Exercises

Every exercise in this playbook is designed to be self-contained — you can do any of them with just an AI tool and a bit of curiosity. No setup, no prerequisites, no special software.

Not Sure Where to Start?

Got 15 minutes? Try one of these beginner-friendly exercises:

- [The Fact-Check Habit](/exercises/ethical-prompting/ep-basic-01/) — Catch an AI making a mistake (you'll be surprised how easy it is)
- [The Signal in the Noise](/exercises/insight-synthesis/is-basic-01/) — Turn a messy AI brainstorm into something useful
- [The Reusable Prompt](/exercises/workflow-automation/wa-basic-01/) — Build a prompt template you'll actually use again

Got 25 minutes? Level up with an intermediate challenge:

- [The Prompt Chain](/exercises/workflow-automation/wa-intermediate-01/) — Build a 3-step AI pipeline
- [The Multi-Source Brief](/exercises/insight-synthesis/is-intermediate-01/) — Triangulate multiple AI perspectives
- [The Verification Checklist](/exercises/ethical-prompting

Ethical Prompting

Ethical Prompting

The AI Governance Playbook

> **One-liner:** Design a practical AI governance framework for a team or project — covering when to use AI, how to verify outputs, and what requires human judgment.

? Jump in (Tinkerers start here)

Pick a real team, project, or organization you work with. You're going to design an AI usage framework they could actually adopt.

Step 1 — Map the AI touchpoints. Send this prompt:

> I'm designing an AI governance framework for a **[team type/project type]** that does **[describe the work]**. Map out all the places where team members might use AI in their workflow. For each touchpoint, classify the risk level:
>
> - **Low risk:** AI errors are easily caught and consequences are minor (e.g., drafting internal emails, brainstorming)
> - **Medium risk:** AI errors could waste significant time or create confusion (e.g., research summaries, data analysis, first drafts of client deliverables)
> - **High risk:** AI errors could cause reputational, legal, or financial harm (e.g., published content, financial recommendations, legal language, customer-facing decisions)
>
> Present this as a table with: Touchpoint | Description | Risk Level | Why

Step 2 — Design the verification tiers. Based on the risk map, create a tiered verification system:

> Based on the risk map above, design a 3-tier verification system:

> **Tier 1 (Low risk):** What's the minimum verification needed? What can proceed without review? **Tier 2 (Medium risk):** What checks are required? Who reviews? What's the turnaround expectation? **Tier 3 (High risk):** What's the full review process? Who signs off? What documentation is needed?
>
> For each tier, specify:
>
> - Verification steps (checklist)
> - Who is responsible
> - What "approved" looks like
> - What happens when issues are found

Step 3 — Write the team guidelines. Now produce the actual document:

> Write a 1-page "AI Usage Guidelines" document for this team. It should be practical, not corporate. Include:
>
> 1. **When to use AI** — Green light scenarios
> 2. **When to be careful** — Yellow light scenarios with required verification
> 3. **When NOT to use AI** — Red light scenarios or scenarios requiring explicit approval
> 4. **Verification standards** — The tier system from above, simplified
> 5. **Attribution** — When and how to disclose AI usage
> 6. **Escalation** — What to do when you're unsure whether AI use is appropriate
>
> Write it in the tone of a senior colleague giving practical advice, not a legal department issuing mandates.

Step 4 — Red-team the framework. Test it:

> Now role-play as a team member who wants to use AI in a gray area. Come up with 3 realistic scenarios where the guidelines are ambiguous or where a reasonable person might interpret them differently. For each scenario, suggest how to clarify the guideline.

Revise the guidelines based on the edge cases.

? Plan first (Planners start here)

Here's what you're about to do:

1. **Choose your context** — Pick a real team or project. The framework should be one you could actually share or implement.
2. **Map AI touchpoints and risk levels** — Identify every place AI could be used in the workflow and classify each by potential harm from errors.
3. **Design tiered verification** — Create different verification processes for different risk levels. Not everything needs the same scrutiny.
4. **Write the guidelines** — Produce a practical 1-page document that a team member could reference in their daily work.
5. **Red-team with edge cases** — Test the framework against ambiguous scenarios. Fix any gaps before sharing.

"Done" looks like: A complete, practical AI governance framework (risk map + tiered verification + 1-page guidelines) that you could present to your team, plus documentation of edge cases you tested against.

? Why this matters (Strategists start here)

Individual verification habits (from EP-Intermediate-01) don't scale to teams. When five people use AI differently with different standards, the team's AI output quality is only as good as the weakest link. A governance framework creates **shared standards without bureaucracy** — it tells people what's safe to do quickly and what requires care, without making every AI interaction feel like a compliance exercise. This is also the document organizations will pay for: a practical, calibrated AI usage policy that actually gets followed.

Reflection

- Which risk classification was hardest to assign? What does that ambiguity tell you?
- Would your team actually follow these guidelines? What would make them ignore it?
- Did the red-teaming step reveal fundamental gaps, or just edge cases?
- □□ *Present your framework to a colleague and ask: "Would you follow this?" Their honest reaction is more useful than any AI review.* (Social Learners)

?? Level up

You've reached the advanced level for Ethical Prompting & Judgment. From here, consider:

- Presenting this framework to your actual team and iterating based on feedback
- Combining this with AC-Advanced-01 to add governance to multi-agent workflows
- Building a case study of how the framework changed AI usage behavior in your team

Back to Ethical Prompting & Judgment

Ethical Prompting

The Fact-Check Habit

> **One-liner:** Catch an AI making a confident mistake — and build a simple verification process you'll use every time.

? Jump in (Tinkerers start here)

Pick a topic you know well — your industry, your hobby, your area of expertise. Something where you can spot errors.

Step 1 — Get a confident answer. Send this prompt:

> Give me a detailed overview of **[topic you know well]**. Include specific facts, statistics, and examples. Be thorough and authoritative.

Read the output carefully. **Find at least one claim that feels off.** It might be a statistic that seems too round, a date that feels wrong, a name that's slightly off, or a causal claim that oversimplifies reality.

Step 2 — Make the AI check itself. Send this:

> Look at your previous response. I want you to fact-check yourself. For each specific claim, statistic, or example you cited:
>
> 1. Rate your confidence (high / medium / low)
> 2. Flag anything you might have fabricated or estimated
> 3. Identify which claims are most likely to be wrong and why
>
> Be ruthlessly honest. I'd rather know what you're uncertain about than have you defend everything.

Step 3 — Verify. Pick the 2-3 claims the AI flagged as lowest confidence. Google them. Were they accurate, close but wrong, or completely fabricated?

Step 4 — Build your check. Based on what you just learned, write a 3-line "verification prompt" you can append to any AI output:

> “ Before I use this, tell me:
>
> 1. Which specific claims are you least confident about?
> 2. What did you estimate or approximate vs. know with certainty?
> 3. What should I verify independently before sharing this?

Save this somewhere you'll see it. Use it as a default follow-up to any AI output you plan to rely on.

? Plan first (Planners start here)

Here's what you're about to do:

1. **Choose a familiar topic** — You need to be able to spot errors, so pick something in your area of knowledge. Don't use an unfamiliar topic — you won't know what to verify.
2. **Generate an authoritative-sounding response** — Ask AI for a detailed, factual overview. The more specific and confident the output, the more likely it contains subtle errors.
3. **Ask AI to fact-check itself** — Use the self-audit prompt to force the AI to rate its own confidence and flag potential fabrications.
4. **Independently verify** — Pick the lowest-confidence claims and check them against reliable sources. Track what was right, close, and wrong.
5. **Create your verification template** — Build a reusable 3-question follow-up that you'll use after any AI output you plan to act on.

"Done" looks like: You've caught at least one AI error, you understand *why* the AI got it wrong, and you have a saved verification prompt you can use going forward.

? Why this matters (Strategists start here)

The community's Ethical Prompting score is 75% — the highest of all five pillars. Most people *know* they should verify AI output, but few have a systematic process for doing so. This exercise closes the gap between awareness and practice by giving you a concrete, reusable tool. The verification prompt you build here becomes a habit — a 30-second step that catches errors before they become problems. At the intermediate level, you'll build a more comprehensive verification

checklist; this exercise establishes the baseline behavior.

Reflection

- What type of error did the AI make — a fabricated fact, a wrong date, or a subtle logical leap? Does the category matter for how you'd catch it?
- Did the AI's self-assessment match what you found when you verified manually? Was it too confident, too cautious, or well-calibrated?
- Will you actually use your verification prompt going forward? What would make it stick as a habit vs. something you forget about?
- □□ *Share a specific AI error you caught with a colleague. Ask them how they currently verify AI output — you may discover they don't.* (Social Learners)

?? Level up

Ready for more? Try EP-Intermediate-01 — where you'll build a comprehensive verification checklist and stress-test it against real AI outputs.

Back to Ethical Prompting & Judgment

Ethical Prompting

The Verification Checklist

> **One-liner:** Build a personal AI verification system — a checklist you'll actually use — and stress-test it against real AI outputs to find its limits.

? Jump in (Tinkerers start here)

You're going to build a verification checklist, then immediately try to break it.

Step 1 — Generate something to verify. Ask AI to produce a piece of content you might actually use in your work:

> Write a **[deliverable type — e.g., client email, project proposal, market analysis, technical recommendation]** about **[topic relevant to your work]** . Make it detailed and specific. Include data points, recommendations, and reasoning.

Step 2 — Build your checklist. Before reading the output carefully, write your own verification checklist. Start with these categories and add your own:

Check	Question	Pass/Fail
Factual claims	Are specific numbers, dates, or statistics verifiable?	
Sources	Could I find the original source for any cited information?	
Reasoning	Does the logic hold? Are there hidden assumptions?	
Completeness	What important perspective or consideration is missing?	
Tone/audience	Is the tone appropriate? Would the intended audience trust this?	
Actionability	Are the recommendations specific enough to actually follow?	

Check	Question	Pass/Fail
Your domain check	[Add a check specific to your field]	
Your domain check	[Add another check specific to your field]	

Step 3 — Apply the checklist. Go through the AI output line by line using your checklist. Mark each check as pass or fail. For every fail, note what the issue is.

Step 4 — Stress-test the checklist. Now deliberately ask AI to produce something harder to verify:

> Write the same type of **[deliverable]** but on a topic I'm less familiar with: **[topic outside your expertise]**. Make it equally detailed and authoritative.

Apply your checklist again. Where does it fail to catch problems? What check do you need to add?

Step 5 — Finalize. Update your checklist based on what you learned. Save it where you'll actually use it — bookmark it, pin it, print it, whatever works.

? Plan first (Planners start here)

Here's what you're about to do:

1. **Generate test content** — Ask AI to produce a work-relevant deliverable. This gives you realistic material to verify.
2. **Draft your checklist** — Build a structured verification list covering factual accuracy, reasoning quality, completeness, tone, and domain-specific concerns.
3. **Apply to familiar territory** — Use the checklist on AI output about a topic you know. This lets you calibrate how well the checklist catches real errors.
4. **Apply to unfamiliar territory** — Use the checklist on AI output about a topic you *don't* know well. This exposes gaps in your process — the errors you can only catch with domain knowledge.
5. **Iterate and save** — Update the checklist based on what it missed. Save it in a format you'll actually reach for.

"Done" looks like: A tested, refined verification checklist (8-12 items) saved in a usable format, with evidence of at least one error it caught and one gap you identified and fixed.

? Why this matters (Strategists start here)

In EP-Basic-01, you built a simple 3-question verification prompt. Here, you're building a **systematic process** — a checklist that works regardless of topic, catches both factual and reasoning errors, and is tuned to your specific work context. The community's 75% Ethical Prompting score means most people *intend* to verify AI output but lack a consistent method. A checklist turns good intentions into reliable behavior. At the advanced level, you'll scale this into a governance framework for a team; this exercise builds the individual practice first.

Reflection

- Which check caught the most problems? Which was least useful?
- How did your verification experience change between the familiar topic and the unfamiliar one?
- Is your checklist something you'd actually pull up before sending an AI-generated deliverable? What format makes it most likely you'll use it?
- □□ *Trade checklists with a colleague. Have them apply yours to an AI output from their work — their feedback will reveal blind spots specific to your domain.* (Social Learners)

?? Level up

Ready for more? Try EP-Advanced-01 — where you'll design an AI governance framework for a team or project.

Back to Ethical Prompting & Judgment

Insight Synthesis

Insight Synthesis

The Research Pipeline

> **One-liner:** Build a complete research synthesis pipeline — from question to evidence-graded conclusions — using structured AI queries and your own critical judgment.

? Jump in (Tinkerers start here)

Pick a question you genuinely need answered for your work. Not a trivia question — something where the answer shapes a real decision.

Phase 1 — Define the research question. Send:

> I need to research this question: **[your question]**
>
> Help me refine it into a research-ready question by:
>
> 1. Breaking it into 3-4 sub-questions that, if answered, would fully address the main question
> 2. For each sub-question, identifying what type of evidence would count as a strong answer (data, expert consensus, case studies, logical argument, etc.)
> 3. Flagging any assumptions embedded in the main question that I should test

Phase 2 — Structured evidence gathering. For each sub-question, run a separate AI query:

> Research sub-question: **[sub-question]**
>
> For this query, I want structured evidence:
>
> - **Strong evidence:** Claims supported by widely documented data, peer-reviewed research, or established expert consensus

> - **Moderate evidence:** Claims supported by credible case studies, industry reports, or respected analysis
> - **Weak evidence:** Claims based on anecdotes, single examples, logical inference without data, or common assertions that may not hold up
>
> Classify every claim you make. If you're not sure about the evidence quality, say so. I'd rather have honest uncertainty than false confidence.

Phase 3 — Contradiction analysis. After running all sub-queries, send this to a fresh session:

> Here are the findings from my research on **[main question]**:
>
> **Sub-question 1 findings:** [paste summary] **Sub-question 2 findings:** [paste summary] **Sub-question 3 findings:** [paste summary]
>
> Analyze the contradictions:
>
> 1. Where do the findings from different sub-questions conflict?
> 2. Which conflicts can be resolved by looking at the evidence quality?
> 3. Which conflicts are genuine unresolved tensions?
> 4. What additional evidence would resolve the remaining tensions?

Phase 4 — Your synthesis. Write a 500-word research brief yourself (not AI-generated) that answers your original question. Structure it as:

1. **Bottom line:** Your answer in 1-2 sentences
2. **Key evidence:** The 3 strongest pieces of evidence supporting your answer, with evidence grades
3. **Key uncertainty:** What you're least confident about and why
4. **What would change your mind:** 1-2 pieces of evidence that, if found, would reverse your conclusion

---

? Plan first (Planners start here)

Here's what you're about to do:

1. **Formulate a research question** — Choose something decision-relevant. Use AI to decompose it into sub-questions with defined evidence standards.
2. **Gather evidence by sub-question** — Run separate queries for each sub-question, requiring the AI to grade its own evidence quality (strong/moderate/weak).

3. **Analyze contradictions** — Feed all findings into a fresh session and ask for conflict analysis. Identify which conflicts are real vs. caused by weak evidence.
4. **Write your own synthesis** — Produce a 500-word brief that answers the question, cites evidence with quality grades, and states what would change your mind.
5. **Assess the pipeline** — Evaluate whether this process produced a meaningfully better answer than a single AI query would have.

"Done" looks like: A research brief that clearly distinguishes strong from weak evidence, acknowledges uncertainty, and provides a decision-ready answer with stated confidence.

? Why this matters (Strategists start here)

This exercise combines the skills from IS-Basic-01 (extracting signal from noise) and IS-Intermediate-01 (triangulating across perspectives) into a **complete research methodology**. The evidence grading system prevents the common failure mode of treating all AI output as equally reliable. The contradiction analysis surfaces genuinely open questions rather than papering over them. This pipeline is directly applicable to due diligence, competitive intelligence, policy analysis, and any context where the cost of being wrong is high and the question is too complex for a single query.

Reflection

- Did the evidence grading change which findings you trusted? Were you surprised by what was classified as "weak"?
- How did the contradiction analysis change your initial view?
- Was the 500-word synthesis harder or easier than expected? What was the hardest part — compression, confidence, or acknowledging uncertainty?
- □□ *Teach this pipeline to a colleague and have them run it on a different question. Compare how you each handle the "what would change your mind" step — that reveals different attitudes toward uncertainty.* (Social Learners)

?? Level up

You've reached the advanced level for Insight Synthesis. From here, consider:

- Using this pipeline for a real decision and tracking whether your evidence-graded conclusion held up
- Combining this with AC-Advanced-01 to delegate different research phases to different agent roles
- Teaching this method to a colleague and seeing how they adapt it

Back to Insight Synthesis

Insight Synthesis

The Signal in the Noise

> **One-liner:** Turn a messy AI brainstorm into a structured, actionable insight — learning to extract what matters and discard what doesn't.

? Jump in (Tinkerers start here)

Pick a topic you're genuinely curious about or working on. It could be a business challenge, a learning goal, or a decision you need to make.

Step 1 — Generate the mess. Send this prompt to any AI:

> Brainstorm 15-20 ideas about **[your topic]**. Don't filter or organize — just generate as many ideas as possible, even contradictory or half-formed ones. Number each idea.

Step 2 — Extract the signal. Now send this follow-up:

> Look at the brainstorm you just generated. Identify:
>
> 1. **The top 3 ideas** that are most actionable within the next week
> 2. **The 1 idea** that's most surprising or non-obvious
> 3. **The 2 ideas** that contradict each other — and what the tension between them reveals
> 4. **The pattern** — what theme or assumption connects most of these ideas?
>
> For each, explain your reasoning in one sentence.

Step 3 — Challenge the synthesis. Send this:

>

> Now tell me what's missing from this brainstorm. What obvious angle or perspective did you fail to include? Add 3 ideas that fill that gap.

Read the final output. You started with noise; you now have structured insight. The skill here isn't prompting — it's knowing what questions to ask *after* the AI generates raw material.

? Plan first (Planners start here)

Here's what you're about to do:

1. **Choose a topic** — Something you care about. The exercise works best with real problems, not hypotheticals.
2. **Generate raw material** — Ask AI for a large, unfiltered brainstorm (15-20 ideas). The messier the better — that's the point.
3. **Apply a synthesis framework** — Use the structured follow-up prompt to force the AI to categorize, rank, and find patterns in its own output.
4. **Identify gaps** — Ask the AI what it missed, then evaluate whether the gap-filling ideas actually change your understanding.
5. **Capture your insight** — Write a single sentence summarizing what you learned that you didn't know before.

"Done" looks like: You have 3 actionable ideas, 1 non-obvious insight, a clear tension to think about, and a unifying pattern — extracted from a wall of brainstorm text.

? Why this matters (Strategists start here)

AI is excellent at generating volume but mediocre at distinguishing signal from noise — that's still a human skill. This exercise builds your ability to use AI as a **thinking amplifier** rather than an answer machine. The synthesis framework (rank, surprise, contradict, pattern) is reusable: apply it to research outputs, meeting notes, customer feedback analysis, or any situation where you need to extract meaning from quantity. At the intermediate level, you'll synthesize across *multiple* AI outputs; this exercise builds the foundation.

Reflection

- Did the AI's ranking match your instinct? Where did you disagree, and what does that tell you about the AI's priorities vs. yours?
- Was the "gap" the AI identified actually a meaningful blind spot, or was it filler?
- Would you use this brainstorm-then-synthesize pattern again? For what kinds of problems does it work best?
- □□ *Run the same brainstorm prompt with a colleague present. Compare which ideas you each gravitate toward — the difference reveals your respective assumptions.* (Social Learners)

?? Level up

Ready for more? Try IS-Intermediate-01 — where you'll synthesize across multiple AI sessions to build a more complete picture.

Back to Insight Synthesis

Insight Synthesis

The Multi-Source Brief

> **One-liner:** Synthesize outputs from three separate AI queries into a single coherent analysis — building the skill of triangulating AI perspectives.

? Jump in (Tinkerers start here)

Pick a question or topic you need to actually understand — a market trend, a technology choice, a strategic decision, a complex issue in your field.

Run three separate queries in three different AI sessions (or clear context between each). Each query approaches the same topic from a different angle:

Query 1 — The Optimist:

> Analyze **[your topic]** from the most optimistic perspective. What's the strongest case that this will succeed/matter/grow? Cite specific evidence, trends, and examples. Be persuasive, not balanced.

Query 2 — The Skeptic:

> Analyze **[your topic]** from a skeptical perspective. What's the strongest case that this is overhyped, risky, or likely to fail? Cite specific evidence, counterexamples, and historical parallels where similar things didn't pan out. Be rigorous, not cynical.

Query 3 — The Analyst:

> Analyze **[your topic]** by identifying the 3-5 key variables that will determine the outcome. Don't argue for or against — map the decision space. For each variable, describe what would need to be true for a positive outcome vs. a

> negative one. Include what we don't yet know.

Now synthesize. Open a fresh document (not an AI chat). Write a 250-word brief that answers:

1. **What do all three perspectives agree on?** (This is likely true.)
2. **Where do the Optimist and Skeptic directly contradict each other?** (This is where the real uncertainty lives.)
3. **Which of the Analyst's key variables would resolve the contradiction?** (This is what you need to investigate.)
4. **Your take** — Given all three inputs, what's your position and what would change your mind?

The brief should be something you'd share with a colleague or decision-maker. No AI jargon, no meta-commentary about the process.

? Plan first (Planners start here)

Here's what you're about to do:

1. **Choose a topic** — Pick something with genuine uncertainty. If the answer is obvious, the exercise won't stretch you. Good candidates: emerging trends, strategic choices, technology bets, or contested ideas in your field.
2. **Run three separate AI sessions** — Optimist, Skeptic, and Analyst. Use fresh contexts (new chats or cleared conversations) so each query isn't influenced by the others.
3. **Read all three outputs** — Don't start synthesizing until you've read all three. Notice your own bias — which perspective did you instinctively agree with?
4. **Write the synthesis yourself** — This is the critical step. Don't ask AI to synthesize for you. The skill you're building is *your* ability to integrate contradictory information.
5. **Distill to 250 words** — Force compression. A good brief is one where every sentence earns its place.

"Done" looks like: A 250-word brief you'd be comfortable sharing with a colleague, built from three distinct AI perspectives, with a clear statement of what you believe and what would change your mind.

? Why this matters (Strategists start here)

In IS-Basic-01, you extracted insights from a single AI output. Here, you're building a fundamentally harder skill: **triangulating across multiple AI perspectives to form your own judgment**. This is exactly what senior decision-makers do with human advisors — they don't take any single perspective at face value. The discipline of writing the synthesis yourself (rather than asking AI to do it) ensures you're developing the judgment, not outsourcing it. This skill directly applies to research, due diligence, competitive analysis, and any situation where multiple data sources tell different stories.

Reflection

- Which perspective (Optimist, Skeptic, Analyst) was most useful? Which felt like filler?
- Did writing the synthesis yourself change your view compared to where you started? At what point in the writing did it shift?
- Would you share this brief with a decision-maker? If not, what's missing?
- □□ *Share your 250-word brief with someone who knows the topic. Ask them what they'd challenge — their pushback will tell you where your synthesis was weakest.* (Social Learners)

?? Level up

Ready for more? Try IS-Advanced-01 — where you'll build a full research synthesis pipeline with structured evidence evaluation.

Back to Insight Synthesis

Workflow Automation

Workflow Automation

The Workflow Blueprint

> **One-liner:** Design, document, and test a complete AI-automated workflow for a real business process — from trigger to output, with error handling and quality gates.

? Jump in (Tinkerers start here)

Pick a real business process that currently takes you 30+ minutes and involves multiple steps. Examples: weekly reporting, content production, customer onboarding documentation, project status updates, invoice processing.

Phase 1 — Map the current process. Send this:

> I'm going to automate this business process: **[describe the process]**.
>
> Help me map the current manual workflow:
>
> 1. What triggers the process? (time-based, event-based, request-based)
> 2. What are the sequential steps from trigger to final output?
> 3. What inputs does each step require?
> 4. What decisions are made at each step? (if/then logic)
> 5. Where are the bottlenecks or error-prone points?
> 6. What's the final deliverable and who receives it?
>
> Present this as a numbered workflow with decision points marked.

Phase 2 — Design the AI workflow. Send:

> Now redesign this as an AI-automated workflow. For each step, specify:

Step	Human or AI?	If AI: what prompt template?	If Human: what decision?	Input	Output	Quality gate

Rules:

- Some steps should remain human (judgment calls, approvals, sensitive decisions)
- Every AI step needs a quality gate — how do you know the output is good enough to proceed?
- Include error handling — what happens when an AI step produces bad output?
- Include a feedback mechanism — how does the workflow improve over time?

Phase 3 — Write the prompt templates. For each AI step in the workflow:

> Write the production-ready prompt template for Step **[N]**: **[step name]**
>
> The template should include:
>
> - Role definition for the AI
> - Clear input specification with [PLACEHOLDERS]
> - Exact output format requirements
> - Quality criteria the output must meet
> - An example of good output vs. bad output
>
> This prompt should work reliably every time with different inputs. It should be usable by someone who didn't design the workflow.

Phase 4 — Run the workflow end-to-end. Execute the full pipeline with real data. Track:

- Time per step (manual vs. AI-assisted)
- Quality gate pass/fail rates
- Where you had to intervene or override
- Total time saved vs. the manual process

Phase 5 — Document the blueprint. Create a 1-page workflow document:

> Write a "Workflow Blueprint" for this process that includes:

> 1. **Trigger:** What starts the workflow
> 2. **Flow diagram:** Step-by-step with decision points (use text-based flowchart)
> 3. **Prompt templates:** Reference to each template (step number and name)
> 4. **Quality gates:** What to check at each stage
> 5. **Error handling:** What to do when something fails
> 6. **Maintenance:** How to update the workflow as requirements change
> 7. **Metrics:** How to measure whether the workflow is working well

---

? Plan first (Planners start here)

Here's what you're about to do:

1. **Choose a process** — Pick something that takes 30+ minutes, involves multiple steps, and happens regularly. The more manual the current process, the bigger the payoff.
2. **Map the current workflow** — Document every step, decision point, and handoff. You can't automate what you don't understand.
3. **Design the hybrid workflow** — Decide what AI handles vs. what stays human. Add quality gates and error handling. Not everything should be automated.
4. **Build the prompt templates** — Write production-grade prompts for each AI step. These should be reusable by anyone, not just you.
5. **Test end-to-end** — Run the full workflow with real data. Measure time, quality, and failure points.
6. **Document the blueprint** — Create a shareable document that anyone could use to run this workflow.

"Done" looks like: A complete, tested workflow blueprint with prompt templates, quality gates, and measured time savings. Something you could hand to a colleague and they could execute without additional explanation.

? Why this matters (Strategists start here)

In WA-Intermediate-01, you built a 3-step prompt chain. Here, you're building a **production-grade workflow** — the kind of thing that saves hours per week and can be delegated. The key differences from an intermediate prompt chain: quality gates (not just chaining outputs blindly), error handling (what happens when AI fails), and documentation (others can run it without you).

This is directly transferable to tools like n8n, Make, or Zapier with AI steps. The blueprint format is also the deliverable that organizations pay consultants to produce.

Reflection

- How much time did the automated workflow save compared to the manual process? Is the saving worth the design effort?
- Which quality gates caught real problems? Which were unnecessary overhead?
- Where did AI fail and require human override? Was that predictable from the design phase, or did it only emerge during testing?
- □□ *Walk a colleague through your workflow blueprint and ask them to find the step most likely to fail. Fresh eyes spot single points of failure you've normalized.* (Social Learners)

?? Level up

You've reached the advanced level for Workflow Automation. From here, consider:

- Implementing this workflow in an automation tool (n8n, Make, Zapier) for true hands-free execution
- Combining this with AC-Advanced-01 to add multi-agent architecture to your workflow steps
- Measuring workflow performance over 4 weeks and iterating based on failure data

Back to Workflow Automation

Workflow Automation

The Reusable Prompt

> **One-liner:** Turn a task you do repeatedly into a reusable AI prompt template that works every time — your first step toward automation.

? Jump in (Tinkerers start here)

Think of something you do at least once a week that involves writing, analyzing, or summarizing. Examples: writing a status update, summarizing meeting notes, drafting an email to a client, reviewing a document.

Step 1 — Do it once with AI. Open any AI chat and do the task the way you normally would — just ask the AI to help. Don't overthink the prompt. Just get the job done.

Step 2 — Reverse-engineer your prompt. Now send this:

> Look at the prompt I just gave you and the output you produced. Help me turn this into a **reusable template** that I can use every time I need to do this task. The template should have:
>
> 1. **Clear placeholders** — marked with [BRACKETS] for the parts that change each time
> 2. **Fixed instructions** — the parts that stay the same every time
> 3. **Output format specification** — exactly what the result should look like (length, structure, tone)
>
> Write the template so someone else on my team could use it without any additional explanation.

Step 3 — Test it. Copy the template. Start a new chat. Paste the template and fill in the placeholders with a different example of the same task. Does the output match the quality of your original? If not, adjust the template.

Here's a concrete example:

Original task: "Summarize this meeting for my team"

Reusable template:

> Summarize the following meeting notes for a team update.
>
> **Meeting notes:** [PASTE NOTES HERE]
>
> **Output requirements:**
>
> - Start with a 1-sentence summary of the main decision or outcome
> - List action items with owner names in bold
> - Flag any unresolved questions
> - Keep the total summary under 150 words
> - Tone: professional but informal

? Plan first (Planners start here)

Here's what you're about to do:

1. **Identify a repeatable task** — Pick something you do weekly that involves text: writing, summarizing, analyzing, or formatting. The more repetitive, the better.
2. **Do it once with AI** — Complete the task normally. Don't try to be clever — just get a result you're happy with.
3. **Extract the template** — Ask the AI to help you identify what's fixed (instructions, format, tone) vs. what changes (the input data). Build a reusable template with clear placeholders.
4. **Test with a new example** — Use the template on a fresh instance of the same task. Compare quality to the original.
5. **Refine if needed** — If the template didn't produce equally good output, identify what was missing and add it.

"Done" looks like: You have a saved prompt template with clear placeholders that consistently produces good output for your repeatable task.

? Why this matters (Strategists start here)

Most people use AI in one-off conversations that disappear. This exercise introduces the shift from **ad-hoc prompting to systematic workflows** — the foundation of all AI automation. A reusable template is the simplest form of an AI workflow: defined input, consistent process, predictable output. At the intermediate level, you'll chain multiple templates together into multi-step workflows. Every automated AI process in production started as someone's reusable prompt.

Reflection

- What did you have to add to the template that wasn't obvious from the original prompt?
- Did the template produce consistent quality with different inputs, or did you need to tweak it? What was missing?
- How much time will this template save you per week? Is it enough to justify the setup effort?
- □□ *Send your template to a colleague who does the same task. Can they use it without any explanation? Their confusion points reveal where the template needs more specificity.* (Social Learners)

?? Level up

Ready for more? Try WA-Intermediate-01 — where you'll chain multiple prompt templates into a multi-step workflow.

Back to Workflow Automation

Workflow Automation

The Prompt Chain

> **One-liner:** Build a multi-step AI workflow where each step's output feeds into the next — turning a complex task into a repeatable pipeline.

? Jump in (Tinkerers start here)

Pick a task that has at least 3 distinct phases. Examples: writing a blog post (research, outline, draft, edit), analyzing a dataset (clean, analyze, summarize, recommend), or preparing a presentation (topic research, slide structure, talking points, Q&A prep).

Build a 3-step chain. Each step is a separate prompt. The output of each step becomes the input of the next.

Step 1 — Research/Gather:

> You are a research assistant. Your job is to gather the raw material for **[your task]**.
>
> Topic/context: **[describe what you're working on]**
>
> Produce a structured collection of: key facts, relevant examples, important considerations, and any constraints. Organize by theme. Do not draft anything — just collect the ingredients.

Copy the output. Start a new prompt (or clearly reset context).

Step 2 — Structure/Draft:

> You are a content architect. Your job is to turn raw research into a structured draft.
>
> Here is the research material: **[paste Step 1 output]**

> The final deliverable is: **[describe what you need — a blog post, a report, a strategy doc, etc.]**
>
> Create a structured draft. Include clear sections, key arguments in order, and placeholders for any examples or data points from the research. Focus on logical flow and completeness.

Copy the output. Start a new prompt.

Step 3 — Polish/Critique:

> You are a senior editor. Your job is to make this draft publication-ready.
>
> Here is the draft: **[paste Step 2 output]**
>
> The audience is: **[describe who will read this]**
>
> Do three things:
>
> 1. Improve clarity — simplify any convoluted sentences, cut unnecessary words
> 2. Strengthen weak points — flag any claim that needs better support and add it
> 3. Check consistency — ensure tone, terminology, and formatting are uniform throughout
>
> Produce the final version with an editor's note listing your key changes.

Now document the chain. Write down the 3 prompts as a reusable template (with [PLACEHOLDERS] for the parts that change). You've just built a prompt pipeline.

? Plan first (Planners start here)

Here's what you're about to do:

1. **Choose a multi-phase task** — Something that naturally has distinct stages (research → create → refine). The more phases, the more the chain helps.
2. **Design the chain** — Write 3 prompts, each with a clear role, input expectation, and output format. The key constraint: each step's output must contain everything the next step needs.

3. **Run the chain** — Execute each step sequentially, passing the output forward. Use fresh contexts between steps to prevent bleed-through.
4. **Evaluate information flow** — Notice where context was lost between steps. What did Step 3 need that Step 2 didn't preserve?
5. **Document as a template** — Save the chain with placeholders so you can reuse it for the same type of task.

"Done" looks like: A completed deliverable that went through a 3-step pipeline, plus a documented prompt chain template with placeholders for reuse.

? Why this matters (Strategists start here)

In WA-Basic-01, you built a single reusable prompt. Here, you're learning to **chain prompts into a workflow** — the building block of all production AI automation. Every AI-powered pipeline (content generation, data analysis, document processing) is fundamentally a prompt chain with handoffs. The skill you're building — decomposing a task into stages, defining clear inputs and outputs, managing context between steps — is the same skill used in tools like n8n, Zapier AI, or custom LLM pipelines. Manual chaining teaches you what to automate and where the bottlenecks live.

Reflection

- Where did context get lost between steps? What information did a later step need that an earlier step didn't pass along?
- Did the 3-step chain produce better output than a single "do everything" prompt? Where specifically was the improvement?
- Which step in the chain was the weakest link? How would you redesign it?
- □□ *Have a colleague run your documented chain on a different task of the same type. Their experience reveals whether your chain is truly reusable or depends on your implicit knowledge.* (Social Learners)

?? Level up

Ready for more? Try WA-Advanced-01 — where you'll design and document a complete AI-automated workflow for a business process.

Back to Workflow Automation

Cross-Domain Reframing

The Cross-Domain Prompt Library

> **One-liner:** Build a documented library of prompt patterns borrowed from 3+ different fields, with tested adaptations and transfer notes for your own domain.

? Jump in (Tinkerers start here)

You're going to build a personal prompt library of techniques stolen from other fields — documented well enough to teach someone else.

Phase 1 — Survey 3 domains. Pick 3 fields that are different from your own *and* from each other. Send this to three separate sessions:

> How do professionals in **[field]** use AI in sophisticated ways? I don't want generic "they use ChatGPT" answers. Give me 5 advanced AI techniques or prompting patterns that are specific to this field. For each:
>
> 1. Name the technique
> 2. Describe the prompt pattern (what input, what instructions, what output format)
> 3. Why this technique works in this domain (what problem it solves)
> 4. Example prompt (ready to use)

Phase 2 — Select your top 5. From the 15 techniques across 3 domains, pick the 5 that are most interesting or most likely to transfer. For each one, send:

> Analyze this technique from **[source domain]**: **[technique description]**
>
> Map the transfer potential:

> 1. **Core principle:** What's the underlying mechanism that makes this work, independent of domain?
> 2. **Direct transfer:** What would this look like applied to **[your field]** with minimal modification?
> 3. **Modified transfer:** What would need to change to make it work well in my context?
> 4. **What doesn't transfer:** What aspect is domain-specific and should be replaced?
> 5. **Adapted prompt:** Write a ready-to-use version for my field

Phase 3 — Test each adapted prompt. Run all 5 adapted prompts on real tasks in your work. For each, document:

Technique	Source Domain	My Task	Result Quality (1-5)	What Worked	What Needed Adjustment

Phase 4 — Build the library entry. For the 3 best-performing techniques, create a library card:

> Create a "Prompt Library Card" for this technique:
>
> **Name:** [give it a memorable name] **Borrowed from:** [source domain] **Core principle:** [1 sentence — why this works] **Original use:** [what it does in the source domain] **My adaptation:** [what it does in my domain] **Ready-to-use prompt:**
>
> ```
> [the tested, refined prompt with placeholders]
> ```
>
> **When to use:** [scenarios where this technique is the right choice] **When NOT to use:** [scenarios where it fails or is overkill] **Transfer notes:** [what I learned about adapting this — tips for others]

? Plan first (Planners start here)

Here's what you're about to do:

1. **Survey 3 unfamiliar domains** — Discover advanced AI techniques in three different fields. Cast a wide net — diversity of domains matters more than depth.
2. **Select the top 5 candidates** — From 15 techniques, choose 5 based on transfer potential and novelty. Look for techniques that solve a problem *structurally* similar to one in your work.

3. **Analyze transfer mechanics** — For each technique, separate the domain-specific elements from the core principle. Identify what transfers directly, what needs modification, and what should be replaced.
4. **Test all 5 adaptations** — Run each adapted prompt on a real task. Document quality, surprises, and adjustments needed.
5. **Document the top 3** — Create library cards with enough detail for someone else to use the technique without your guidance.

"Done" looks like: A 3-entry prompt library with tested techniques from other domains, complete with ready-to-use prompts, usage guidance, and transfer notes.

? Why this matters (Strategists start here)

In CDR-Basic-01, you borrowed a single technique. In CDR-Intermediate-01, you transplanted an entire framework. Here, you're building a **systematic practice** — a personal library that compounds over time. The library card format forces you to articulate *why* a technique transfers, which is the meta-skill: once you can spot the structural similarity between domains, you can generate new cross-domain adaptations on your own. This library also becomes a shareable team asset — a collection of non-obvious AI techniques that others in your field won't have discovered.

Reflection

- Which of the 3 domains produced the most transferable techniques? Why?
- Did any technique work *better* in your domain than in its original domain? What does that tell you?
- What pattern do you notice in what transfers well vs. what doesn't? Can you articulate a rule of thumb?
- □□ *Share your prompt library with colleagues and have them test the techniques on their own tasks. The techniques that work across multiple people's contexts are genuinely domain-agnostic — those are your keepers.* (Social Learners)

?? Level up

You've reached the advanced level for Cross-Domain Reframing. From here, consider:

- Expanding the library monthly — add one new cross-domain technique per month from a new field

- Sharing the library with colleagues and collecting their transfer notes
- Combining this with WA-Advanced-01 to build cross-domain techniques into automated workflows

Back to Cross-Domain Reframing

Cross-Domain Reframing

The Stolen Technique

> **One-liner:** Take an AI technique from a completely different field and apply it to your own work — discovering that the best prompting ideas are often borrowed.

? Jump in (Tinkerers start here)

Pick a field that is **not** your own. If you work in marketing, pick engineering. If you're a designer, pick finance. If you're a developer, pick journalism. The more unfamiliar, the better.

Step 1 — Discover a technique. Send this prompt:

> How do professionals in **[unfamiliar field]** use AI in their daily work? Give me 5 specific, concrete techniques — not general concepts. For each technique, describe: what they prompt the AI to do, what input they provide, and what output they get. Focus on techniques that are unique to this field.

Step 2 — Steal the best one. Pick the technique that seems most interesting or most different from how you currently use AI. Then send:

> I work in **[your field]**. Take the technique you described as #[number] — **[briefly describe it]** — and help me adapt it for my work. Specifically:
>
> 1. What would the equivalent input look like in my field?
> 2. How would I modify the prompt to fit my context?
> 3. What output would I expect?
> 4. Write me a ready-to-use prompt that applies this borrowed technique to **[a specific task you do]**.

Step 3 — Test it. Copy the adapted prompt. Use it on a real task. Compare the result to how you'd normally approach it.

Example — a marketer borrowing from investigative journalism:

The technique: Journalists use AI to cross-reference claims across multiple sources and flag inconsistencies.

The adaptation: A marketer uses the same technique to cross-reference their product claims against competitor claims and customer reviews, flagging gaps between promise and reality.

? Plan first (Planners start here)

Here's what you're about to do:

1. **Pick an unfamiliar field** — Choose something genuinely outside your expertise. The discomfort is the point — that's where non-obvious ideas live.
2. **Research AI techniques in that field** — Use AI to discover how professionals in that domain use AI tools. Look for specific techniques, not generalities.
3. **Identify a transferable technique** — Pick one that solves a problem similar to something in your work, even though it looks completely different on the surface.
4. **Adapt with AI's help** — Ask the AI to bridge the gap between the source domain and your domain. Get a ready-to-use prompt.
5. **Test the borrowed technique** — Apply it to a real task and evaluate whether it gives you a different (and possibly better) result than your usual approach.

"Done" looks like: You have a working prompt borrowed from another field that gives you a new angle on a familiar task.

? Why this matters (Strategists start here)

Most people prompt AI using patterns from their own field — but the most powerful AI techniques are often domain-agnostic. Researchers structure AI analysis differently than marketers, engineers test AI outputs differently than writers, and each field has developed prompting patterns the others rarely see. Cross-domain reframing is how you break out of local optima in your AI usage. At the intermediate level, you'll systematically adapt entire prompt strategies across domains; this exercise builds the muscle of looking outside your field for AI inspiration.

Reflection

- Did the borrowed technique produce a noticeably different result than your usual approach? Better, worse, or just different?
- What made the technique transferable? Was it the structure, the question type, or the underlying problem it solves?
- Which other field would you explore next for AI techniques? What made you choose it?
- □□ *Ask a colleague from a different department how they use AI. You'll likely discover a technique you've never considered — that's cross-domain reframing in action.* (Social Learners)

?? Level up

Ready for more? Try CDR-Intermediate-01 — where you'll systematically adapt an entire prompting strategy from an unfamiliar domain.

Back to Cross-Domain Reframing

Cross-Domain Reframing

The Framework Transplant

> **One-liner:** Take a complete problem-solving framework from another domain and systematically adapt it to solve a challenge in your own work.

? Jump in (Tinkerers start here)

Pick a challenge you're currently facing in your work — something you've been approaching the same way without breakthrough results.

Step 1 — Find a foreign framework. Send this prompt:

> I'm struggling with **[your challenge]** in my field of **[your field]**. I want a completely fresh approach. Give me 3 well-known problem-solving frameworks from **different** fields (engineering, medicine, military strategy, game design, ecology — anything outside my domain). For each framework:
>
> 1. Name and origin field
> 2. How it works (3-4 step process)
> 3. Why it might apply to my problem
>
> Choose frameworks that are genuinely different from each other, not variations on the same idea.

Step 2 — Deep-dive one framework. Pick the most promising or most surprising framework. Send:

> Let's go deeper on **[chosen framework]**. Walk me through how a professional in **[its origin field]** would apply this framework to a real problem in their domain. Be specific — give me a concrete example with actual steps, not abstractions.

Step 3 — Systematic transplant. Now adapt it:

>

> Now help me transplant this framework to my challenge: **[restate your challenge]**.
>
> Map each step of the framework to my context:
>
> - **Step 1 of framework** → What does this look like in my situation?
> - **Step 2 of framework** → What's the equivalent action?
> - (continue for all steps)
>
> For each mapping:
>
> - What translates directly?
> - What needs to be modified and how?
> - What doesn't transfer at all, and what should replace it?
>
> End with a concrete action plan I can execute this week.

Step 4 — Stress test. Send:

> Play devil's advocate. Where does this transplanted framework break down when applied to my field? What assumptions from the original domain don't hold in mine? How should I adjust?

? Plan first (Planners start here)

Here's what you're about to do:

1. **Identify your challenge** — Pick something real where your current approaches have stalled. The exercise only works if you're genuinely stuck.
2. **Discover foreign frameworks** — Use AI to surface structured problem-solving approaches from unfamiliar fields. Look for frameworks with clear steps, not just theories.
3. **Study the framework in its native context** — Understand how it actually works in practice before trying to adapt it. This prevents shallow borrowing.
4. **Map step-by-step to your context** — Systematically translate each step, noting where the mapping is direct, where it needs modification, and where it fails entirely.
5. **Stress test the adaptation** — Identify where the transplant breaks down and adjust before committing to action.

"Done" looks like: A concrete action plan for your challenge, based on a framework from another field, with clear documentation of what translated, what was modified, and what was replaced.

? Why this matters (Strategists start here)

In CDR-Basic-01, you borrowed a single technique from another field. Here, you're transplanting an **entire framework** — a much harder and more valuable skill. This is how breakthrough innovations happen: the structure of a solution transfers across domains even when the details don't. Toyota's production system was adapted from supermarket inventory management. Agile software development borrowed from lean manufacturing. The ability to systematically adapt frameworks across domains is what separates insight from coincidence. At the advanced level, you'll build an entire cross-domain prompt library; this exercise builds the adaptation methodology.

Reflection

- Which parts of the framework transferred most easily? What does that tell you about the underlying structure of your problem?
- Where did the transplant break down? Was the breakdown due to domain differences, or did it reveal an assumption you hadn't questioned?
- Did the stress test change your action plan significantly, or just refine the edges?
- □□ *Explain the transplanted framework to someone in the original field. Their reaction ("that's not how we use it" or "interesting adaptation") tells you whether you captured the core principle or just the surface.* (Social Learners)

?? Level up

Ready for more? Try CDR-Advanced-01 — where you'll build a cross-domain prompt library with documented transfer patterns.

Back to Cross-Domain Reframing

Agent Collaboration

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

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

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