

Design Your Agent Workflow

“ **One-liner:** Architect a complete multi-agent workflow for a real project — defining roles, inputs, outputs, handoffs, and a feedback loop — then test it.

? Jump in (Tinkerers start here)

Pick a real project that involves at least three distinct types of work (research, analysis, creation, review, etc.). Examples: writing a report, planning an event, developing a proposal, building a content calendar.

Design a 3-4 agent workflow on paper or in a doc. For each agent, define:

Agent Role	What it receives (input)	What it produces (output)	Handoff trigger
Agent 1: Researcher	The project brief	A structured summary of key findings	"Research complete" + summary ready
Agent 2: Drafter	Research summary + project brief	A first draft	Draft complete
Agent 3: Critic	The draft + original brief	Specific critique with improvement suggestions	Review complete
Agent 4: Editor	Draft + critique notes	Final polished output	Revisions applied

Now **implement it** using chained AI prompts. Open a chat for each agent (or reuse one chat with fresh role prompts). Run the workflow end-to-end:

Agent 1 prompt:

“ You are a **research analyst**. Your job is to gather and organize relevant information. Here is the project brief: **[paste your brief]**. Produce a structured summary of the key information I'll need. Organize it by theme. Include 3-5 key insights and any risks or gaps you see.

Take Agent 1's output and feed it to Agent 2:

Agent 2 prompt:

“ You are a **content drafter**. Your job is to turn research into a clear first draft. Here is the project brief: **[paste brief]**. Here is the research summary: **[paste Agent 1 output]**. Write a first draft that addresses the brief. Focus on clarity and completeness. Don't self-edit — that's someone else's job.

Take Agent 2's output and feed it to Agent 3:

Agent 3 prompt:

“ You are a **critical reviewer**. Your job is to find weaknesses and suggest improvements. Here is the original brief: **[paste brief]**. Here is the draft: **[paste Agent 2 output]**. Identify: (1) gaps — what's missing that the brief requires, (2) weaknesses — arguments or sections that aren't convincing, (3) specific improvement suggestions with rationale. Do NOT rewrite the draft. Just critique.

Take the draft and critique to Agent 4:

Agent 4 prompt:

“ You are a **senior editor**. Your job is to produce the final version. Here is the draft: **[paste Agent 2 output]**. Here is the review feedback: **[paste Agent 3 output]**. Revise the draft to address the critique. Maintain the original structure where it works. Explain your key changes in a brief editor's note at the end.

Feedback loop (optional): Take the final output and feed it back to Agent 3 for a second review. Notice how the quality changes with each iteration.

? Plan first (Planners start here)

Here's what you're about to do:

1. **Choose a project** — Pick something with enough complexity to benefit from specialization. A single-paragraph task won't stretch this exercise. Good candidates: a report, a strategy document, a proposal, or a content plan.

2. **Design the agent architecture** — Map out 3-4 agent roles using the table format above. Define clear inputs, outputs, and handoff triggers for each. The key design decision: what does each agent *not* know or *not* do?
3. **Write the role prompts** — Create a system-level prompt for each agent that sets its role, scope, and constraints. Explicitly state what's out of scope for each agent.
4. **Run the workflow sequentially** — Execute each agent in order, manually passing outputs between them. Track what you pass and what you leave out.
5. **Evaluate the result** — Compare the final output to what you'd get from a single "do everything" prompt. Document what the workflow architecture added.

"Done" looks like: You have a documented agent workflow (the architecture) and a finished output that went through the full pipeline. You can explain why you split the work the way you did and what each agent contributed.

? Why this matters (Strategists start here)

This is what agent collaboration looks like at professional scale — **architecture before implementation**. Every multi-agent framework (CrewAI, AutoGen, LangGraph) requires you to define roles, handoffs, and feedback loops before writing a single line of code. By doing it manually first, you understand the design decisions that make or break an agent system: what context each agent needs, where handoffs lose information, and when feedback loops help vs. when they add noise. This exercise builds the mental model that transfers to any agent tooling.

Reflection

- Which agent in your workflow had the biggest impact on output quality? Would the workflow still work without the weakest agent?
- What information was lost between handoffs? Would you design the handoffs differently next time?
- Where did the feedback loop help, and where did it just add noise? Is there a point of diminishing returns?
- ☐ *Walk a colleague through your agent architecture diagram before showing them the output. Ask them to predict where the pipeline would break — their predictions vs. reality reveals whether your architecture is intuitive or over-designed. (Social Learners)*

?? Level up

You've reached the advanced level for Agent Collaboration. From here, consider:

- Exploring agent frameworks like CrewAI or AutoGen to automate the handoffs you did manually
- Combining this skill with [WA-Advanced-01](#) to build end-to-end automated workflows
- Revisiting this exercise with a more complex project to push the architecture further

Back to [Agent Collaboration](#)

Revision #1

Created 2026-03-16 15:48:24 UTC by Admin

Updated 2026-03-16 15:48:24 UTC by Admin