What You Can't Rent

A practical brief for executives deciding who should own AI

There are AI tools in your company right now. Somebody chose them. Somebody is paying for them. And if you had to name the person accountable for what they return, you would probably have to think about it for a second.

That isn't a gap. It's a decision. You've decided to let the program be shaped by whoever is most enthusiastic about it.

Whether you've hired for it, appointed someone inside, engaged a firm, are mid-search right now, or have done none of it yet, the same question applies:

Who is designing the whole thing while everyone else operates its parts?

Among US companies under 500 people that posted an AI leadership role in the last ninety days: nine in ten had not posted one in the previous eighteen months, and seven in ten posted no other AI role at all — no engineer, no analyst, no architect. One seat, created for the first time, with nobody beside it.

Most companies are wrong about the answer. And what they're wrong about is which part of this an outside hand can supply: **you can rent design capability. You cannot rent internal accountability.**

SIX SECTIONS · ONE MISSING FUNCTION

1. FOUR JOBS
2. HIRED ALONE
3. URGENT VS DESIGN
4. TWO SHAPES
5. YOUR ROW
6. ALREADY INSIDE

1

The job is four jobs

Most first AI roles are four jobs wearing one title.

THE JOB	WHAT IT TAKES	CAN AN INTERNAL OWNER HOLD IT?
Platform	Select tools, manage vendors, control cost	Yes
Adoption	Change workflows, train people, build internal support	Yes — this becomes most of the week
Governance	Set data rules, make risk calls, assign accountability	Yes, given the authority
Build	Analyze, design, build, test, deploy — one initiative at a time	Yes, with the right hire

Read that last column again. Mostly yes.

A capable internal owner can hold all four. That isn't the problem. The problem is the last five words of the Build row: **one initiative at a time.** This tool. That workflow. The automation somebody needs by Friday. Each one can be designed well and run well.

Nobody owns the space between them.

That's where the cost lives. Two teams maintain the same information and don't know about each other. Three departments pay for the same capability under different product names. Last quarter's automation quietly contradicts this quarter's policy. A workflow is technically successful and creates more work somewhere nobody measured.

None of those problems appears inside a single initiative. All of them appear across initiatives. So the missing function isn't initiative design. It's **designing the whole** — deciding how the pieces fit together, where they conflict, which capabilities should be shared, and what shouldn't exist at all. It isn't a fifth box in the job description. It's the work of making the four boxes cohere.

2

And one person is doing all four

I went looking for what the market actually does when it creates this seat. I took every US company under 500 people that posted an AI leadership role in the last ninety days and asked two plain questions of the hiring data.

Had they done this before? Nine in ten had not posted an AI leadership role in the preceding eighteen months. For almost all of them, this is the first time.

Were they building a team around it? Seven in ten posted no other AI role in the same window — no AI engineer, no data scientist, no analyst, no architect. One seat.

Put those together and you get the actual shape of the thing. The person you hire walks into a company that has never had this role, so there is nobody inside who can tell them what the job is. And they are not the head of a function. **They are the function.**

That matters more than the wording of any job description. Four jobs and no team doesn't divide the work – it queues it. Platform questions, adoption requests, governance calls, and the build somebody needs by Friday all arrive on one calendar, and they arrive in the order other people need them, which is never the order that makes the program coherent.

One more thing worth knowing about the company you're benchmarking against. Four out of five organizations creating this role have more than 500 people, and nearly a third have more than ten thousand. The salary comparisons you're reading are set by that group. **If you're under 500 and you've opened this req, you are early** – which is a good position, and it means there is no playbook waiting for you inside the building.

3

The urgent work eats the design work

If you've made the hire, the candidate probably knew more about the subject than anyone on your side of the table. That isn't a hiring failure – it's the reason the role exists. But it means you evaluated what you could judge: how they'd land with your people, whether they'd survive your operating cadence, whether you'd want them in a hard meeting. Those were the right things to examine. They just don't tell you whether anyone will protect the work between the initiatives.

If you're still mid-search, this is the cheapest moment you will ever have to fix it. Deciding what the seat is *for* costs you a paragraph before the offer. It costs a renegotiation after.

And whoever takes the job inherits an operating load that starts at full volume, because people are already doing god-knows-what. Somebody has a personal account they expense. Somebody built an automation three teams now depend on and no one has reviewed. A department is moving company data through a workflow leadership has never seen.

Getting your arms around that is not a distraction from the owner's job. **It is the job.** It's relationship work, political work, and constant work. It runs on standing, continuity, and being findable when someone has a question they'd rather not put in writing. An outsider cannot own it for you. Your internal owner's hands are full. Correctly so.

Which is why designing the whole goes unprotected – not because the person you appointed is incapable, but because the urgent work of operating the program keeps eating the work of designing it.

The task with no deadline is the task that loses.

4

Two jobs, two shapes

The distinction isn't between an internal operator and an outside genius. It's between two kinds of work that run on different conditions.

	THE INTERNAL OWNER	THE OUTSIDE ARCHITECT
Focus	Organizational execution	The whole system, across initiatives
Owns	Priorities, adoption, final decisions	Options, architecture, consequences, recommendations
Load shape	Back-loaded – grows as adoption grows	Front-loaded – should decay by design
Runs on	Presence, continuity, trust, standing	Range. Having seen systems succeed and fail
Says	*"This is what the business will support."*	*"This is what each choice will require."*
Fails as	A helpdesk with no authority to prioritize or refuse	A dependency that never ends

The decision rights are the important row.

An outside architect should not decide what business you're in or what your priorities are. The architect's job is to make the choices and their consequences visible. The internal owner – and above them, the executive team – decides.

Which is why a firm can't simply replace the internal owner. You can rent design capability. You cannot rent internal accountability.

And you may not need to rent the design capability either. If someone inside already has the range, the experience, and the protected time to do the whole-system work, give them the mandate and make the work explicit. The important thing isn't that the capability comes from outside. It's that it exists, and that someone's week is actually reserved for it.

5

Where you are right now

WHERE YOU ARE	WHAT YOU HAVE	WHAT MAY BE MISSING	WHAT YOU NEED NEXT
You're mid-search	An open req and a shortlist	A definition of what the seat is *for*	Decide before the offer whether this role owns designing the whole or only running it — then write it into the mandate
You made the hire	An internal owner	Protected capacity for the whole-system work	Find out whether the capability exists internally; bring it in temporarily if not
You appointed someone inside	Standing, relationships, demand ownership	Technical capability and a model for the program	Build capability around them before adding permanent headcount
You brought in a firm	Analysis and design, probably	Internal demand ownership and final accountability	Name an internal owner — often someone already on your payroll
Neither yet	Whoever's most enthusiastic	Both	Name the internal owner first; use temporary help before committing to a build or a permanent hire
You have both	Ownership and design capacity	Possibly nothing	Make the decision rights explicit, and let the outside role decay

If you can already name your row, you have what you came for. Reply to the email that brought you this — or write me at scott@sufficiently-advanced.ai — with the row, and I'll tell you what I'd look at first.

If you've already hired, the recommendation to *also* bring someone in is the most self-serving argument in this brief. So don't take it on my word. Take it from the work.

Look at the last thirty days. If your internal owner spent the month answering requests, reviewing tools, running sessions, resolving risk questions, and negotiating priorities — the operating function is real and it's working.

Now look at the space between the initiatives. Who traced a workflow end to end, across departments? Who tested whether a proposed automation should exist at all? Who went looking for duplicated capabilities, contradictory decisions, or a system you'll regret building? Who named something that should be consolidated, retired, or stopped?

If you can't name the work product, the function doesn't exist.

6

The internal owner may already work for you

The best internal owner is often not a new hire.

If the job runs on relationships, horse-trading, knowing whose support is needed and where the bodies are buried, the person best positioned for it may already be inside. A COO. An operations lead. A chief of staff. A functional leader with enough authority to set priorities and say no.

They aren't missing standing. They may be missing technical capability, protected capacity, and a model for how the program should work.

Capability can be built in months. Standing takes years.

If you haven't started a search, that often makes the cheapest correct answer an internal owner who already knows the company, supported for a while by someone who has designed one of these before. No search. No six-month ramp. No permanent outside dependency.

If the search is already running, the same fact points somewhere else. Whoever you hire will need that standing on day one and won't have earned it yet — and on the evidence above, they'll be building it alone. So name the person inside who will lend it: the sponsor who opens doors, settles the priority fights, and absorbs the political cost of the first hard call. That person is part of the mandate. Leaving them unnamed is one of the quieter ways a good hire fails.

ABOUT THIS BRIEF

I've held both of these roles, in a previous technology wave and in this one.

I've been the outside architect who designed the system and then stepped back so the team could operate it. I've also been the internal owner whose real work was maintaining relationships with department heads, understanding what they needed next, doing the horse-trading, and being the face of the program so people didn't feel it was happening to them.

They are not the same job.

I'm a Claude Certified Architect. My view of the outside role is that it should be temporary by design: an architect still indispensable in year three has failed at the job.

If it needs more than a conversation, I run a one-week audit: every AI tool you're paying for and what it returns, an integration map of what talks to what, two workflows traced end to end, three prioritized findings, and one scoped first build. $3,000, five working days, fixed scope. That week is spent entirely in the space between your initiatives — the spend ledger finds what you're paying for that nobody opens, the integration map shows what actually fits together, and the scoped build is one decision about what should exist next. You own the document outright. No license, no dependency, nothing owed in either direction afterward.

The brief is yours to keep whether or not we talk.

If you want to know which function you're missing, tell me which row of that table you're in — reply to the email that brought you this, or write me directly. I'll tell you what I'd look at first.

Scott Jennings

scott@sufficiently-advanced.ai

Method, verified August 18, 2026. Source: TheirStack job-posting data, US postings from the preceding ninety days, direct employers only, staffing and IT-services excluded, filtered by regex to non-technical AI leadership titles — 1,047 postings across 627 companies, of which 120 companies have 500 employees or fewer. The "first time" and "no other AI role" figures were measured on a 28-company sample drawn from that sub-500 group and are subject to sampling error at that size; both describe a ninety-day observation window, so a team hired outside that window would not appear. Company-size shares are computed on companies, not postings, because larger employers post two to three times more often and posting-weighted shares overstate them. Employer records occasionally split one organization across several entries, so company counts are upper bounds. Full queries and per-pattern validation are retained and reproducible.