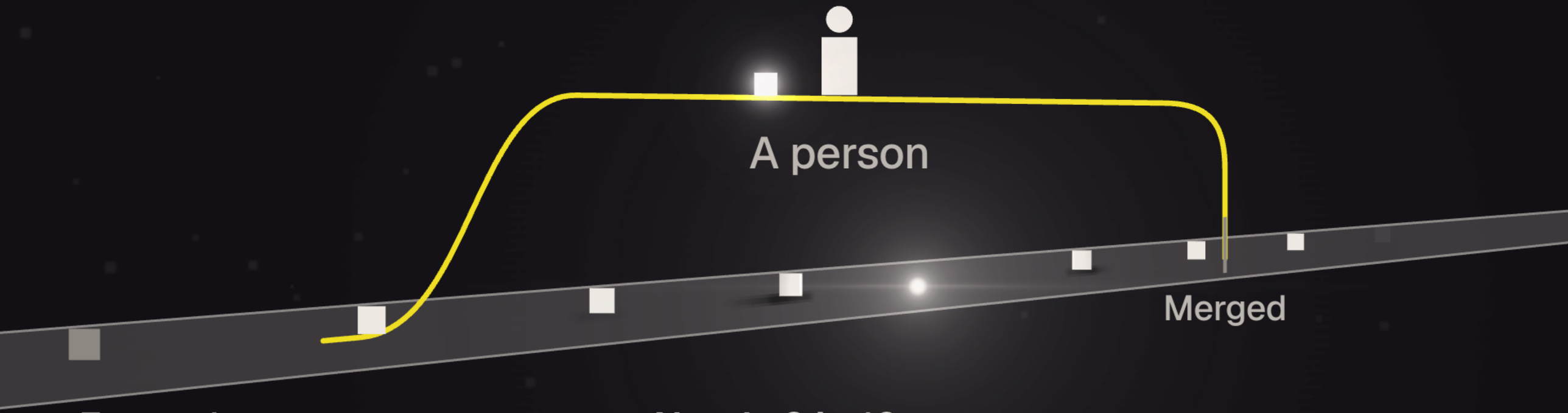
03 · **Prioritization**

The changes with the most value and risk go to a person.



Every change gets
an adversarial review

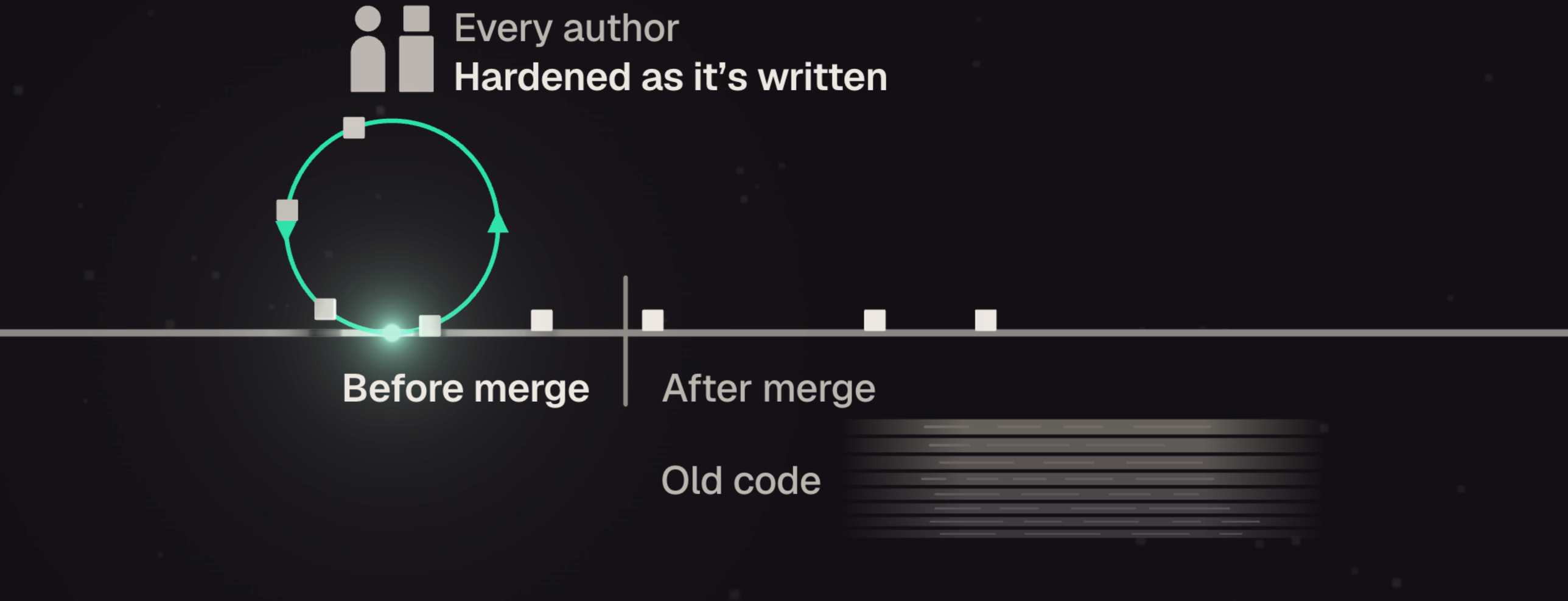
The changes with the most value and risk go to a person.



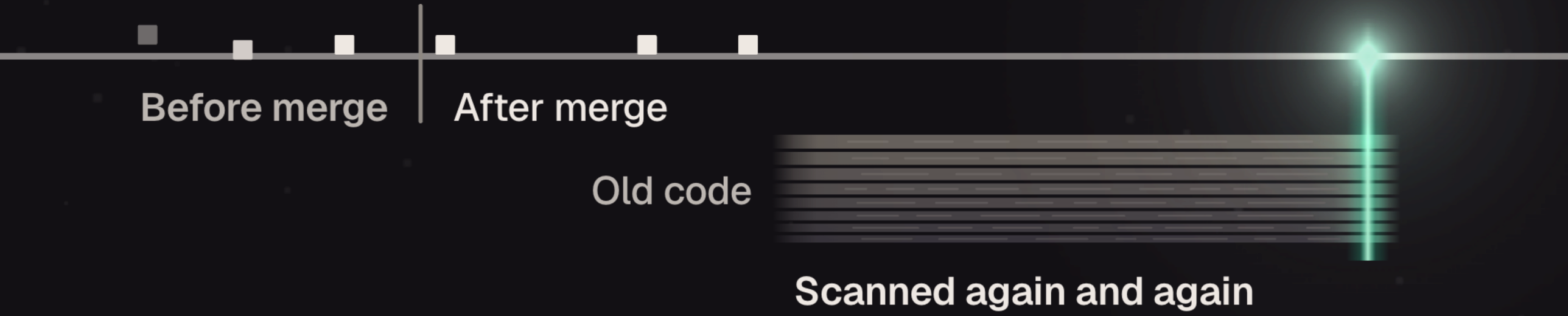
Every change gets
an adversarial review

**Nearly 9 in 10 merge
with no person reviewing them**

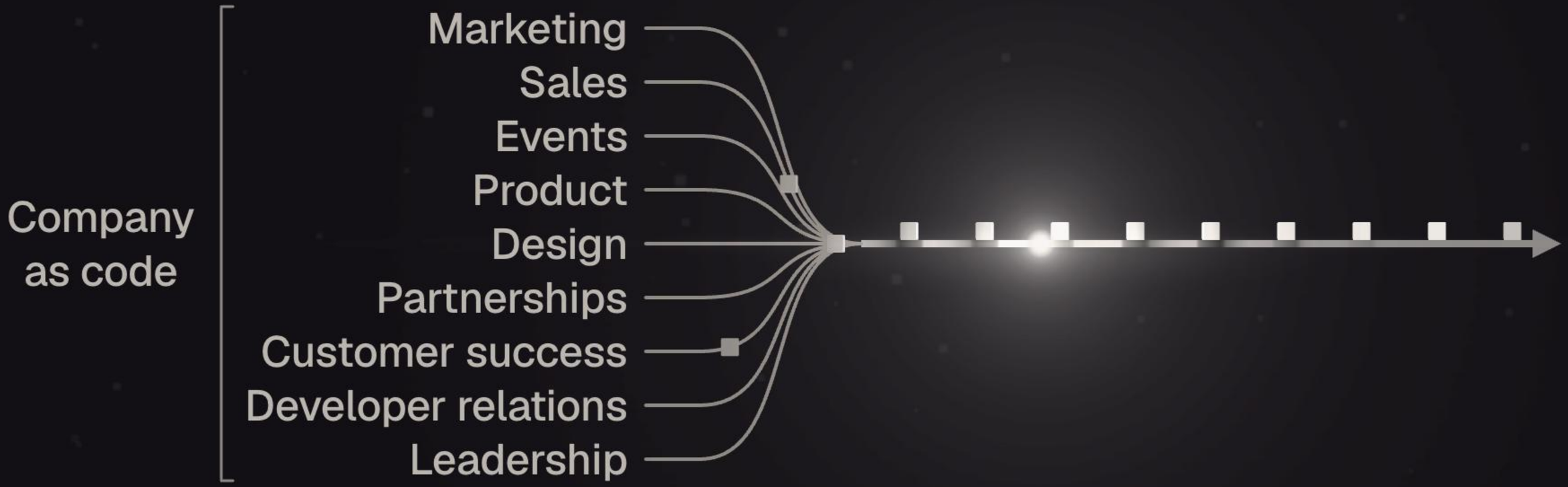
Security is everybody’s problem now.



After merge, we keep scanning old code.

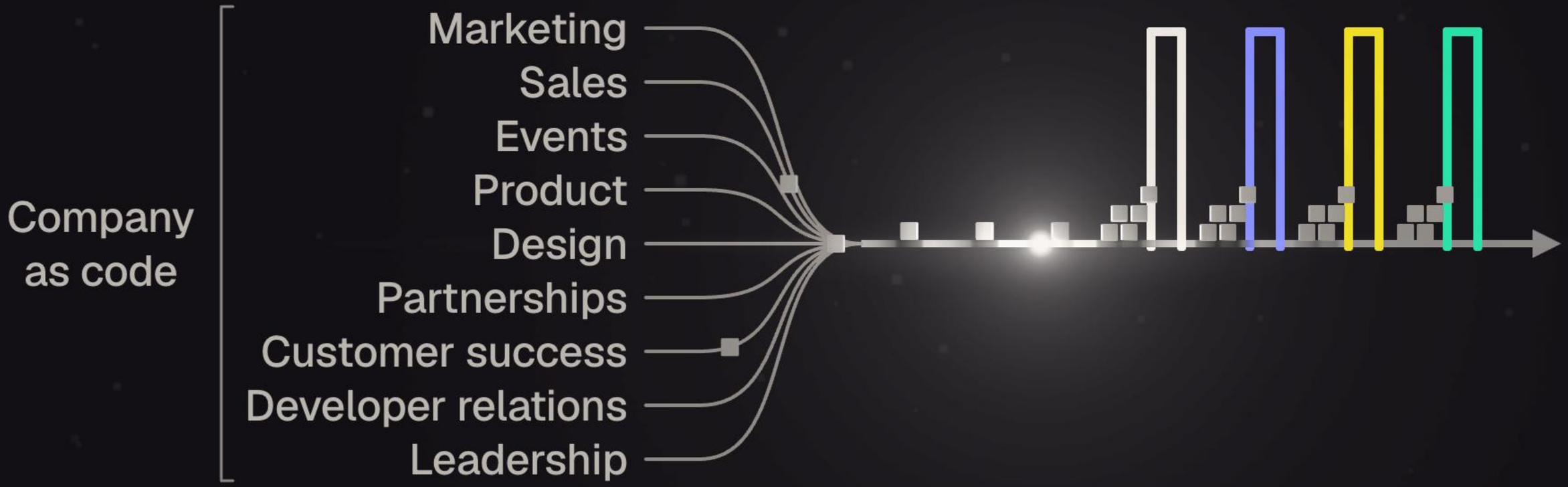


Coding agents spread beyond engineering.

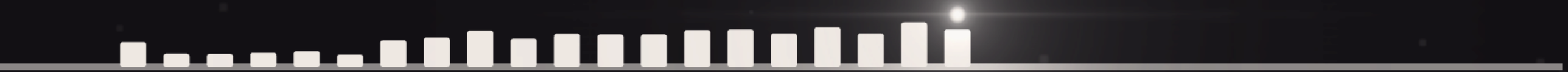


Coding agents spread beyond engineering.

So the four bottlenecks are every team's now.



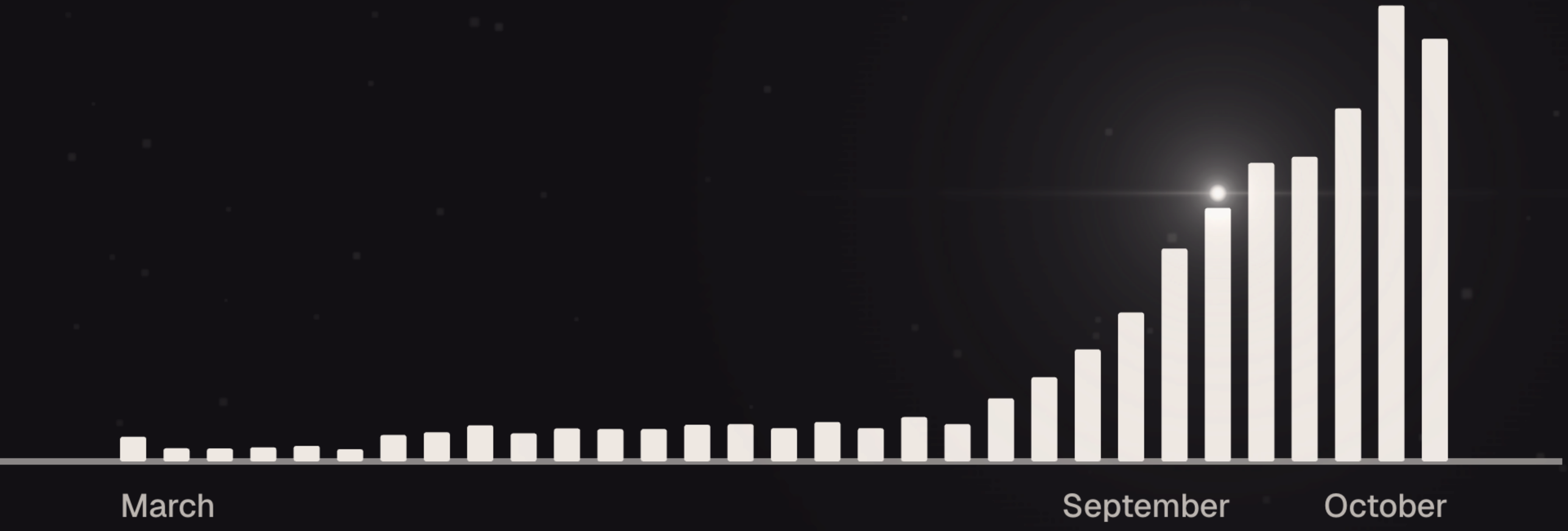
Product code



March

We cleared each bottleneck by solving our own problems.

Product code



We cleared each bottleneck by solving our own problems.

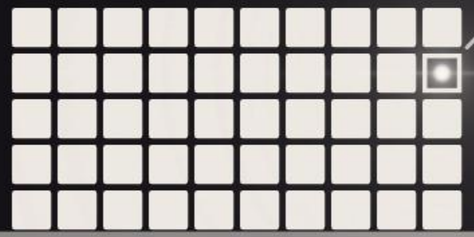
Product code

About **10x** more merged changes than in March



More changes. A smaller share rolled back.

Product code

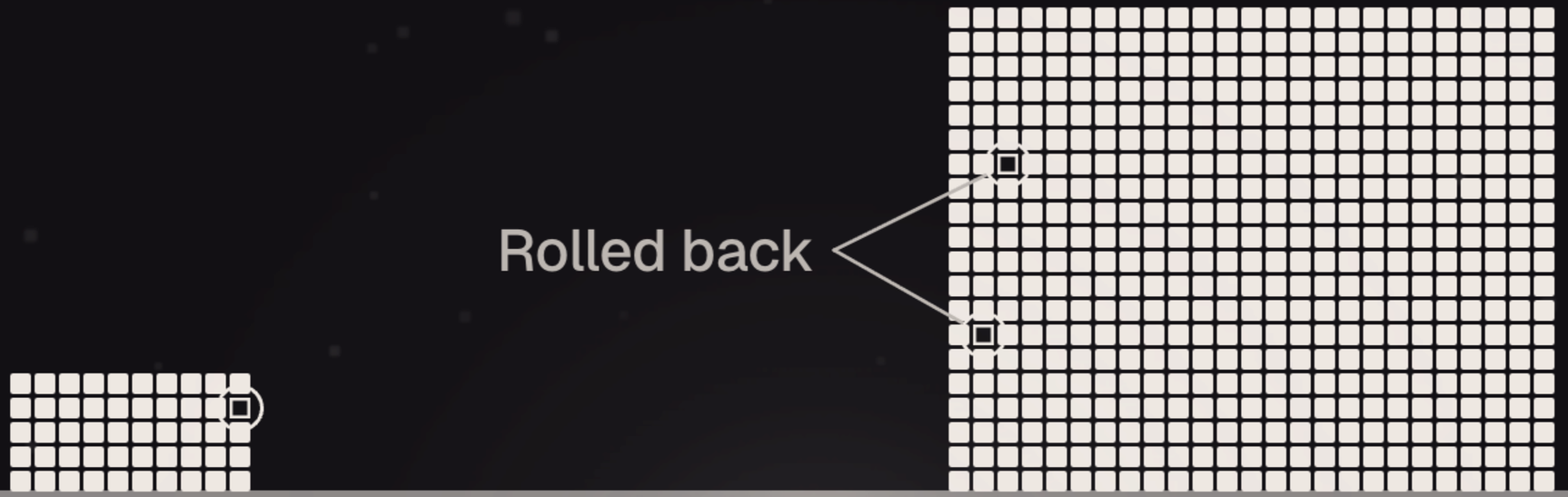


March

About 2%

More changes. A smaller share rolled back.

Product code



About 2% → Under 0.5%

What it takes



04

Today's questions

What does it take to move from human-centered to fully agentic software development?

And what does it take to ship dramatically faster without sacrificing quality or security?



Today's questions

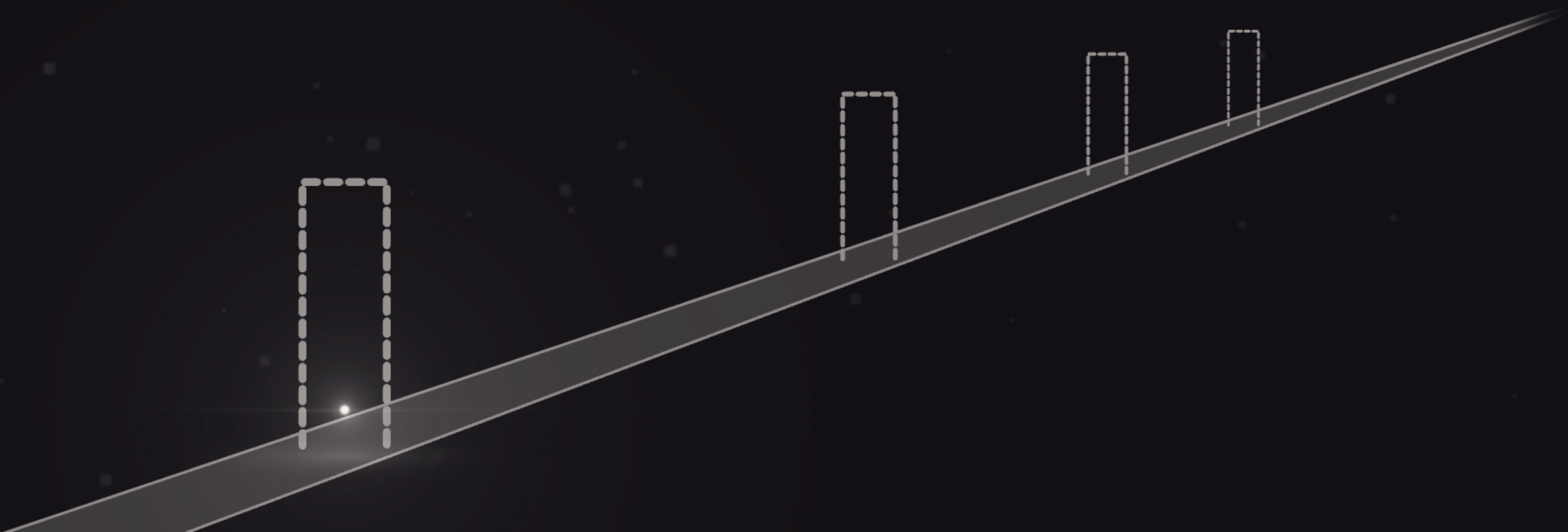
What does it take to move from human-centered to fully agentic software development?

And what does it take to ship dramatically faster without sacrificing quality or security?

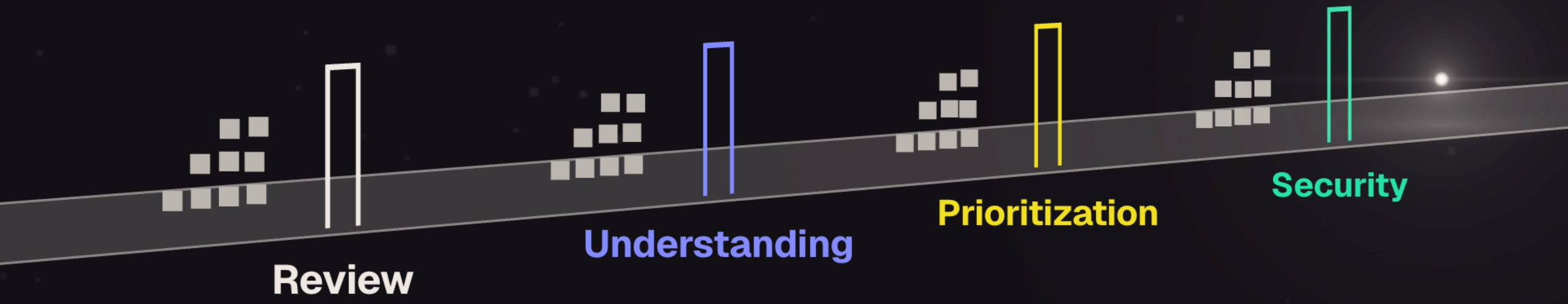
It takes a system you can trust.

Trust used to be a side effect of expensive code.

Now you have to build it.

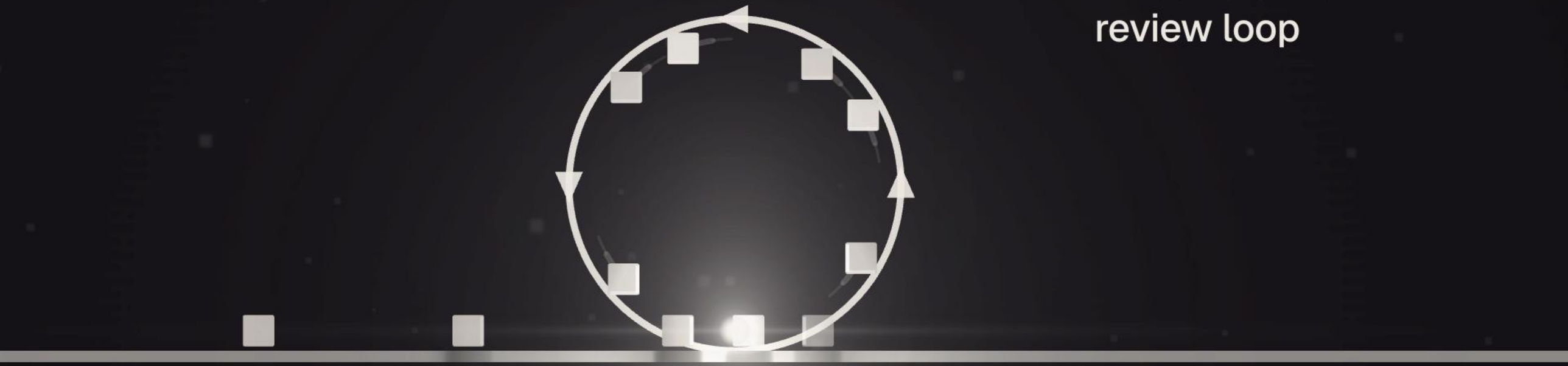


A leader's job becomes building the system that clears these bottlenecks.



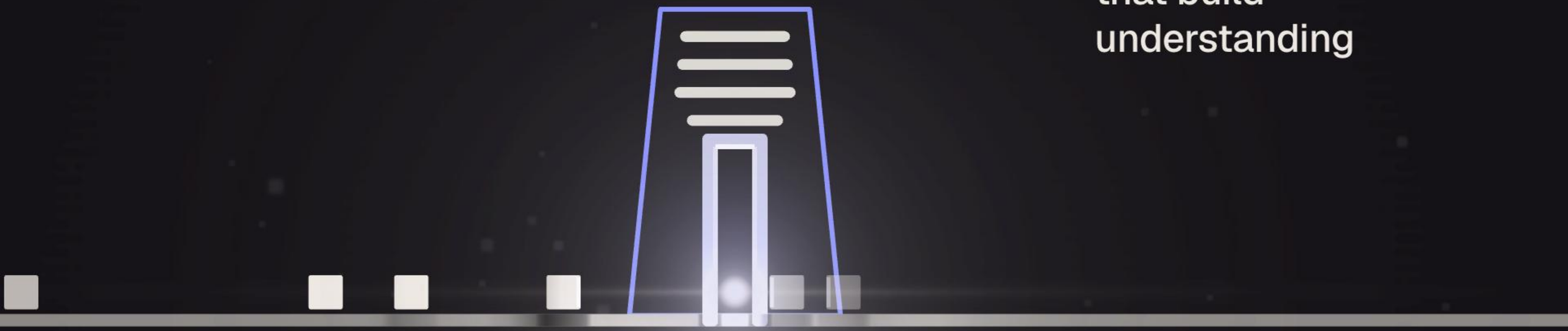
Review

An adversarial review loop



Understanding

Explanations
that build
understanding

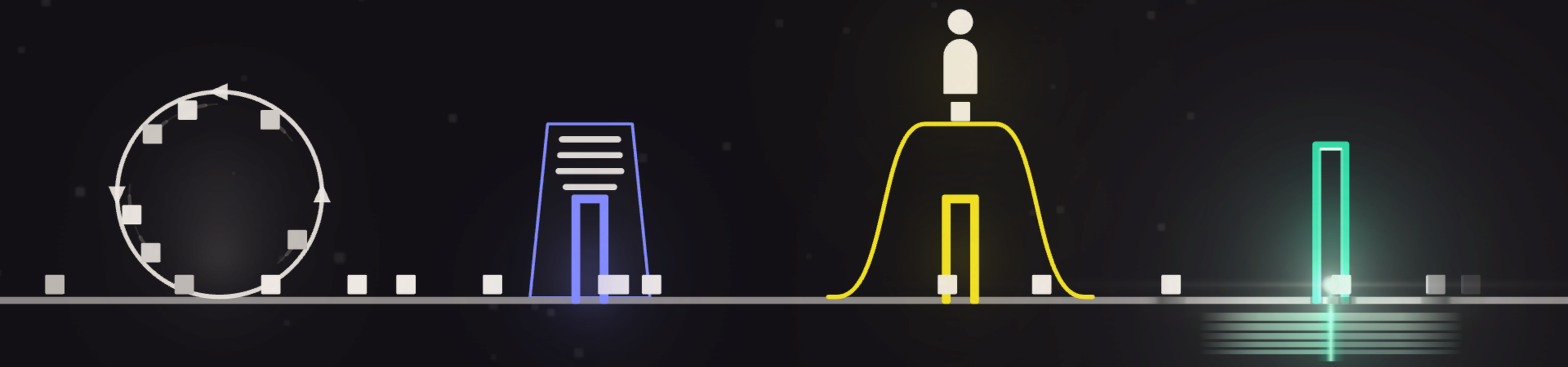




Prioritization

A person for
the changes that
matter most

A leader's job becomes building the system that clears these bottlenecks.



Review

An adversarial review loop

Understanding

Explanations that build understanding

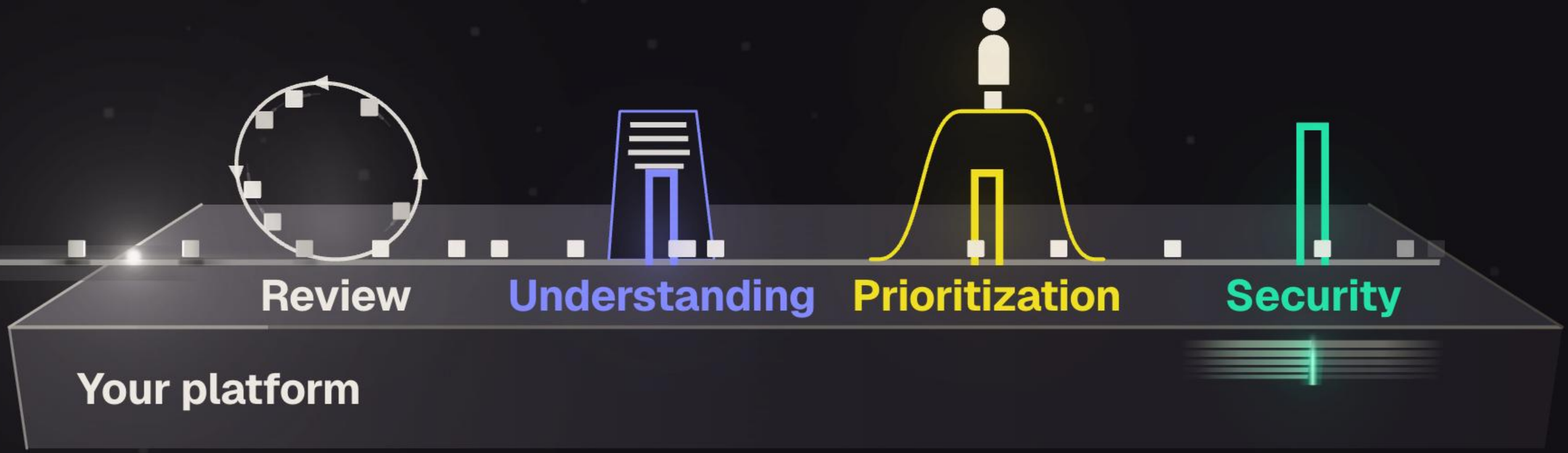
Prioritization

A person for the changes that matter most

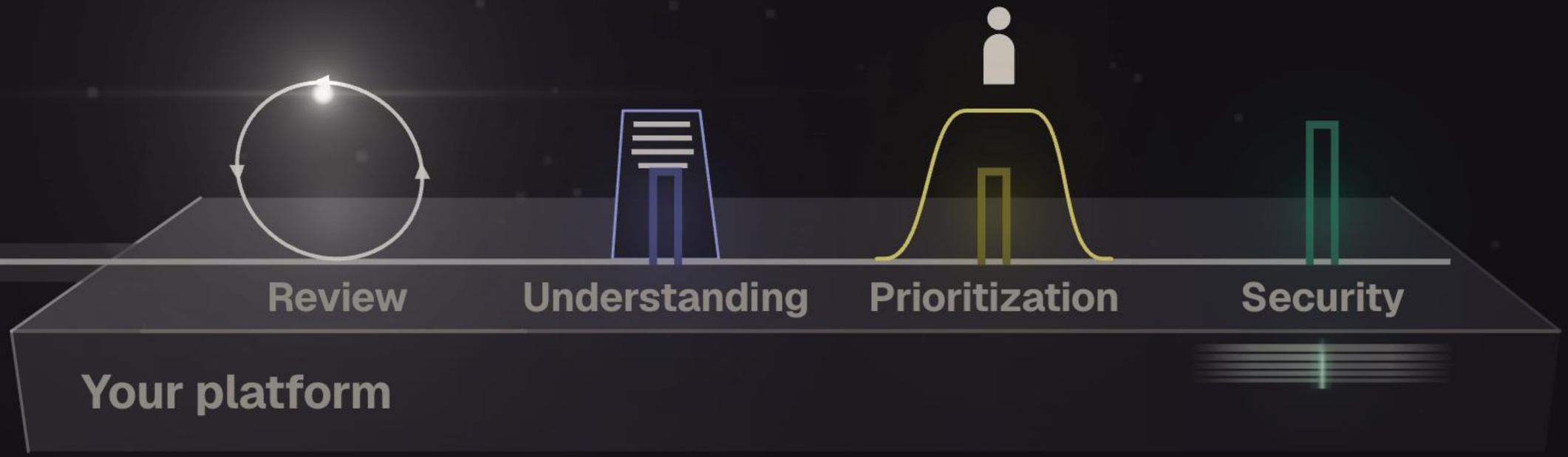
Security

Governance for all code, new and old

For many of you, that system is your platform.

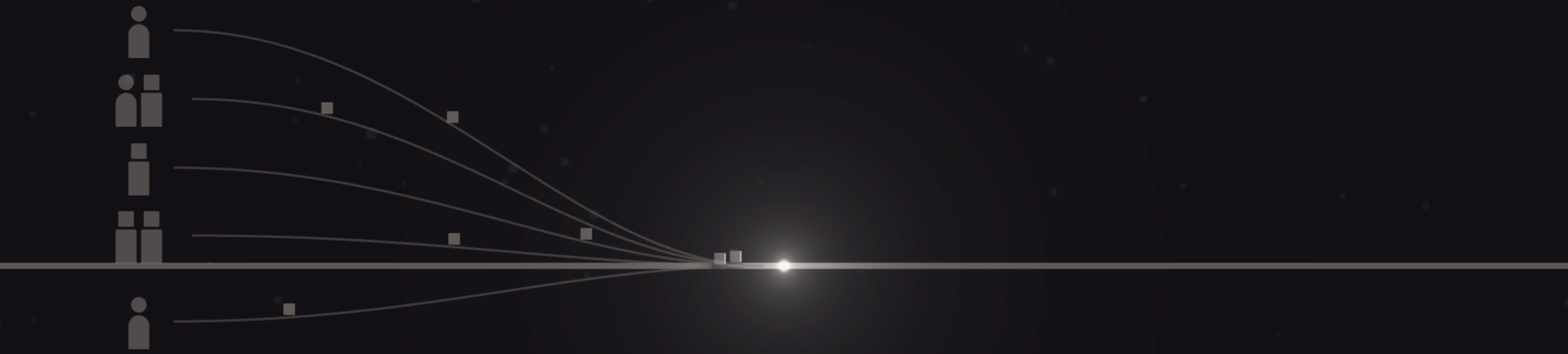


Agentic Change Management



Agentic Change Management

How a team decides which changes to trust when anyone and anything can write them.



Agentic Change Management

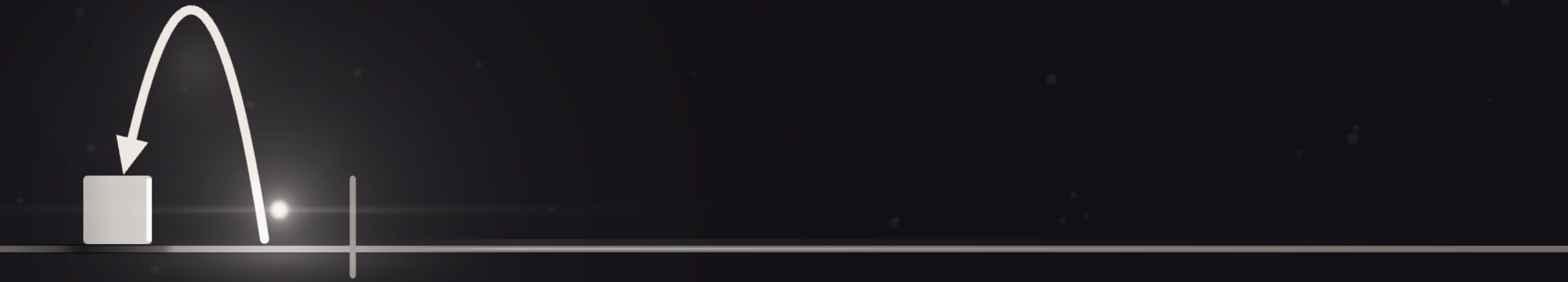
Trust must be earned
through independent assessment,
grounded in organizational context,
verified through human judgment.

The discipline

Agentic Change Management

Adversarial review

AI challenges
every change
before it merges.



The discipline

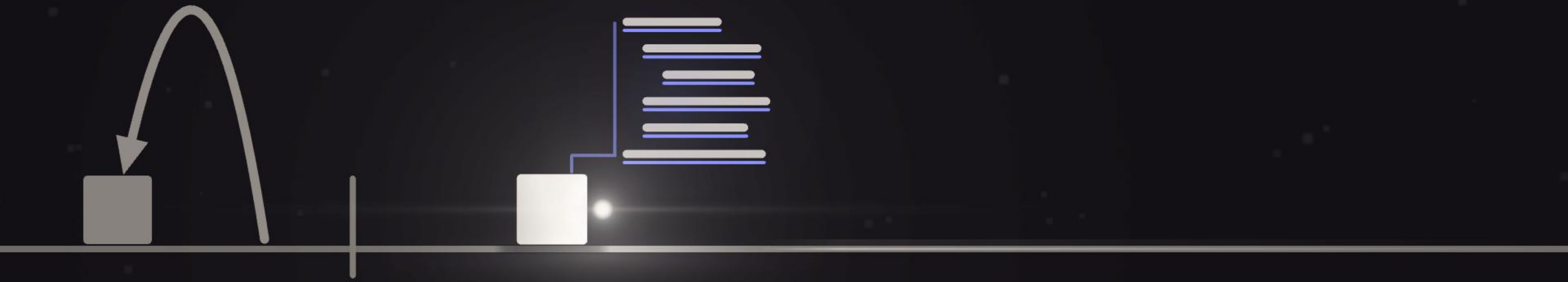
Agentic Change Management

Adversarial review

AI challenges every change before it merges.

Explainability

Reviewers know what each change does and why.



The discipline

Agentic Change Management

Adversarial review

AI challenges every change before it merges.

Explainability

Reviewers know what each change does and why.

Prioritization

People judge the changes with the most at stake.



The discipline

Agentic Change Management

Adversarial review

AI challenges every change before it merges.

Explainability

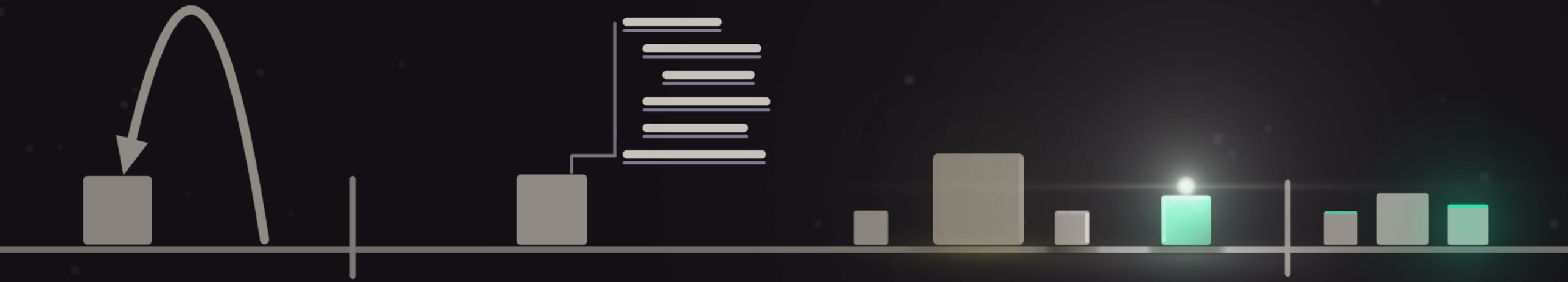
Reviewers know what each change does and why.

Prioritization

People judge the changes with the most at stake.

Lifecycle security

Code is hardened before merge and scanned after.



The discipline

Agentic Change Management

Adversarial review

AI challenges every change before it merges.

Explainability

Reviewers know what each change does and why.

Prioritization

People judge the changes with the most at stake.

Lifecycle security

Code is hardened before merge and scanned after.



How much agent work we can trust.



For decades, the hard part of software was writing it.

For decades, the hard part of software was writing it.

Now the hard part is trusting it.