

Upskilling for the agentic shift, and the governance that has to come with it

The roles are being redrawn. More than 180,000 people have gone in tracked layoff rounds this year, over half of them citing AI. The people staying are the ones who can direct an agent, check its output, and know when it is wrong. This is where to start, and what to keep in mind while you do.

THE TWO THINGS BEING TESTED

The market is drawing one line. On one side, people who can work **with** these systems. On the other, people who work around them. Closing that gap is a few weekends of deliberate practice, not a degree.

Learn to build with agents

Not prompt tricks. The actual skill: wiring an LLM to tools, giving it a task, and staying in control of what it does. Frameworks matter less than the habit of building something real and watching where it breaks.

Learn to check what it produces

The half of the job that survives automation is judgement. Reading agent output and knowing when it is confidently wrong. Upskilling that stops at "how to run it" leaves you on the wrong side of the line.

Free and cheap beats prestigious

The best agent material sits on Udemy for the price of a lunch, often with the full code free on GitHub. Vendor certifications teach you one product. Building teaches you the shape of the problem.

WHY GOVERNANCE RIDES ALONG

Upskilling to ship faster without upskilling to check is how the failures happen. The roles being removed in the current wave are often the ones where a human sat between an action and its consequence. Learn the tools, and learn where the checkpoint has to stay.

THE THEATRE GAP

Most teams reach the point where the review exists on paper. A checklist, a sign-off, a documented process. That is not the same as review that changes what ships. The distance between a control that is written down and a control that can actually halt a bad action is where most governance quietly fails.

A checkpoint you can point to in a document, but that cannot stop the thing it is meant to stop, is not a control. It is the appearance of one.

What I point people to when they ask how to learn this

These are the ones I recommend to my own team. Ed Donner is a former JP Morgan MD, so the material is built for people who have to make agents work inside a real organisation, not just a demo. Start wherever your gap is.

AI Engineer Agentic Track · the complete agent and MCP course

[START HERE](#)

Ed Donner · Udemey · six weeks · 4.7 across ~36,000 ratings

The one to name if someone asks for a single course on building agents. OpenAI Agents SDK, CrewAI, LangGraph, AutoGen and MCP, across eight hands-on projects from a career digital twin to a multi-agent research team. Project-heavy, which is the point.

Best for: anyone who can write a little Python and wants to build real agents.

AI Coder · Claude Code and coding agents

[CODING AGENTS](#)

Ed Donner · Udemey · three weeks

For using agents to write code, rather than writing code to build agents. Covers Claude Code, Cursor, Copilot and Codex through vibe coding, then engineering discipline, then multi-agent orchestration. The throughline is the responsible one: the code still has to come from you.

Best for: developers who want agents in their daily workflow without losing the plot.

LLM Engineering · core track

[FOUNDATIONS](#)

Ed Donner · Udemey · eight weeks

The deeper foundation. Eight projects across frontier and open models, RAG, QLoRA fine-tuning and agents. Take this if you want to understand what is happening underneath, not just call the API.

Best for: people who want to become an engineer here, not just a user.

Free to start

- The full course code is public at github.com/ed-donner/agents. Clone it and read before you pay.
- Anthropic's "Building effective agents" is the seminal short read. Free, and referenced throughout the courses above.

How to spend the time

- Build one thing end to end before starting the next module. Reading is not learning here.
- Break it on purpose. Watch what the agent does with a bad instruction. That is the governance lesson.

Once you can build with agents, this is how you check the output

The decisive question is not which model is underneath. It is whether a competent check exists, and whether it can halt the action. These six dimensions are the frame for asking that honestly.

01 Provenance integrity

Can you tell what the agent produced from what a human did? When the two blur, accountability goes with them.

02 Supervisory clarity

Is it clear who is answerable for the agent's output? A named owner, not a diffuse sense that someone is watching.

03 Scope definition

Is what the agent is allowed to do written down and bounded, or does it expand quietly with each new task?

04 Review substance

Is review catching real defects, or completing in seconds? This is where the theatre gap shows up first.

05 Constraint fidelity

Do the rules reach the agent at the moment it acts, or live in tribal knowledge it never sees?

06 Capability calibration

Does agent scope expand on evidence, or because someone senior saw a demo and liked it?

THE FIVE LEVELS, AND WHERE THE GAP SITS

Each dimension is scored on the same ladder. The jump that matters is the last one. Level four is documented. Level five is a control that demonstrably changes what happens at the point of decision. Most teams stall at four and call it done.

1 · Absent

No practice exists. The dimension is left to chance.

2 · Informal

It happens sometimes, on intuition. No standard.

3 · Documented

A process is written down. Followed selectively.

4 · Complete

The process is thorough and consistently applied on paper.

5 · Substantive

The control demonstrably changes behaviour where the decision is made. The theatre gap is closed.

USE IT AS A QUESTION, NOT A SCORECARD

You do not need the full index to benefit from it. Take any agent workflow you are about to trust and ask the six questions out loud. The ones you cannot answer are where the risk lives. That is the whole practice, and it is free.