

How Claude Turns One Call Into a Week of Content

A step-by-step walkthrough for taking one messy call and turning it into a client-ready project plan, then into a full week of content. Part 1 sets Claude up across Chat, Projects, Artifacts, and Connectors. Part 2 hands the whole job to Cowork and runs the transcript-to-content workflow. Part 3 is the full recap: what you get, how the pieces fit, and how skills lock in your voice.

Watch the full walkthrough:
youtube.com/@EmileGiovannie

Work with me: emile@giovannielabs.ai

Part 1: Turn a Messy Call into a Client-Ready Project Plan

1. Start with Chat and give Claude real context

Use Chat as the front door for quick, one-off tasks. Don't ask vague, Google-style questions. Before pasting anything, tell Claude:

- **Who it should act as** (e.g. executive assistant, marketing manager, business owner)
- **What you want**
- **The format you want back**

Example prompt: *Act as an executive assistant to the CEO of an AI business. Pull the exact commitments from this call and list the action items with deadlines.*

If the first answer is close but not quite right, refine it. Example: *Give each action item two deadlines: my internal deadline and the client-facing deadline.*

2. Move repeatable work into a Project

Use Projects when you want Claude to remember your business context across chats. A project stores the things you would otherwise repeat every time:

- Services
- Pricing
- Tone of voice
- Best examples of past work
- Rules for how you like things done

Create one from the left sidebar: click **Project**, choose **New Project**, name it, and add a short description of what you're trying to achieve. Inside, add **Instructions** (how Claude should structure answers) and **Files** (sample emails, proposals, past work). Keep it focused: use only your best 2-3 examples so it stays fast. Then move chats into the project so future work inherits that context.

3. Turn the output into a polished Artifact

Use Artifacts when you want Claude to produce something visual and nearly finished, such as a document, a checklist, a calculator, or a small webpage. Here, Claude turns call commitments into a clean

project plan you can edit directly, drag and adjust, shift by date, reset, and download or copy.

Why this matters:

- It's much easier to review a visual deliverable than a wall of text
- It's better for client handoff
- It saves time when you need something usable, not just explained

4. Connect Claude to the tools where your work already lives

Use Connectors so Claude can pull information from your existing apps instead of copy-pasting. Common sources:

- Calendar
- Email
- Call recorder tools
- Google Drive
- Fireflies

To set one up: click the **plus** button in chat, choose **Connectors**, search for the app, then authenticate and grant access. Permission settings matter: **read-only tools** can often be set to always allow, while **write/delete tools** should usually require approval. Example: Claude goes into Fireflies and pulls the latest meeting transcript automatically.

Part 2: Use Cowork to Turn a Transcript into a Week of Content

1. Use Cowork when you want Claude to handle the whole job

Use Cowork when you want to hand Claude a full task instead of managing every step. In normal chat, you drive the process one message at a time. In Cowork, Claude works through the task using your project context, connectors, and files, and checks in with you as needed. Cowork can also act on your computer: open a folder, read files, and carry out steps on your behalf. That's what makes it feel like an assistant, not a chatbot.

To open it: in the message bar, **toggle from Chat to Cowork** (the switch sits right next to the Chat button). Then bring in the project you created in Part 1 so Cowork inherits that context.

2. Give Cowork one clear, multi-output goal

The power of Cowork is handing it a single, specific job and letting it carry the work through. In this demo the goal is to turn one transcript into a full content package:

- 7 short video scripts
- 1 long-form video script
- 1 LinkedIn post (the final output later shows multiple LinkedIn posts)

A prompt for that job looks like:

- Find the latest call in the folder
- Turn it into a week of content
- Write a short video script from the strongest point
- Create an outline for a longer video
- Draft a LinkedIn post in your voice

3. Point Cowork to the source file on your computer

The transcript file is stored in the Downloads folder. Tell Cowork to go into Downloads, find the meeting transcript, and use it to generate the content. Cowork can access files on your computer and work directly from them.

4. Send the task and grant access

Press **Send**. Cowork locates the transcript, and because it needs the Downloads folder, it asks for permission first. The interface offers two choices:

- **Ask before acting**
- **Act without asking**

Start with **Ask before acting** for safety. Approve access, and Cowork opens the folder and retrieves the transcript. This is where it begins working inside your computer environment on your behalf.

5. Answer Cowork's follow-up questions

Cowork asks clarifying questions before generating the final deliverable, such as:

- What language should the content be in?
- How should the deliverables be packaged?

Example choices: English, and one combined document. This matters because Cowork needs enough direction to format the output correctly.

6. Review the finished content package

Cowork produces the final result: a full document containing a week of content, pulled directly from the transcript so you never start from a blank page. The output includes:

- 7 short video scripts
- 1 long-form video script
- LinkedIn posts

This is the payoff: one messy call becomes a content system, and one source becomes several client-ready assets.

Part 3: The Full Workflow, Start to Finish

1. Show the end result first

The final output is a full week of content created from one conversation:

- 15 total pieces of content
- 7 shorts
- 1 long-form video
- 7 LinkedIn posts

The point: Claude can turn one strong conversation into a complete content package in your own voice.

2. Understand why the workflow worked

Nothing started from a blank page. Every line came from one conversation. It worked because the project already had the transcript, the right context, the instructions, and the source material loaded in advance.

Claude works best when it already has enough context to act on. Cowork is powerful but can go wrong if you skip that setup. One note: Cowork runs in the desktop app only. On mobile, use Dispatch to connect to Cowork and trigger actions from your phone.

3. Learn what a skill is

A skill is how you teach Claude your specific way of doing a recurring task. Once created, it helps Claude do that task your way every time, and it works across projects, chat, and Cowork. Example: a skill called **Giovannie Labs Content Engine** with rules like:

- No em dash
- No corporate filler
- Specific sentence patterns
- Preferred rhythm and tone

4. Create a skill with Skill Creator

Press **forward slash** and choose **Skill Creator**. It helps you create, modify, and improve skills. Feed Claude startup information, past scripts, and examples of your own writing. Claude may ask follow-up questions, and answering them improves the skill.

5. Let Claude build your voice into the system

Access the skill again with **forward slash**. Over time Claude remembers and loads it automatically when relevant; the first time, load it manually. The goal is consistent output in your voice without re-explaining everything.

6. Edit and refine skills when needed

To edit: click **Customize**, go to **Skills**, find the skill, click the **three dots**, and choose **Edit**. You can also ask Claude to edit it directly. This has saved clients real time, such as cutting prep from 2 hours to 15 minutes.

7. Take the big lesson

AI can turn one call into a week of output, and that output can directly support business growth. The workflow combines context, Cowork, skills, and structured inputs to create content faster and with less blank-page friction.

8. Where Claude Code fits

Claude Code is for software, programs, prototypes, automations, and websites: the engineering layer underneath. Most people don't need it yet if they run a coaching practice, shop, service business, or a regular job. Don't stress about it until you hit the ceiling of the other tools.

9. Recap the full Claude workflow

- **Chat**: define the shape instead of asking like Google
- **Projects**: save context once so future work sounds like you
- **Artifacts**: generate the actual deliverable, like a client-ready plan
- **Connectors**: let Claude read real tools and files without copy-paste
- **Cowork**: hand off the whole job so one call becomes a week of content
- **Skills**: teach Claude your method so the output doesn't sound robotic

10. Start with a simple setup

Open Claude, go to chats, set up a project, upload a few files, include 2-3 of your best examples, and add clear instructions. That's the fastest way to start saving time and improving output.

Watch the full walkthrough

youtube.com/@EmileGiovannie

Work with me

emile@giovannielabs.ai

@EmileGiovannie • AI automation for solo entrepreneurs and small businesses