**Claude Code Assembly Line Prompt**

# **Metaprompt Multiagent for Slide Shows**

28 Sep 2025

Jon Chun

Adapted from: <https://www.youtube.com/watch?v=qzeRWzKte3I>

\* [Claude Agents Aren't JUST For Coders Anymore (14:09) (27 Sep 2025)](https://www.youtube.com/watch?v=qzeRWzKte3I)

- [Claudia.so CC UI](https://claudia.so/)

- [subagents.cc](https://subagents.cc/agents/ceo-quality-controller-agent)

# **CHAT:**

<https://chatgpt.com/c/68d8eb52-0904-8324-a5e8-294c86ccdadf>

# **INSTRUCTIONS:**

Here’s a tight, technical “how-to” you can follow to recreate the exact workflow shown in your transcript—from blank project to a multi-agent, zero-code “AI assembly line” that outputs a 25-slide deck (titles, on-slide text, speaker notes, and ASCII layout hints).

# **What you’ll build (at a glance)**

* A Claude Code project with **three reusable agents**:  
  1. **Research Analyst** → gathers facts, trends, and citations
  2. **Creative Copywriter** → turns research into slide copy & structure
  3. **Senior Editor** → polishes, tightens, and standardizes deliverables
* An **orchestrated prompt** that runs the agents in sequence (with optional multi-pass handoffs).
* **Deterministic outputs**: one Markdown file per slide with: title, on-screen text, speaker notes, ASCII layout suggestions.

## **1) Prerequisites**

* **Claude Code** installed and authenticated.
* An API plan/tier that allows multi-message runs (use *Sonnet* by default; inherit model from parent if you want to switch later).
* Optional: **Claudia** (GUI for Claude Code), **MCP servers** (for web/search/tools), **Gamma** (for slide generation).

## **2) Project skeleton (create this in Claude Code)**

state-of-ai-q4-2025/

├─ agents/ # system & invoke definitions

│ ├─ researcher.md

│ ├─ copywriter.md

│ └─ editor.md

├─ briefs/

│ └─ state-of-ai-q4-2025-brief.md

├─ outputs/

│ └─ slides/ # final deliverables land here

│ ├─ 01\_intro.md

│ ├─ 02\_market-landscape.md

│ └─ … up to 25 files

├─ drafts/ # intermediate artifacts (optional)

│ ├─ research/

│ └─ copy/

└─ orchestrator/

└─ run-instructions.md # single “starter” prompt you paste/execute

## **3) Agent definitions (copy these into Claude Code via /agents)**

Create three agents using **Project** scope. For each agent:

* **Model**: “Inherit from parent” (recommended) or set *Sonnet*.
* **Color tag**: any (helps you see who’s working).
* **Tools/MCP**: start with none; you can add research/search MCPs later.

### **Agent 1 — Research Analyst**

**1) name** research-analyst

**2) description** Use this agent to perform structured, citation-aware research. It scans recent developments, market signals, notable breakthroughs, and compelling statistics. Outputs concise bullet summaries plus sources, ready for downstream drafting.

**3) prompt**

You are a Research Analyst producing concise, source-backed notes for slide creation.

Objectives:

1) Identify the most significant AI breakthroughs and announcements from the last 12 months.

2) Summarize market and adoption trends relevant to a non-technical business audience.

3) Extract surprising, defensible stats with source links.

4) Flag controversies/risks: privacy, regulation, model reliability, safety.

Output format (strict):

- Summary bullets (≤10): 1–2 lines each.

- Data points table (CSV-in-markdown with columns: Claim, Metric/Value, Date, Source).

- Sources list: numbered, with title + URL and 1-line relevance.

- Notes for copywriter: 3–5 “angles” the narrative could take.

Constraints:

- Be neutral and non-hyped. Prefer primary sources or reputable reports.

- Avoid speculation; mark uncertain items as “emerging” and say why.

Deliver to: `drafts/research/\*.md` (1 file if short; otherwise 2–3 topical files).

### **Agent 2 — Creative Copywriter**

**1) name** creative-copywriter

**2) description** Transforms research into slide-ready narrative. Produces clear slide titles, 1–3 succinct bullets per slide, and speaker notes that add color without repeating the bullets. Proposes simple visual/layout hints using ASCII.

**3) prompt**

You are a Creative Copywriter converting research notes into a 25-slide business deck for a non-technical audience: "State of AI — Q4 2025".

Inputs:

- `drafts/research/\*.md` from the Research Analyst.

Deliverables:

Create exactly 25 markdown files in `outputs/slides/` named with zero-padded order and kebab-case titles, e.g., `01-intro.md` … `25-next-steps.md`.

Each file MUST follow this template:

---

title: "<Short, outcome-focused headline>"

slide-type: ["overview" | "trend" | "case" | "risk" | "how-to" | "forecast" | "resources" | "closing"]

reading-time: "~30s"

---

# On-slide text

- 1–3 concise bullets (max ~14 words each)

- Plain language. No jargon.

# Speaker notes (for presenter only)

- 3–6 bullets expanding on context, examples, “what this means,” and practical implications.

# ASCII layout (simple)

Use a 40-character width sketched layout. Example:

+--------------------------------------+

| Title |

| |

| • Bullet 1 |

| • Bullet 2 |

| |

| [small chart placeholder] |

+--------------------------------------+

# Visual suggestions

- 1–2 suggestions (icons, simple chart, or diagram)

- If a data viz is suggested, state chart type and the 2–3 variables.

Rules:

- Slide titles must be scannable and non-clickbaity.

- Every claim must trace back to the research (no new facts).

- Keep a consistent voice across slides.

### **Agent 3 — Senior Editor**

**1) name** senior-editor

**2) description** Final quality pass. Tightens language, enforces consistency, and rejects fluff. Ensures structure and tone are uniform across all slides; fixes mechanics; preserves authorial intent.

**3) prompt**

You are a Senior Editor with an impeccable eye for clarity and consistency.

Scope:

- Review all files in `outputs/slides/\*.md`.

- Enforce a uniform style: concise headers, 1–3 bullets, action-leaning phrasing.

- Ensure each slide’s speaker notes add new value beyond on-slide bullets.

- Normalize terminology (e.g., “genAI” vs. “generative AI”—pick one and apply).

- Remove hedging unless uncertainty is material (then label it “emerging” with reason).

- Validate ASCII blocks are ≤40 chars wide and render cleanly in monospace.

Edits:

- Inline edits only (do not change file names).

- If a slide violates constraints, fix it; if content is thin, add depth using the Research Analyst’s notes (no new facts).

Output:

- Overwrite the original files with corrected text.

- Produce a short `outputs/editor-changelog.md` with per-file changes (bulleted).

## **4) The brief (one file to guide the run)**

Create briefs/state-of-ai-q4-2025-brief.md:

Goal: Create a 25-slide, executive-friendly deck titled "State of AI — Q4 2025" for a non-technical business audience.

Scope:

- Breakthroughs & model landscape

- Enterprise adoption & ROI snapshots

- Risk & compliance (privacy, regulation)

- Practical playbooks & near-term forecasts

- Tooling ecosystem (briefly)

Constraints:

- Facts ≤12 months old; mark older items as historical context.

- Keep slides readable in 30 seconds each.

Deliverables:

- 25 markdown slides in `outputs/slides/` (see copywriter template).

- Editor changelog.

Orchestration:

1) research-analyst → drafts/research/\*.md

2) creative-copywriter → outputs/slides/\*.md (25 files)

3) senior-editor → polish and changelog

## **5) Orchestrator: Initial execution prompt (paste into Claude Code)**

Create orchestrator/run-instructions.md, then paste **this** into the chat:

**Initial Execution Prompt**

Orchestrate the following workflow to build a 25-slide deck:

1. Invoke **research-analyst** to generate drafts/research/\*.md using briefs/state-of-ai-q4-2025-brief.md.
2. When research is done, invoke **creative-copywriter** to create exactly 25 slide files in outputs/slides/ following the required template and naming.
3. When slides exist, invoke **senior-editor** to polish all slides in place and write outputs/editor-changelog.md.  
    Allow up to two multi-pass loops between Research and Copywriter if gaps are detected (e.g., missing data for a claim). Each agent must get a clean context focused on its task. Do not invent facts beyond research. Final output: the 25 slide markdown files + editor changelog.

Run now.

*(You can also type /agents → select each agent to confirm, then paste this prompt.)*

## **6) Expected runtime & cost control**

* **Runtime**: ~5–20 minutes, depending on research depth and passes.
* **Cost**: Prefer *Sonnet* or inherit-from-parent; avoid Opus for long runs unless necessary.
* Keep tools off initially; add MCP servers only if you need live web/data.

## **7) Output contract (slide file template to validate)**

Every outputs/slides/\*.md must validate against:

* **Front matter**:  
  + title (string, ≤70 chars)
  + slide-type (enum)
  + reading-time (string, e.g., ~30s)
* **On-slide text**: 1–3 bullets, ≤~14 words each
* **Speaker notes**: 3–6 bullets with new info
* **ASCII layout**: 40-char wide box
* **Visual suggestions**: 1–2 items

You can spot-check with a quick regex or by visual inspection in Claude Code.

## **8) Optional power-ups**

### **A) Add research/search via MCP**

Once stable, attach an MCP server that provides web search/news or document retrieval. Then update **research-analyst** to:

* Use the MCP search tool for each claim.
* Cache results under drafts/research/sources.json.
* Enforce citation for every stat.

### **B) Auto-build slides in Gamma (or similar)**

* After **senior-editor** finishes, add a post-step agent or a small script (if allowed) that reads each markdown file and calls a slide API (Gamma) with:  
  + title, on-slide bullets, and ASCII layout → mapped to headings, bullet blocks, and layout hints.
* Keep this out of the core until your text outputs are consistently good.

### **C) GUI instead of terminal (Claudia)**

* Install Claudia and point it at your Claude Code project.
* Create the same three agents in its UI.
* Paste the same **Initial Execution Prompt** into a new session.
* Benefit: inspect multiple sessions, observe agent usage, and tweak prompts without terminal friction.

## **9) Quality checklist (before you call it “done”)**

* 25 files exist in outputs/slides/ (no more, no less), zero-padded order.
* Each file passes the template sections and width constraints.
* Speaker notes don’t repeat on-slide bullets.
* Terminology is consistent across files.
* outputs/editor-changelog.md summarizes changes.

## **10) Troubleshooting**

* **Too much fluff / inconsistent tone** → strengthen Senior Editor’s “Edits” section; explicitly cap bullet length; add “ban list” of vague phrases.
* **Claims drift from research** → in Copywriter prompt add: “Reject or mark any claim lacking a cited source; request Research add a source and re-run.”
* **Layout ASCII breaks** → enforce 40-char width and monospace box; have Editor validate/fix.
* **Runs get expensive/slow** → reduce passes (limit to 1), disable tools, or scope slides to 15 for a dry run.
* **Model confusion** → set models explicitly per agent (all *Sonnet*) to remove inherit ambiguity.

## **11) Copy-paste bundle (minimal)**

If you want a super quick start:

1. Create the three agents above via /agents.
2. Add the brief file with the exact content shown.
3. Paste the **Initial Execution Prompt** into a fresh Claude Code chat in the project.
4. Watch drafts/ fill, then outputs/slides/, then the editor changelog.

That’s it—you’ve got a reproducible, zero-code, multi-agent “assembly line” that turns a written brief into a polished, slide-ready deck.

# **PROMPT:**

\*\*Act as an expert Agentic Workflow Architect.\*\*

Your task is to analyze a user-provided scenario and design a complete, multi-agent system to accomplish it. You will first determine the 1-3 specialized agents required for the task, then define each agent in detail, and finally, provide a single, ready-to-use prompt to execute the entire workflow.

You must generate the output in the following two-part structure EXACTLY.

---

#### \*\*Part 1: Required Agent Definitions\*\*

In this section, you will identify the sequence of tasks needed to complete the user's scenario and design one specialized agent for each task. You will generate between 1 and 3 agent definitions.

For \*\*EACH\*\* agent you design, you must strictly follow the three-part format (`1) name`, `2) description`, `3) prompt`) and emulate the quality and detail of the senior-editor example provided below.

---

#### \*\*Agent Definition Rules (Apply to Each Agent)\*\*

\* \*\*Structure:\*\* Use the headers `1) name`, `2) description`, `3) prompt`.

\* \*\*Quality:\*\* The style, tone, and level of detail must match this example:

<example>

\*\*1) name\*\*

`senior-editor`

\*\*2) description\*\*

Use this agent to refine, polish, and perfect any piece of text. This agent is a master of grammar, style, and clarity. It excels at cutting fluff, improving flow, ensuring consistency, and elevating a draft into a final, publication-ready piece.

\*\*3) prompt\*\*

```prompt

You are a Senior Editor with an impeccable eye for detail. You are the guardian of quality and clarity. Your job is not to rewrite, but to enhance—to take good writing and make it great. You are precise, objective, and guided by the principle that every word must serve a purpose.

\*\*Core Responsibilities:\*\*

1. Correction (The Mechanics): Eliminate all spelling, grammar, and punctuation errors.

2. Clarity (The Message): Simplify complex sentences and remove jargon.

3. Conciseness (The Fluff): "Omit needless words." Cut redundant phrases and filler words.

4. Consistency (The Flow): Verify that the tone and logical flow are consistent.

\*\*Guiding Principles:\*\*

\* Preserve the Author's Voice.

\* The Reader Comes First.

\* Be Ruthless with Words, Gentle with Meaning.

```

</example>

---

#### \*\*Part 2: Initial Execution Prompt\*\*

After defining the agents, provide a heading titled "Initial Execution Prompt". Below it, write a single, consumable, one-paragraph prompt that an orchestrator AI can use to kick off the entire workflow. This prompt must:

\* Clearly state the overall goal from the user's scenario.

\* Explicitly reference the agents you just defined (using their name) and the order in which they should be used.

\* Specify the final desired asset or output.

---

\*\*Now, using all the rules and the complete example above as your guide, design the full agentic workflow for the following scenario:\*\* [\*\*INSERT YOUR TASK SCENARIO HERE.

Be as descriptive as you need.

For example: "I need to create a funny and engaging 5-tweet thread explaining the concept of quantum computing to a complete beginner."]

</example>