

How to become an AI agent manager.

Ninety days, no computer science degree, and a portfolio you can build on your own time. A practical route into the job companies are quietly desperate for and have mostly forgotten to hire.

BY DAVE BROWN

Every so often a job appears in the gap between two things a company already believes. Right now the two beliefs are: *AI agents are going to run real parts of our business*, and *somebody, somewhere, is presumably making sure that goes well*. The first one is funded. The second one is mostly a shrug.

KPMG put numbers on the shrug. Their Q2 2026 Global AI Pulse surveyed 2,145 senior leaders across 20 countries, and the headline is not that companies are backing off. Seventy-nine percent say AI would stay a priority even in a recession. Planned spending is holding steady. What has changed is the expectation: leaders now want accountability, cost discipline, and evidence that any of this is working.

And here is the part worth reading twice. Three-quarters of organizations say their CEO actively owns AI as a strategic priority. But only 24 percent can name the CEO or executive committee as *ultimately accountable* for decisions made using AI outputs. Twenty-nine percent say accountability is shared across multiple roles or teams. Another 13 percent say it is unclear or undefined.

Translated out of survey language: a very large amount of AI work currently has no owner. That is not a crisis. That is an open lane.

3x

more likely to report
established ROI where
accountability is clear

5x

more likely to report ROI
with full visibility into AI
operating costs

49%

have delayed or scaled
back agent rollouts
on cost

35%

have "very clear" rules
for when a human
overrides AI

Sit with that last one for a second. Sixty-five percent of organizations deploying AI do not have a clear answer to the question "when is a person supposed to step in?" If you write that document for your team, you have not just written a document. You have become the person who owns the question.

An AI agent manager is not the person who builds the agent. It is the person who makes the agent trustworthy enough to run inside a real business.

That distinction is the whole opportunity. Building agents is an engineering problem, and engineers are already on it. Making an agent safe to point at accounting, HR, customer support, or a client inbox is a *process* problem. It needs someone who can define a job, set boundaries, design approvals, run tests, watch the cost, and know when to pull the plug. If you have worked in operations, project management, compliance, finance, HR, support, or product, you already own most of those muscles. What you are missing is vocabulary, a handful of frameworks, and proof.

This guide is how you get all three in 90 days.

WHAT'S IN HERE

- 01 Why this role exists now. The accountability gap, in numbers, and why agents made it urgent.
- 02 What the job actually is. Five responsibilities that fit on one page.
- 03 Your transferable skills. What your current job already taught you, translated.
- 04 The literacy floor. The eleven concepts you need, and the ones you can skip.
- 05 The agent charter. Four questions every agent must answer in writing.
- 06 Governance, override, and risk. Including the human-override playbook nobody has written.
- 07 Cost, measurement, and ROI. The money skill that is currently a named gap.
- 08 Security for non-engineers. Seven controls you can specify without writing code.
- 09 The portfolio. Six artifacts, one fictional case study, zero NDA problems.
- 10 The 90-day plan. Week by week, with a deliverable attached to each one.
- 11 Titles, résumé, interviews. What to search for and how to talk about it.
- 12 The resource list. Real links, honestly annotated, mostly free.

One honest positioning note before we start. **"AI agent manager" is not yet a standardized job title.** You will rarely see it printed on a req. It is an umbrella for a cluster of roles that are being posted right now under other names: AI operations, AI governance, responsible AI, automation program management, AI product operations, agent lifecycle management. Treat the phrase as a description of the work, and treat section eleven as the list of words you actually type into a search box.

Agents broke the thing that made chatbots safe.

A chatbot that is wrong produces a bad sentence. An agent that is wrong produces a bad action. Those are not the same category of problem, and most governance was written for the first one.

For two years the risk conversation about AI was fundamentally about text. Hallucinations, tone, bias, citations. Annoying, sometimes serious, but bounded: a human read the output before anything happened. The human was the last mile, and the human was free.

Agents removed the last mile. An agent takes multi-step actions using tools and permissions you gave it. It reads a system, decides something, writes to another system, sends a message, files a ticket, issues a refund, updates a record. The output *is* the action. And that changes the failure math completely: a bad answer in a chat window is a minor irritation, while a bad action inside accounting, payroll, security tooling, or a client email thread is an incident with a name and a meeting attached.

Companies figured this out fast. Almost half of organizations, 49 percent, report having questioned, delayed, or scaled back agent deployments once expected costs started outweighing value. That is not fear. That is a market learning that agents need management, and discovering nobody was assigned to do it.

The gap, stated plainly.

WHAT ORGANIZATIONS HAVE

WHAT THEY'RE MISSING

75% CEO ownership of AI as a priority

Only 24% can name the CEO or exec committee as ultimately accountable for AI-informed decisions

Widespread agent deployment

Only 35% have “very clear” rules for when a human should override AI output

Growing AI spend, steady budgets

Only 35% have full visibility into what their AI actually costs to run

Data feeding AI systems everywhere

Only 35% have “very clear” accountability for AI data quality and refresh cycles

Cost dashboards (53%) and cost reviews (54%)

Only 40% have usage or token budgets; only 39% have architecture or prompt design standards

Read the right-hand column again. Every line is a job description. Unclaimed, unglamorous, and directly tied to outcomes leadership has already said they care about. Organizations with clearly defined accountability report established ROI at three times the rate of those without it. Organizations with full cost visibility report it at five times the rate.

THE TELL

KPMG's own recommendation list for leaders starts with "clarify accountability, make decision rights explicit." When a Big Four firm tells 2,145 executives that their top priority is figuring out who owns this, that is career advice wearing a suit. Somebody has to be the answer to that question. It might as well be someone who volunteered.

There is a workforce signal underneath the governance one, too. Seventy-eight percent of leaders expect AI fluency to become more important, and believe roles will change for people who do not develop it. Employee adoption of AI agents rose to 28 percent this quarter. The work is arriving whether or not anyone is managing it. That is the window: adoption is ahead of governance, and it will not stay that way for long.

The uncomfortable flip side, which you should hold onto: windows close. The people who move into these roles in the next 12 to 18 months will define them. The people who move in three years will be hired into structures somebody else designed.

PART TWO · WHAT THE JOB ACTUALLY IS

Five responsibilities, and none of them require a compiler.

If you can describe the job crisply in an interview, you are already ahead of most candidates, because most candidates describe enthusiasm instead.

Microsoft's current agent lifecycle guidance treats agents the way good product teams treat products: named owners, monitoring plans, evaluation cycles, improvement loops, and explicit retirement criteria. Not one-time projects. That framing is the right one, and it collapses neatly

into five things.

01

Define the agent's job and its boundaries.

One sentence on what it does. A list of what it must never do. The scope question is where most agent programs quietly fail, because "handle customer emails" is not a job description, it is a hope. "Draft replies to shipping-status questions on orders under \$500, for a human to approve" is a job description. Narrow scope is not timidity. It is the only thing that makes evaluation, cost tracking, and blame assignment possible later.

02

Control its data, tools, permissions, and actions.

What can it read? What can it write? Which systems, which records, which fields, under whose identity? This is access management applied to a non-human actor that works fast and does not get tired. If you have ever run a permissions review or scoped a vendor integration, this is the same discipline with new vocabulary.

03

Test it before it touches anything real.

Agents need the equivalent of QA, and almost nobody is doing it properly. That means a test set of real cases with known-correct answers, a scorecard, a pass threshold, and a documented result. It also means deliberately trying to break it: weird inputs, missing data, hostile instructions buried in a document it reads. "We tried it a few times and it seemed good" is the current state of the art at most companies. Beating that is not hard.

04

Monitor quality, cost, and business value in production.

Accuracy drifts. Data goes stale. Costs climb with usage in ways nobody forecast. Somebody has to watch three dials at once and know what each one means. This is the responsibility with the shortest supply of qualified people, which is exactly why section seven exists.

Improve it, narrow it, pause it, or retire it.

Agents are not monuments. The mature move is having pre-agreed thresholds that trigger a decision: accuracy below X, cost above Y, escalation rate above Z, and we revisit. Almost no one sets these in advance, which is why the 49 percent who scaled back deployments mostly did it in a panic rather than on a schedule.

HOW TO SAY IT IN AN INTERVIEW

“I make sure an agent has a defined job, controlled access, a test it has to pass, numbers we watch after launch, and a decision point where we improve it or turn it off.”

Twenty-nine words. Delivered without notes, it separates you from every candidate who opens with “I’m really passionate about AI.”

PART THREE · YOUR TRANSFERABLE SKILLS

You are probably 60% qualified and 0% fluent.

That combination feels like being unqualified. It is not. Fluency is the fast part, and it is the part this guide fixes.

Find your row. The left column is what you have done. The right column is the same skill, described the way a hiring manager for an AI operations role would recognize it.

IF YOU COME FROM	YOU ALREADY DO THE AGENT-MANAGEMENT VERSION OF
Operations	Workflow decomposition and process controls. You know how to break a messy process into steps, find the handoffs, and spot where it breaks. That is 80% of scoping an agent.
Project or program management	Ownership, approval gates, stakeholder coordination, status reporting. Agent lifecycle management is program management with a non-human team member.

Compliance, legal, or audit	Policy writing, documentation trails, risk assessment, evidence. You are natively good at the part everyone else finds tedious and regulators find mandatory.
Finance or FP&A	Unit economics, budgets, variance analysis, ROI cases. Token pricing is just a new unit cost, and one-third of leaders named cost literacy as a blocker to deploying agents.
HR or L&D	Adoption, training, change management, role redesign, and the human oversight question. Somebody has to teach people how to work alongside these things.
Customer support or success	Escalation paths, QA scoring, tone standards, incident handling. You have run the human version of exactly the review loop an agent needs.
Product management	Requirements, success metrics, testing, iteration, sunseting. Microsoft's lifecycle model is essentially product management applied to agents.
IT, security, or data	Least privilege, identity, data classification, logging, incident response. You are closest to the technical end and should lean into risk tiering.
Marketing or comms	Brand safety, approval workflows, review chains, factual accuracy under deadline. Content agents are the most-deployed and least-governed category out there.

The World Economic Forum's Future of Jobs research keeps landing on the same point: analytical thinking, leadership, and collaboration stay critical alongside growing technical AI skills. The skills that transfer are the ones that were always about judgment.

Coding is genuinely useful here and I am not going to pretend otherwise. If you can read a bit of Python or SQL, you will be faster and more credible. But it is a multiplier, not a prerequisite. The bottleneck in most organizations right now is not people who can build agents. It is people who can decide what an agent should be allowed to do, and defend that decision to a room.

ONE HONEST CAVEAT

"Transferable" does not mean "automatic." Nobody is going to look at your ops résumé and infer the AI part. You have to build the bridge yourself, in artifacts, which is what sections nine and ten are for. The audit above tells you the bridge is short. It does not build it.

Eleven concepts. About two weekends.

You need enough fluency to ask sharp questions and call nonsense when you hear it.

You do not need to be able to build the thing.

CONCEPT 01

What an agent is, versus a chatbot, versus automation.

A chatbot answers. Traditional automation follows a fixed script you wrote in advance. An agent is given a goal and a set of tools, and it decides the steps itself, in a loop, until it thinks it is done. The word that matters is **decides**. That is where the value and the risk both come from, and it is why the governance you would apply to a macro is not sufficient here.

CONCEPT 02

The anatomy: model, prompt, tools, knowledge, memory.

Five parts, and you should be able to name all five. The **model** is the reasoning engine. The **system prompt** is its standing instructions and boundaries. **Tools** (or connectors, or function calls) are the actions it can take in other systems. **Knowledge sources** are the documents and data it can look things up in. **Memory** is what it carries between steps or sessions. Every governance question you will ever ask maps onto one of those five boxes.

CONCEPT 03

Tokens, and why the bill is variable.

Models charge per token, roughly a word fragment, for both what goes in and what comes out. Output tokens usually cost several times more than input tokens. Agents are expensive because a single “task” might be dozens of model calls, each carrying the whole conversation history forward, so cost grows faster than task count. This is the single highest-leverage concept in the guide, and it gets its own section.

CONCEPT 04

Model choice is now a budget decision.

Access to lower-cost, high-fidelity models was the fastest-rising strategic influence in the KPMG data, moving from 15 to 22 percent in a single quarter. Frontier models can cost many times what a small fast model costs for the same job. Knowing which tier a task actually

needs, and being willing to test the cheaper one, is a money skill that currently reads as a technical one.

CONCEPT 05

Failure modes you must be able to name.

- **Hallucination.** Confident invention. Worse in agents, because it becomes an action.
- **Stale or wrong grounding.** The agent is right about a document that is out of date.
- **Prompt injection.** Malicious instructions hidden in content the agent reads.
- **Scope creep.** The agent solves a nearby problem you never approved.
- **Silent degradation.** Quality drifts slowly and nobody notices because nobody is measuring.
- **Cascading error.** Step three was wrong, so steps four through nine are confidently wrong.

CONCEPT 06

Non-determinism, and what it does to testing.

The same input can produce different output twice. That breaks the pass/fail testing instinct most people bring from software or process work. You test agents statistically: run 50 cases, score them, track the rate. "It worked when I tried it" is meaningless. A 92 percent pass rate across a documented test set is an actual claim.

CONCEPT 07

Identity and permissions for non-humans.

When an agent acts, whose account is it using? If it runs as a shared service account with broad access, it can see and do more than any individual user should, and your audit log says "service account" instead of a name. Ask that question in any meeting about an agent and watch the room go quiet.

CONCEPT 08

Evaluation, or "evals."

The industry word for structured testing. A set of inputs, expected outputs or grading criteria, and a score. Evals can be automated, human-graded, or model-graded (one model scoring another's work, which is cheap, fast, and needs a human spot-check). Learning to design a good eval set is arguably the single most marketable non-coding skill in this whole field.

CONCEPT 09

Observability: traces and logs.

A **trace** is the full record of what the agent did on one task: every step, every tool call, every token, every cost. Without traces you cannot debug, price, or audit anything. When you evaluate a platform, "can I see a full trace of a single run?" is the question that matters most.

CONCEPT 10

Human in the loop, on the loop, out of the loop.

Three postures. **In the loop**: a human approves every action before it happens. **On the loop**: the agent acts, a human monitors and can intervene. **Out of the loop**: fully autonomous, reviewed after the fact or by sampling. Choosing the right posture per action, not per agent, is a craft, and it is the heart of the override playbook in section six.

CONCEPT 11

Fluency is not coding.

You are aiming to be the person who can sit between a business owner and an engineer and make both of them more effective. That means you can translate a fuzzy business process into explicit instructions and controls, and you can read a technical answer well enough to know whether it addressed your question. Build toward that. If you find yourself trying to learn transformer architecture, you have wandered off.

A QUICK SELF-TEST

Can you explain, to a colleague, in under two minutes, why an agent that summarizes 200 support tickets might cost forty times more than one that summarizes five, even though both "do a summary"? If yes, you have concept three. If not, that is your first evening.

Four questions. One page. Enormous leverage.

This is the artifact to build first, because it is the one that makes you useful in a meeting before you have any credentials at all.

Every agent running in a business should have written answers to four questions. Most have zero. Walking into a room with a filled-in charter for an agent your company already runs is the fastest credibility move available to you, and it costs an afternoon.

—— THE AGENT CHARTER

01 · WHO OWNS THIS AGENT?

A named human, not a team. Plus a named backup, and the executive this rolls up to. If the answer takes more than five seconds, you have found the problem. Twenty-nine percent of organizations report accountability spread across multiple teams, which is functionally the same as nobody.

02 · WHAT JOB DOES IT DO, AND WHAT DOES IT TOUCH?

One sentence of scope. Then the inventory: which systems it can read, which it can write to, which actions it can take, whose identity it runs as, and what data classification it handles. Include an explicit “never does this” list. The never-list is usually more useful than the scope statement.

03 · HOW DO WE KNOW IT’S HELPING?

The baseline before the agent existed. The two or three metrics you now track. The review cadence and who attends. And the cost, per task and per month, sitting right next to the benefit so the comparison is unavoidable.

04 · WHAT HAPPENS WHEN IT’S WRONG?

How a mistake gets detected, who gets told, who can pause it, how the action gets reversed, how the customer or employee gets made whole, and what changes afterward so it does not recur. If you cannot describe the rollback, the agent is not ready for production.

Notice what this document quietly does. It forces a named owner into existence, makes access explicit, puts cost next to value, and pre-commits an incident path. Those are exactly the four things the KPMG data says organizations are missing, and it is one page.

Make this your calling card. Write it once as a blank template, then fill it in for a real or fictional agent. You now have something to show people, which is more than most candidates in this space have.

The unglamorous half, which is where the leverage is.

Only about one-third of organizations describe their governance roles and processes as very clear and well managed. This is a field with plenty of principles and very little practice.

You do not need to invent any of this. Two frameworks cover nearly everything, and knowing them by name is itself a hiring signal.

NIST AI RMF: govern, map, measure, manage.

The US National Institute of Standards and Technology publishes an AI Risk Management Framework built on four functions. It is free, readable in an afternoon, vendor-neutral, and increasingly the shared vocabulary in job descriptions.

- **Govern.** Culture, policies, roles, accountability. The layer everything else sits on.
- **Map.** Understand context: what the system does, who it affects, what could go wrong.
- **Measure.** Analyze and track risk with actual metrics and testing.
- **Manage.** Prioritize, respond, monitor, and act on what you measured.

Learn these four words well enough to structure a conversation around them. NIST also publishes a companion Playbook with concrete suggested actions per function, and a Generative AI Profile that maps the framework onto generative and agentic systems specifically.

ISO/IEC 42001: the certifiable management system.

Where NIST gives you a way of thinking, ISO/IEC 42001 gives you an auditable management system for AI: policies, defined responsibilities, risk assessment, monitoring, internal audit, and continuous improvement. If you have worked anywhere near ISO 27001 or SOC 2, the shape is instantly familiar. Enterprises pursuing certification need people who can run the program, and there are not many of them.

You do not need to memorize the standard. You need to know what it is, why a company would pursue it, and roughly what conformance requires. That is enough to be the person in the room who has heard of it.

Risk tiering, which is the practical core.

Not every agent deserves the same scrutiny, and pretending otherwise is how governance programs die. Tier agents by what they can access and what they can change:

TIER	LOOKS LIKE	CONTROLS
Low	Read-only, internal, non-sensitive. Drafts and summaries a person reviews anyway.	Named owner, basic logging, quarterly spot check.
Medium	Writes to internal systems, or touches employee or business-confidential data. No external exposure.	Full charter, eval set with a pass threshold, human approval on writes, monthly review, cost budget.
High	Customer-facing, money-moving, or touching regulated, personal, or safety-relevant data.	Everything above, plus pre-deployment sign-off, human in the loop on every consequential action, full trace retention, tested rollback, incident runbook, scheduled access reviews.

Build this table for your own organization and you have produced something most companies do not have. It is also a beautiful interview artifact, because it demonstrates proportional thinking rather than blanket caution.

The human-override playbook.

This is the highest-value document in the guide, precisely because only 35 percent of organizations have very clear guidance here. Six questions, answered in writing, per agent:

1. **What can it decide alone?** List the actions that need no approval. Be specific and be generous where the blast radius is genuinely small, because an override policy that stops everything gets ignored within a week.
2. **What always needs approval?** Anything involving money above a threshold, anything a customer sees, anything touching personal or regulated data, anything irreversible. Name the approver by role.
3. **What auto-escalates?** Define the triggers, not just the vibes. Low confidence, an unrecognized input, a repeat contact from the same customer, a value outside normal range, a tool failure, more than N steps taken.

4. **Who can pause it, and how fast?** Name the people and describe the mechanism. A kill switch that requires a ticket and an engineer is not a kill switch. Test it before you claim to have one.
5. **How do people report bad behavior?** One channel, publicly known, that a frontline employee can use in ten seconds without needing permission. Most bad agent behavior is spotted by someone junior who assumes it is not their place to say anything.
6. **What happens after?** Who reviews, what gets logged, what gets fixed, who tells the affected person, and how the fix gets verified. Incidents without a written aftermath repeat.

THE DESIGN RULE THAT MATTERS

Set override thresholds per *action*, not per agent. One agent might be free to draft and categorize all day while requiring approval for a single kind of write. Governing at the agent level forces you to choose between too much friction and too much risk, and you will end up with both.

The rest of the governance kit.

- **An agent inventory.** A living list of every agent, its owner, its tier, and its status. Astonishingly few companies have one. Proposing to build it is a great first internal move.
- **Approved and prohibited data sources.** What the agent may ground itself in, and what is off limits, in writing.
- **Audit logs and transcript retention.** What gets kept, for how long, who can read it, and how it satisfies your existing records policy.
- **Change control.** Prompt changes and model swaps are production changes. Treat a model version upgrade like a deployment, because it is one.
- **Scheduled access and policy reviews.** Quarterly for high tier. Permissions granted for a pilot have a way of becoming permanent.
- **A regulatory watch item.** If you operate in or sell into the EU, the AI Act's obligations phase in on a published timeline. You do not need to be a lawyer. You need to be the person who raised it first.

The money skill nobody has, named in a survey.

One-third of leaders cite AI cost and economic literacy as a challenge to deploying agents. That is a stated skill gap from 2,145 executives. Go be the person who has it.

Here is the situation in three numbers. Only 35 percent of organizations have full visibility into what their AI costs to run. Forty-two percent describe that visibility as “somewhat.” Thirteen percent find out when the bill arrives. And organizations with full cost visibility report established ROI at five times the rate of those without it, 15 percent versus 3 percent.

Knowing what your tools cost turns out to be a competitive advantage. I find this genuinely funny, and also the most actionable fact in the entire report.

Why agent costs surprise people.

The mental model most people carry over from SaaS is per-seat, predictable, and flat. Agent economics are none of those things. Four things drive the surprise:

- **Usage-based pricing.** You pay per token, so cost scales with how much work gets done, not how many people are licensed.
- **Loops multiply calls.** One user request can trigger many model calls as the agent plans, calls tools, reads results, and revises.
- **Context accumulates.** Each step often resends the growing conversation, so later steps cost more than earlier ones. Long-running agents get expensive at an accelerating rate.
- **Retries and failures still bill.** An agent that fails a task three times before succeeding charged you four times.

None of this is a reason not to use agents. It is a reason to know your cost per completed task before you scale from ten a day to ten thousand.

The measurement set that actually convinces people.

METRIC	WHAT IT TELLS YOU, AND THE TRAP
Baseline	What the process cost in time, money, and error rate <i>before</i> the agent. Capture it first. Without a baseline every later claim is unfalsifiable, and you will be asked.
Completion rate	Share of tasks the agent finishes end to end without help. The most honest headline number.
Accuracy or quality score	From your eval set, sampled continuously in production. Say how it is graded and by whom.
Escalation rate	How often a human has to step in. Rising escalation is the earliest warning sign you will get.
Rework rate	How often the output needs fixing afterward. This is the number that separates real savings from moved work, and almost nobody tracks it.
Cost per completed task	Total spend divided by successful completions. Not per call, per <i>completion</i> , because failures cost money too.
Time saved	Useful, and the most abused metric in the field. Twenty minutes saved is only value if it went somewhere. Say where.
Cost delta versus the old process	The agent's run cost plus oversight time versus the baseline. Oversight time is a real cost. Include it or someone else will.

The four controls, and the two nobody has.

Companies are watching the spend without changing the behavior. Cost reviews (54 percent) and dashboards (53 percent) are the common controls. Usage or token budgets (40 percent) and architecture or prompt design standards (39 percent) lag behind. That is the exact pattern of someone who checks their bank balance frequently and has no budget, which, respectfully, is most people's personal finances.

The full set, in the order I would build them:

1. **Visibility.** A dashboard showing spend by agent, by team, by day. You cannot manage what you see monthly and in aggregate.
2. **Review gates.** A cost estimate required in the approval process, before a single line is built. Force the forecast.
3. **Budgets and limits.** A hard token or spend cap per agent, with an alert at 80 percent. This is the one most organizations skip and most regret.
4. **Design standards.** Rules about which model tier fits which task, how much context to carry, when to cache, when to stop retrying. This is where the real savings live and where the fewest people are working.

A MOVE YOU CAN MAKE IN ONE WEEK

Pick one agent or AI workflow at your company. Find out what it cost last month and what it produced. Write both numbers on one page with a per-task figure. In most organizations, nobody has done this, and the act of doing it makes you the person who owns AI cost visibility. That is not a metaphor. That is how these roles actually get created.

PART EIGHT · SECURITY FOR NON-ENGINEERS

Seven controls you can specify without writing code.

You are not going to be the security engineer. You are going to be the person who insists the questions get asked and written down, which is a different job and currently a vacant one.

Autonomous, multi-step, tool-using systems create risks that traditional application security was not designed for. OWASP's Agentic Security Initiative is the best free starting point for understanding the emerging threat model. Here is the practical floor.

01

Least privilege, enforced and reviewed.

The agent gets the narrowest access that lets it do its defined job, and nothing held "just in case." Write down what it has, and re-check quarterly. Pilot-era permissions are the most common source of quiet over-access in any organization.

02

Know every connector and data source.

Maintain the list. Each connector is a door in both directions: data the agent can reach, and a system it can affect. If nobody can produce this list on request, that is your finding.

03

Separation between users, customers, and tenants.

Can the agent, while helping customer A, surface something belonging to customer B? Can it show an employee data their own login would not permit? Agents inherit whatever access they were given, not the access of the person asking, and that mismatch is where the ugliest incidents come from.

04

Treat retrieved content as untrusted.

Prompt injection is the defining new risk: instructions hidden inside a document, email, web page, or support ticket that the agent reads and obeys. The mitigation is architectural (the agent should not act on instructions found in data), but the governance move is yours: for any agent that reads external content, require that the injection question was explicitly considered and the answer written down.

05

Classify and constrain sensitive data.

Which categories may this agent touch, where does that data go, is it retained by a vendor, is it used for training, and does that satisfy your existing policy? These are the questions your privacy team already asks about vendors. Bring them to agents.

06

Log the consequential actions.

Every write, send, payment, permission change, or deletion, with a timestamp, the triggering input, and the identity used. If you cannot reconstruct what happened after an incident, you cannot fix it and you cannot prove anything to anyone.

Rehearse the shutdown.

Actually pause an agent in a test window and time it. Then walk through reversing an action it took. A rollback plan that has never been executed is a paragraph, not a control, and you will find out which one it is at the worst possible moment.

PART NINE · THE PORTFOLIO

Six artifacts. This is the part that gets you hired.

Nobody can verify that you understand agent governance from a résumé bullet. They can verify it in about ninety seconds from a well-made document.

Everything in this section can be built without writing code, without your employer's data, and without anyone's permission. Build them around a fictional but realistic company, which solves the confidentiality problem completely and lets you make the example as clean as you want.

THE WORKED EXAMPLE I'D USE

Meridian Freight, a mid-size logistics company, 400 employees. Their accounts payable team manually triages about 600 supplier invoice exceptions a month: mismatched amounts, missing purchase order numbers, duplicate submissions. Average 11 minutes each. The proposed agent reads the invoice and the PO record, classifies the exception, drafts a resolution, and routes it for human approval. It never pays anything. It never emails a supplier directly.

Boring, specific, plausible, and touching money without moving it. That last part is what makes it a great portfolio subject: it is genuinely medium-risk, so all your controls have a reason to exist.

The six.

1. **The agent charter.** One page, the four questions from section five, fully answered for Meridian. This is your lead artifact.
2. **The workflow and approval map.** The current process, then the proposed process, with decision points marked: agent decides, agent proposes and human approves, or auto-escalate. A simple diagram or a clean table both work. The point is showing you can see a process, not that you can use a drawing tool.

3. **The risk-tier assessment.** Which tier, why, and the specific controls that tier requires. Include the two or three risks you rated as most serious and what you did about each. Reference NIST's four functions by name.
4. **The evaluation scorecard.** Thirty to fifty test cases with expected outcomes, your grading criteria, your pass threshold, and the results of an actual run. Include the failures. Especially include the failures.
5. **The cost and value model.** Baseline cost of the manual process, estimated agent run cost per task, oversight cost, and the break-even point. One page, real arithmetic, stated assumptions. This is the artifact hiring managers remember, because almost nobody brings one.
6. **The incident and rollback playbook.** Detection, notification, pause, reversal, communication, and the post-incident change. Two pages maximum. Written so a stranger could follow it at 2am.

Then actually build something small.

The artifacts are the proof of judgment. A working agent, however humble, is the proof you have touched the material. Use a no-code platform, pick a genuinely low-risk workflow from your own life or work, build it, and then, critically, **run your own eval set against it and write up what broke.**

The write-up matters more than the agent. "I built a thing and it worked" is a hobby. "I built a thing, tested it on 40 cases, found it failed on a specific category of input, narrowed the scope, retested, and here is the cost per run" is the job. Failures documented honestly are the strongest signal in your entire portfolio, because they prove you were actually measuring.

PART TEN · THE 90-DAY PLAN

Thirteen weeks, five to seven hours each.

Every week has one deliverable. If you finish the week without producing the artifact, you did not finish the week. Reading does not count.

This assumes you have a job and a life, so it is built for roughly one hour on four weekday evenings and a two-hour block on a weekend. Slip a week if you need to. Do not skip the deliverables, because the deliverables *are* the pivot. The learning is just what makes

them possible.

DAYS 1 TO 30 · LEARN GOAL: VOCABULARY, FRAMEWORKS, AND ONE MAPPED WORKFLOW

WEEK 01

Get fluent in the anatomy.

Work through section four until all eleven concepts are yours. Read Anthropic's "Building effective agents" essay, which is short, vendor-honest, and the single best explanation of when an agent is and is not the right tool. Then go hands-on: spend two hours actually using an agentic tool, watching what it does step by step rather than just reading its answer.

DELIVERABLE A one-page glossary in your own words. Model, prompt, tool, knowledge, memory, token, eval, trace, human in the loop. Writing definitions yourself is the test of whether you have them.

WEEK 02

Learn the governance frameworks by name.

Read the NIST AI Risk Management Framework and skim its Playbook. Understand govern, map, measure, manage well enough to structure a conversation around them. Read a summary of ISO/IEC 42001 and understand what certification involves. Skim OWASP's agentic security material for the threat vocabulary.

DELIVERABLE A two-page brief: what each framework is, who it is for, and how you would use it. This becomes an interview answer and, later, an internal memo.

WEEK 03

Learn the money.

Read the pricing pages of two or three major model providers until token pricing is intuitive. Compare model tiers on a price-and-quality comparison site. Do the arithmetic on a hypothetical workload: 500 tasks a day, eight model calls each, roughly this much context, what does that cost per month at three different model tiers? Get the estimate wrong, then figure out why.

DELIVERABLE A reusable cost estimator, in a spreadsheet. Inputs at the top, assumptions stated, output as cost per task and cost per month. This is a portfolio artifact and a genuinely useful tool.

WEEK 04

Map one real workflow from your own field.

Pick a process you already understand deeply. Document it honestly: every step, every handoff, every decision point, how long each part takes, how often it goes wrong, and what happens when it does. Then mark which steps an agent could plausibly take and which absolutely need a person. Your domain knowledge is the advantage here, so use a process you know better than a consultant would.

DELIVERABLE A workflow map with a baseline: time per instance, volume, error rate, cost. You now have the “before” half of every claim you will make later.

DAYS 31 TO 60 · BUILD

GOAL: SIX ARTIFACTS AND ONE WORKING AGENT YOU HAVE BROKEN ON PURPOSE

WEEK 05

Write the charter and the approval map.

Using your week-four workflow, or the Meridian example if your own work is too sensitive, produce artifacts one and two. Answer all four charter questions completely, including the uncomfortable ones. Mark every decision point in the approval map as decide, propose, or escalate.

DELIVERABLE Agent charter (1 page) and workflow and approval map (1 to 2 pages).

WEEK 06

Build the thing.

Use a no-code agent builder. Microsoft Copilot Studio, Zapier Agents, n8n, or a custom assistant in Claude or ChatGPT with documents and instructions attached. Keep the scope small enough to finish. Expect the first version to be disappointing, because it will be, and that disappointment is the curriculum.

DELIVERABLE A working agent, plus a short build log: what you tried, what did not work, what you changed.

WEEK 07

Test it properly and write the risk tier.

Build 30 to 50 test cases with known-correct answers, including the awkward edge cases and at least three deliberately hostile inputs. Define your grading criteria before you run it, because defining them after is how you fool yourself. Run the set, score it, log every failure with a category. Then write the risk-tier assessment: which tier, why, and which controls follow.

DELIVERABLE Evaluation scorecard with real results, and a risk-tier assessment referencing NIST's four functions.

WEEK 08

Price it, and plan for it going wrong.

Fill in your week-three estimator with actual numbers from your week-six agent. Compare against the week-four baseline, including oversight time. Find the break-even volume. Then write the incident and rollback playbook: detection, notification, pause, reversal, communication, follow-up. Rehearse the pause step for real.

DELIVERABLE Cost and value model with break-even, and the incident and rollback playbook. Portfolio complete: all six artifacts.

DAYS 61 TO 90 · POSITION

GOAL: BE VISIBLY THE PERSON WHO DOES THIS, INTERNALLY AND EXTERNALLY

WEEK 09

Write the case study and publish it.

Eight hundred to twelve hundred words. Structure: the process and its baseline, what you built, what broke during evaluation, what you changed, what it costs, and what you would require before running it for real. Lead with the failures. Publish it somewhere public: LinkedIn, a personal site, a newsletter. Public is the whole point, because it converts a private project into a thing people can find and forward.

DELIVERABLE A published case study with a stable link you can put in an application.

WEEK 10

Rewrite the résumé and the headline.

Rebuild your bullets around ownership and outcomes, not tools. Every AI-adjacent line should contain a scope, an action, and a number. Rewrite your LinkedIn headline to state the role you want alongside the domain you have. "Operations manager" becomes something closer to "Operations and AI agent governance, supply chain." Add the case study link to your profile.

DELIVERABLE A rewritten résumé, an updated headline, and a two-sentence positioning statement you can say out loud without cringing.

WEEK 11

Make the internal move.

This is the highest-probability path by a wide margin, and most people skip it because it feels less exciting than applying somewhere new. Propose one concrete thing to your leadership: an inventory of the AI tools and agents already running, with owners and risk tiers. It is small, cheap, obviously sensible, and impossible to argue against. Bring your charter template as the format. If an inventory does not fit, propose the override policy instead, since only 35 percent of organizations have a clear one.

DELIVERABLE A one-page proposal, sent to a named person, with a specific ask and a date attached.

WEEK 12

Go find the people who do this.

Search the titles in section eleven and save ten real job postings. Read them for the vocabulary they use and adjust your language to match theirs. Then reach out to eight people currently doing the work, at companies you find interesting, with a specific question about how they handle something you actually built. Not "can I pick your brain." Something like: "I built an eval set for an invoice-triage agent and struggled to grade partial correctness. How does your team handle that?" People answer that email.

DELIVERABLE Ten saved postings, eight sent messages, and at least three applications submitted.

Prepare the stories, then keep going.

Prepare four stories, each two minutes: a process you redesigned, a risk you caught before it landed, a number you owned and moved, and the failure mode you found in your own agent and the control you designed in response. That last one is the story that wins interviews, because it demonstrates exactly the judgment the role exists to supply. Bring printed artifacts. Offer them.

DELIVERABLE Four rehearsed stories and a printed portfolio. Then start month four, because the pivot is a direction, not a finish line.

IF YOU ONLY HAVE THREE HOURS A WEEK

Do weeks 1, 2, 5, 7, 9, and 11, in that order, over about five months. That sequence gets you the vocabulary, the frameworks, the charter, a real eval, a published case study, and an internal proposal. Everything else is depth. Those six are the spine.

PART ELEVEN · TITLES, RÉSUMÉ, INTERVIEWS

What to type in the search box, and what to say in the room.

Remember: the job exists, the title does not. Search for the work, not the phrase.

Search these terms.

- **Governance and risk:** AI Governance Analyst · Responsible AI Program Manager · AI Risk and Controls Manager · AI Compliance Manager · AI Assurance Lead
- **Operations and lifecycle:** AI Operations Manager · Agent Lifecycle Manager · AI Product Operations Manager · Automation Program Manager · AI Delivery Manager
- **Enablement and adoption:** AI Center of Excellence Program Manager · AI Adoption Lead · AI Enablement Manager · AI Transformation Manager
- **Adjacent and underrated:** AI Quality Analyst · Human-in-the-loop Operations Lead · AI Evaluation Specialist · AI FinOps Analyst

Set alerts on all four clusters. Also search for the *work* rather than the title: try phrases like "agent evaluation," "AI governance framework," "human in the loop," and "LLM cost" in job search text, because plenty of relevant roles are titled something forgettable and describe exactly this in the third bullet.

Rewriting your bullets.

INSTEAD OF	WRITE
"Used AI tools to improve team productivity"	"Defined scope, approval thresholds, and escalation rules for an AI-assisted invoice triage workflow handling 600 cases a month"
"Familiar with AI governance"	"Built a three-tier risk classification for AI use cases mapped to NIST AI RMF functions, with tier-specific controls"
"Managed vendor relationships"	"Ran access and data-handling reviews for third-party AI tools, including connector inventory and retention terms"
"Interested in AI and its business applications"	"Built and documented a cost model for agent workloads, establishing cost per completed task and break-even volume"
"Tested new AI features"	"Designed a 45-case evaluation set with defined grading criteria; identified a failure category affecting 12% of inputs and narrowed agent scope in response"

The pattern in the right column is always the same: a specific scope, a concrete action, and a number. Generic AI enthusiasm on a résumé now reads as a negative signal, because everyone has it and it distinguishes no one.

The four questions you will be asked.

- **"How would you decide whether an agent is safe to deploy?"** Answer with tiering, evals with a threshold, override design, and rollback. Not with a feeling.
- **"How would you measure whether it's working?"** Baseline first. Then completion, accuracy, escalation, rework, cost per completed task. Mention that time saved is only value if you can say where it went.

- **“Tell me about a time something went wrong.”** Use your own agent’s failure. Detection, root cause, control designed, retest. This is the story that separates you.
- **“You’re not technical. How would you work with engineering?”** Do not apologize. Say that your job is to arrive with the business process decomposed, the boundaries defined, and the success criteria written, so engineering spends its time building rather than interviewing stakeholders. That is a genuinely valuable division of labor and everyone in the room knows it.

PART TWELVE · THE RESOURCE LIST

Real links, honestly annotated.

Mostly free. Where something costs money, I have said so, and none of it is required.

FRAMEWORKS AND STANDARDS

NIST AI Risk Management Framework FREE

The govern, map, measure, manage structure, plus a companion Playbook with concrete suggested actions and a Generative AI Profile. Start here. It is the vocabulary the field is standardizing on.

nist.gov/itl/ai-risk-management-framework · airc.nist.gov

ISO/IEC 42001, AI management systems STANDARD IS PAID, SUMMARIES ARE FREE

The certifiable management-system approach. Read the free overview page first. Buy the standard only if your organization is actually pursuing conformance.

iso.org/standard/42001

OWASP GenAI Security Project and Agentic Security Initiative FREE

The best open resource on agent-specific threats, including prompt injection, excessive agency, and tool misuse. Their Top 10 for LLM Applications is the fastest way to sound informed about failure modes.

genai.owasp.org · [Agentic Security Initiative](#)

Google Secure AI Framework (SAIF) FREE

A practical, well-organized security framing with a self-assessment tool. Useful complement to OWASP if you want something more checklist-shaped.

saif.google

EU AI Act explorer FREE

A readable, searchable version of the regulation with implementation timelines. Relevant if you operate in or sell into the EU, and worth an hour regardless because it shapes how everyone else writes policy.

artificialintelligenceact.eu

— UNDERSTANDING AGENTS

Anthropic, “Building effective agents” FREE

Short, clear, and unusually honest about when you should *not* build an agent. The best single thing to read in week one.

anthropic.com/engineering/building-effective-agents

Microsoft Learn, agent lifecycle and Center of Excellence guidance FREE

Treats agents as ongoing products with owners, monitoring, evaluation, improvement, and retirement criteria. The closest thing to an official job description for the role you are pursuing.

learn.microsoft.com/en-us/agents/center-of-excellence/agent-lifecycle

Model Context Protocol documentation FREE

The open standard for connecting agents to tools and data. You do not need to implement it. You need to know what it is, because connectors are where your access-control questions land.

modelcontextprotocol.io

— COST AND ECONOMICS

Provider pricing pages FREE

Read the actual pricing pages for Anthropic, OpenAI, and Google. Note the input versus output split, the tier differences, and any caching or batch discounts. An hour here makes you literate in the thing one-third of leaders named as a blocker.

Artificial Analysis FREE

Independent benchmarking of models on quality, speed, and price. The fastest way to build intuition for which model tier a task actually needs, which is the core of the design-standards control only 39 percent of organizations have.

artificialanalysis.ai

FinOps Foundation FREE RESOURCES, PAID CERTIFICATION

The cloud cost-management community, now extending its framework to AI workloads. If your background is finance, this is your fastest credible bridge into the AI world.

finops.org

— EVALUATION AND OBSERVABILITY

Promptfoo OPEN SOURCE, FREE

Define test cases in a config file and run them against models and prompts. About as close to no-code evals as you will get, and building one eval set here teaches you more than a month of reading.

promptfoo.dev

Langfuse OPEN SOURCE, FREE TIER

Tracing, evaluation, and cost tracking for LLM applications. Even if you never deploy it, poking at the demo teaches you what a trace is and what production monitoring should show you.

langfuse.com

— BUILDING WITHOUT CODE

Microsoft Copilot Studio PAID, TRIAL AVAILABLE

The most enterprise-relevant no-code agent builder, which matters because it is what a lot of large employers are actually standardizing on. Highest résumé value of the options here.

n8n OPEN SOURCE, FREE SELF-HOSTED

Visual workflow automation with solid AI agent nodes. Best option if you want to see the wiring without paying for anything, and self-hosting it teaches you more than the managed alternatives.

n8n.io

Zapier Agents, or a custom assistant in Claude or ChatGPT FREE TIERS AVAILABLE

The lowest-friction way to get something running this week. Perfectly adequate for a portfolio project. The artifacts around the agent are what you are being judged on, not the platform.

DeepLearning.AI short courses MOSTLY FREE

One to two hour courses on agents, evaluation, and LLM application patterns. Practical, current, and short enough to actually finish. The evaluation-focused ones are the ones to prioritize.

deeplearning.ai/short-courses

IAPP AI Governance Professional (AIGP) PAID

The most recognized governance-specific credential right now, and it shows up by name in job postings. Worth it if you are pointing at compliance, legal, or risk roles. Not necessary for operations roles. Do the 90 days first and decide after.

iapp.org/certify/aigp

ISACA AI audit and AI security management credentials PAID

Strong fit if you are coming from audit, IT risk, or security and already hold a CISA or CISM. Redundant if you are not.

isaca.org

Microsoft AI-900 and Applied Skills LOW COST

A cheap, fast, recognizable fundamentals credential. It will not get you hired on its own, but it is a clean signal on a résumé that is otherwise light on AI, and it takes a weekend.

learn.microsoft.com/training

Stanford HAI AI Index Report FREE

The annual data reference on capability, cost, adoption, and policy. Use it when you need a defensible number in a proposal.

hai.stanford.edu/ai-index

WEF Future of Jobs Report FREE

Useful for the skills-demand argument, and for the reminder that analytical thinking, leadership, and collaboration stay critical alongside technical AI skills. Good ammunition for an internal proposal.

[weforum.org, Future of Jobs Report](https://weforum.org/Future-of-Jobs-Report)

What you should have on day 91.

If you can tick all nine, you are not pivoting toward this role. You are doing it, without the title.

—— DAY 91

01 · A TARGET ROLE, NAMED

One of the title clusters from section eleven, chosen deliberately based on where your existing experience is strongest.

02 · A TRANSFERABLE-SKILLS MAP

Your background translated into the language of the role, with the specific gaps you are still closing named honestly.

03 · ONE DOCUMENTED AGENT USE CASE

Real or fictional, with a baseline you captured before anything was built.

04 · SIX PORTFOLIO ARTIFACTS

Charter, workflow and approval map, risk tier, evaluation scorecard, cost and value model, incident playbook.

05 · ONE AGENT YOU BUILT AND BROKE

With the evaluation results and the failure log written up, including the fix and the retest.

06 · A PUBLISHED CASE STUDY

Public, linkable, leading with what went wrong rather than what went well.

07 · A REWRITTEN RÉSUMÉ AND HEADLINE

Scope, action, number in every relevant bullet. No generic enthusiasm anywhere on the page.

08 · ONE INTERNAL PROPOSAL, SENT

An agent inventory, an override policy, or a cost review. Sent to a named person with a date on it.

09 · FOUR REHEARSED STORIES

Process redesigned, risk caught, number moved, failure found and controlled. Two minutes each, no notes.

The agents are already deployed. The accountability is not.

Companies have the budget, the executive sponsorship, and the tools. What they are missing, by their own account, is someone who can say who owns this agent, what it is allowed to do, how we know it is working, what it costs, and what happens when it is wrong. Twenty-nine percent describe that ownership as shared across teams. Thirteen percent say it is undefined. Only 35 percent have clear override rules, and only 35 percent know what any of it costs to run.

You do not need to become an engineer to fill that gap. You need enough fluency to ask the right questions, two frameworks you can name, a cost model you can defend, and six documents that prove you have actually done the thinking. That is a quarter of deliberate evenings, not a career restart.

The people who move first will define what this role means. Everyone else will interview for a job description somebody else wrote. Raise your hand.

ONE MORE THING

If you want a second set of eyes.

Building the artifacts is the easy part. Knowing whether they're any good is harder on your own.

I put this together because I keep watching capable people assume they are locked out of AI work on the grounds that they cannot code, while the roles that are actually going unfilled are the ones their existing experience prepares them for almost perfectly. The gap is real, it is documented, and it is closing.

So I am doing a small number of informal reviews for people working through this. Send me an artifact (a draft agent charter, a risk tier you are unsure about, an eval scorecard, or just a description of the workflow you are thinking of mapping) and I will give you honest, specific feedback on what I would change and what I would leave alone.

This is early and informal. No pressure either way. The playbook above is yours to run with regardless.

GET IN TOUCH

Email hi@davecto.com with the subject line **"Agent Manager Review"** and a couple of sentences about where you are in the 90 days.

More guides like this one, for people trying to use AI without embarrassing themselves.

Weekly, plain-language breakdowns on Instagram.

[@davecto](#)

Survey figures cited throughout are from the Global AI Pulse Q2 2026, KPMG International, June 2026, drawing on 2,145 senior leaders across 20 countries and territories, with Q1 comparisons from Global AI Pulse Q1 2026, April 2026. Lifecycle framing draws on Microsoft Learn's agent lifecycle guidance; risk framing on the NIST AI Risk Management Framework and ISO/IEC 42001; agent security framing on the OWASP GenAI Security Project. Skills-demand context from the World Economic Forum's Future of Jobs Report. "Meridian Freight" is a fictional company used as a worked example. Nothing here is legal or compliance advice.