Agent Collaboration

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

Your First AI Team Meeting

> **One-liner:** Run a multi-perspective AI session where one prompt gets you two expert viewpoints on the same problem — no extra tools required.

? Jump in (Tinkerers start here)

Pick a real decision you're currently facing. It could be a work decision, a project direction, or a problem you're stuck on.

Paste this prompt into any AI chat (ChatGPT, Claude, Gemini — anything works):

> I want you to act as two different experts giving me advice on **[your problem here]**.
>
> First, respond as a **[Role A]** — someone who focuses on **[their priority]**. Then, respond as a **[Role B]** — someone who focuses on **[their different priority]**.
>
> Keep each perspective clearly labeled. Be specific and give concrete recommendations, not vague advice.

Example — choosing whether to launch a feature now or wait:

> I want you to act as two different experts giving me advice on whether to launch our new onboarding flow this week or wait until next month.
>
> First, respond as a **growth-focused product manager** — someone who prioritizes user acquisition and speed to market. Then, respond as a **risk-aware QA lead** — someone who prioritizes stability, edge cases, and user trust.
>
> Keep each perspective clearly labeled. Be specific and give concrete recommendations, not vague advice.

After reading both perspectives, send this follow-up:

>

> Now, act as a **neutral facilitator**. Summarize where these two experts agree, where they disagree, and what the key trade-off is. End with a single question I should answer before making my decision.

Read the synthesis. Notice how one prompt gave you a structured debate that would normally require two people in a room.

? Plan first (Planners start here)

Here's what you're about to do:

1. **Choose your problem** — Pick a real decision or challenge you're working on right now. It works best when reasonable people could disagree about the right approach.
2. **Pick two expert roles** — Choose two perspectives that would naturally see your problem differently. Examples: marketer vs. engineer, short-term thinker vs. long-term strategist, customer advocate vs. operations manager.
3. **Write and send the dual-role prompt** — Use the template in the "Jump in" section. Fill in your problem and your two roles.
4. **Read both perspectives** — Notice where they conflict, where they agree, and which one you instinctively lean toward.
5. **Send the facilitator follow-up** — Ask the AI to synthesize the two views and surface the core trade-off.

"Done" looks like: You have a summary of two contrasting expert viewpoints and a clear understanding of the key trade-off in your decision.

? Why this matters (Strategists start here)

This exercise builds the foundational skill behind all multi-agent AI workflows: **defining specialized roles and comparing their outputs**. At the intermediate level, you'll split these roles across separate AI sessions with different contexts. At the advanced level, you'll design entire agent architectures. But it all starts here — training yourself to think in terms of roles, perspectives, and structured disagreement rather than asking AI once and accepting the first answer.

Reflection

- Did one perspective feel stronger than the other? Why — was it genuinely better argued, or did it just align with what you already believed?
- What did the facilitator synthesis surface that you hadn't considered?
- Would you use this dual-role technique for real decisions going forward? What types of decisions benefit most?
- □□ *Run this exercise with a colleague in the room. Have them choose different expert roles than you did for the same problem — the role selection itself reveals different priorities.* (Social Learners)

?? Level up

Ready for more? Try AC-Intermediate-01 — where you'll split these roles across separate AI sessions and learn to manage handoffs between them.

Back to Agent Collaboration

The Handoff Protocol

> **One-liner:** Split a problem across two separate AI sessions with different roles and contexts, then synthesize their outputs yourself — like managing a real team.

? Jump in (Tinkerers start here)

You'll need two AI chat windows open at the same time (two browser tabs, or two different AI tools — either works).

Pick a project or decision that has at least two distinct dimensions. For example: "Create a content strategy for launching our new product."

Chat A — The Strategist. Open your first chat and send:

> You are a **brand strategist** with 15 years of experience. Your focus is positioning, audience targeting, and messaging clarity. You do NOT think about implementation details — that's someone else's job.
>
> I'm working on: **[your project]**
>
> Give me your strategic recommendations. Focus on: who the audience is, what the core message should be, and how to position this differently from competitors. Be specific and opinionated.

Chat B — The Executor. Open your second chat and send:

> You are an **operations-focused content producer**. Your focus is practical execution: channels, formats, timelines, and resource requirements. You do NOT set strategy — you receive it and figure out how to make it real.
>
> I'm working on: **[your project]**

> Give me an execution plan. Focus on: which channels to prioritize, what content formats work best, a realistic timeline, and what resources I'll need. Be specific and practical.

Now you're the manager. Read both outputs. Notice what Chat A assumed that Chat B would question, and vice versa. Then write your own synthesis:

- Where do these perspectives align?
- Where do they conflict?
- What did each one miss that the other caught?
- What's your actual plan, informed by both?

Optional bonus round: Take your synthesis and paste it back into one of the chats:

> “ Here's the combined strategy and execution plan I've built from two different advisors. Poke holes in it. What's still weak?

? Plan first (Planners start here)

Here's what you're about to do:

1. **Choose a project** — Something real with both a strategic and practical dimension. Content launches, product decisions, event planning, and hiring processes all work well.
2. **Set up Chat A (Strategist)** — Give it a clear strategic role with explicit boundaries. Tell it *not* to worry about implementation.
3. **Set up Chat B (Executor)** — Give it a clear operational role with explicit boundaries. Tell it *not* to set strategy.
4. **Run both chats** — Send the same project description to each, but with their respective role prompts.
5. **Synthesize manually** — You are the integration point. Compare outputs, find gaps, resolve conflicts, and produce a combined plan.

"Done" looks like: You have a plan that neither AI session could have produced alone, and you can articulate what each perspective contributed.

? Why this matters (Strategists start here)

In AC-Basic-01, you simulated multiple perspectives in a single chat. Here, you're practicing a fundamentally different skill: **managing separate agents with isolated contexts**. This mirrors how real multi-agent systems work — each agent has a specific role, limited scope, and doesn't see the other's work. The human (you) acts as the orchestrator. This is the skill that scales: from two chats to entire AI-assisted workflows with specialized roles, handoff points, and quality gates.

Reflection

- How did the outputs differ when each AI had a constrained role vs. a single AI doing both? Was the split worth the extra effort?
- What context got lost in the handoff between sessions? How would you design a better transfer summary?
- Did the synthesis step feel harder or easier than you expected? What made it difficult?
- □□ *Run this exercise with two colleagues, each managing one AI session. Compare the experience of synthesizing someone else's AI output vs. your own — it highlights how much implicit context lives in your head.* (Social Learners)

?? Level up

Ready for more? Try AC-Advanced-01 — where you'll design a complete multi-agent workflow with defined roles, handoffs, and feedback loops.

Back to Agent Collaboration