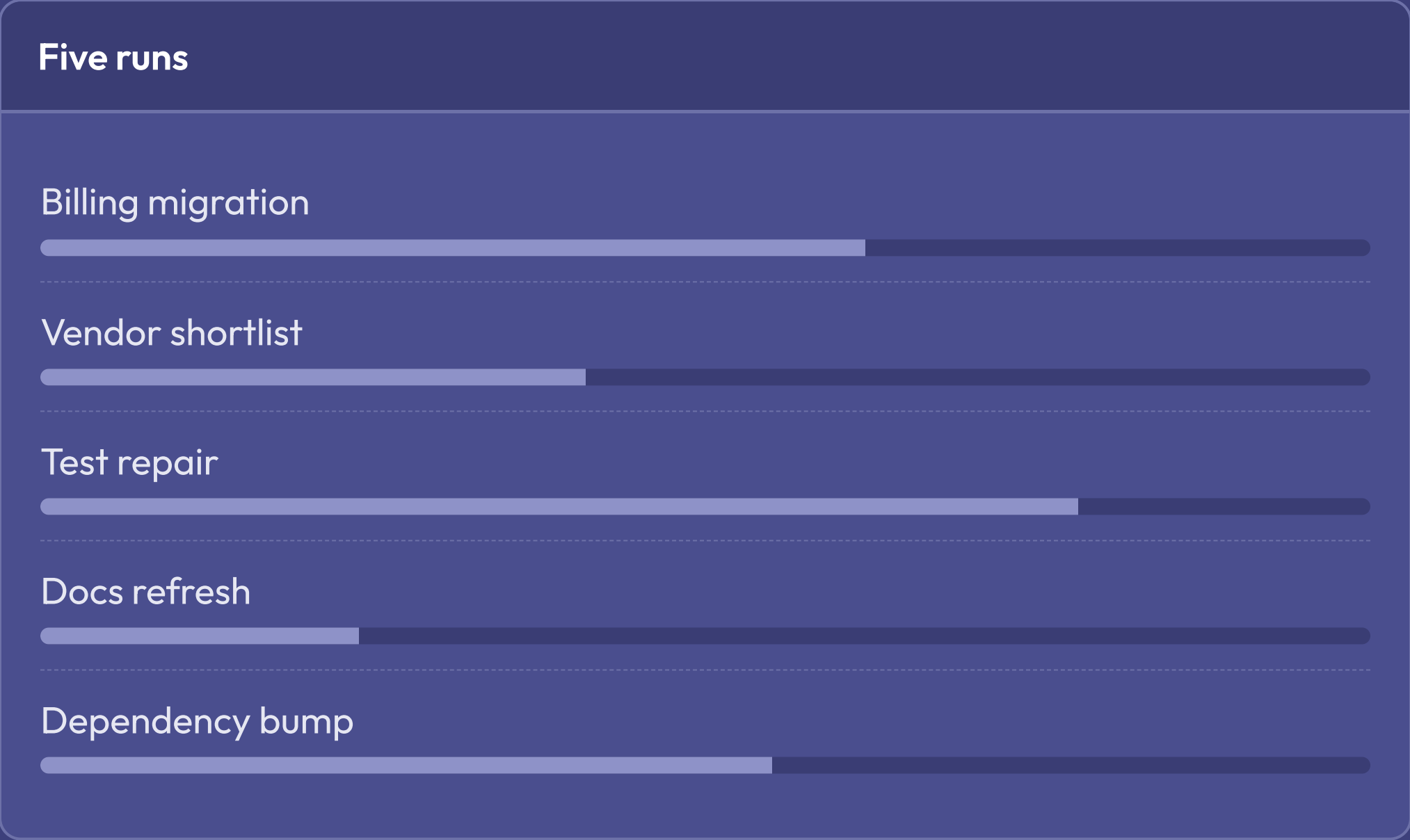


Five agents running. Which one is STUCK?



Where the research came from

We went through what Google, Anthropic, Microsoft, LangChain, Asana and Notion have shipped in their agent products, alongside the studies published in 2025 and 2026.

Twenty patterns came out of it. **Four deal with the same problem:** more than one agent working at the same time, and one person meant to be watching.

Two agents is manageable. Eight is a second job. **The effort of watching climbs with every agent you add.**

1: Watching does not scale

Each agent you add costs the same attention again.

One agent

You read the transcript. That works.

Five agents

Five transcripts, five tabs, five places to look.

Five runs, no status

Billing migration



Vendor shortlist



Test repair



Docs refresh



Dependency bump



What people prefer

Steering from above, rather than driving each one.

The catch

Nothing on this screen says which one needs you.

The finding comes from research on people supervising **robot swarms**, which hit this problem a decade before agents did.

2: The shape you can supervise

One lead agent plans and hands out the work. The helpers never talk to each other.

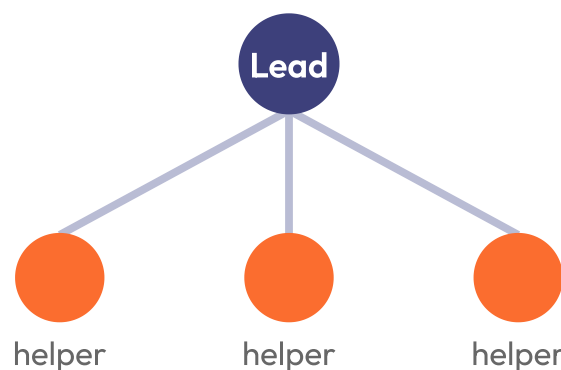
The lead plans

And pulls the results together at the end.

Helpers work alone

Each with its own memory of the job.

Lead and helpers



You read the plan

Not the helpers.

Only when work splits

If every helper needs the same background, this does not pay.

Anthropic reported this beating a single agent by **90.2%** on their own research tests, at roughly **15 times the cost** of a normal chat.

3: The shape you cannot

Agents talking to each other, sharing notes as they go.

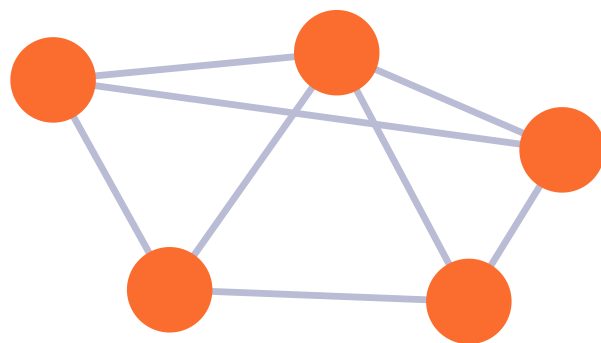
Flexible

Any agent can pick up any thread.

Hard to read

Any agent may have influenced any other.

Everyone talking to everyone



When it goes wrong

You cannot reconstruct who decided what.

The trade

More capability, far less oversight.

The rule in the previous shape, that helps never talk, is exactly **what makes the work checkable.**

4: Goals up front, steps on request

Show what each agent is trying to achieve. Hide how, until something fails.

Outcomes first

People look at results before process.

Collapse the rest

Steps stay hidden until a goal fails.

Trip to San Francisco

- ✓ Flight booked under \$400

- › Hotel within 2km of the venue

- ! Evening plans
clashes with your flight home ·
open

Progress per goal

Readable in seconds, not steps.

From OrchVis

A prototype that turns a request into goals with checkable conditions.

A screen with every step on it is **a screen nobody reads.**

5: Pause one branch, not the run

When two agents disagree, stop the pair involved and let the rest carry on.

Flag it in place

At the goal it affects.

Keep the rest running

One clash should not stall everything.

Two goals clash

The 18:40 flight leaves before the show ends. Hotel and restaurant agents are still running.

Take the 21:15 flight	+\$85
-----------------------	-------

Move the show to Thursday	no cost
---------------------------	---------

Drop the show	refund \$60
---------------	-------------

Offer repairs

With the cost of each one shown.

Still unsolved

Named as one of six open design challenges in 2025.

Left alone, a clash gets settled by **whichever agent finished last.**

Supervising a crowd

- 1 Watching does not scale.** Every agent you add costs the same attention again.
- 2 One lead, helpers that do not talk.** The shape you can still check.
- 3 Only split work that genuinely splits.** Otherwise you pay more and see less.
- 4 Show goals.** Hide the steps until something fails.
- 5 Pause the branch, not the run.** And price each way out.

The question a supervision screen has to answer in ten seconds: which one needs me?

How many agents have you had running at once?

Comment with a number.

If you change one thing

Sort by what is blocked

Why this one

A board of parallel runs is only useful if the ones waiting on a person sit at the top. Everything else on it is reassurance.

That one row is the work. **The rest is a progress bar.**

Go deeper

The whole pattern library

On the site

Seven families, forty-one patterns.
What each one prevents, and the
product or the paper it came from.
Free, and no sign-up.

thinkbigleaders.com/resources/agent-patterns

Or talk it through

Curious, and nowhere near a decision?
Good. Have a use case in mind? Also
good. Already building one, or want
agents inside your own product? Bring
what you have.



Thirty minutes with Think Big Leaders.

Scan the code to pick a time.