EPISODE 15 · BLUEPRINT

The Delegation Prompt Pattern

Reconstructed from **Pam Didner**'s Microsoft Copilot agent build, written to run in any AI chat tool · The AI Social Playbook

> **What this is, and what it isn't.** Pam built one custom agent, live, inside Microsoft Copilot's New Agent screen. She spoke her prompt aloud rather than reading it from a script, so the prompts below reconstruct that structure from what she said and from what appeared on screen. They are not a transcript of her exact keystrokes and they are not Copilot's internal prompts. The agent's finished social posts were never generated on camera, so nothing here claims a quality result. Where the episode left a gap, it is marked rather than filled in.

01 Technical Stack

TOOL	ROLE IN THE WORKFLOW	NOTES
Microsoft Copilot (web)	Hosts the chat and the New Agent builder. Writes the agent's instructions, skills, knowledge list and suggested prompts when asked to.	copilot.microsoft.com. Pam used the web version. A desktop app exists and behaves the same way for this workflow.
Copilot custom agent	The artifact being built. A saved, reusable configuration for one repeating task, equivalent to a ChatGPT custom GPT.	Reached through Agent, then New Agent, then Skip to go straight to the builder.
Microsoft 365 knowledge sources	Supplies the agent's reference material: past chats, meeting notes, and files on SharePoint or OneDrive.	Named in the episode as the main difference from Claude and ChatGPT. Requires your organisation to be on M365.
Any general-purpose AI chat tool	Runs System Prompt 1 and System Prompt 3 below. The delegation pattern itself is not Copilot-specific.	System Prompt 2 assumes an agent builder with instruction, skill and knowledge fields.

02 Configuration

Have these ready before you open an agent builder.

A task that actually repeats

The pattern pays off on work you do on a cycle, not on one-off requests. In the episode the task is writing post-event social posts after every event. If you would only run it twice a year, a saved prompt is a better fit than an agent.

The knowledge files Copilot asked for

When asked what to add to knowledge, Copilot named five things: a brand voice guide, a social media policy, photo consent rules, campaign messaging, and the approved handles, hashtags, names and links. Gather these before you build. Pam's own reaction was that she would not have thought to include that much detail.

A living source instead of a static upload M365 ONLY

Point knowledge at a document on SharePoint or OneDrive rather than uploading a PDF, so the agent reads the current version on every run. Past chats and meeting notes can be added the same way. This is the one step that does not transfer to Claude or ChatGPT.

Five real runs budgeted for testing ADDED STEP

Pam's standard is that the agent is not finished at creation. Plan to run it against the next five real cycles and correct the instructions, knowledge or skill based on what comes out. This is stated in the episode but is not a step the builder prompts you to take.

03 Flowchart Map

1. **Describe the problem in plain language** SYSTEM PROMPT 1
   IN Your situation, in your own words, including what you can and cannot supply → **OUT** A structured prompt written by the model, plus its list of what it still needs from you
2. **Answer the model's questions**
   IN The gaps it named → **OUT** A prompt with no invented assumptions left in it
3. **Open the agent builder and skip the wizard**
   IN Agent, then New Agent, then Skip → **OUT** The two-panel screen: chat on the left, agent fields on the right
4. **Delegate all four fields in one prompt** SYSTEM PROMPT 2
   IN One prompt naming the task, the cadence, the outputs and the four fields → **OUT** Instructions, skills, knowledge list and three suggested prompts, filled in on the right
5. **Review what the model wrote**
   IN The generated instruction set and success criteria → **OUT** Corrections where it guessed wrong about your workflow

6 **Attach knowledge and save**

IN The five knowledge files, pointed at living documents where possible → OUT A saved agent. In Copilot this needs the Create button, upper right, or nothing persists

7 **Test against five real cycles** SYSTEM PROMPT 3

IN Real inputs from five consecutive runs → OUT A list of corrections to fold back into instructions or knowledge

8 **Widen the agent rather than building a second one**

IN The working agent → OUT Additional channels or adjacent outputs handled by the same configuration

04 System Prompts

Copy each panel whole. Replace every highlighted variable before running.

SYSTEM PROMPT 1

The Closing Sentence

COPY WHOLE

```
I need help with a task. Here is my situation in plain language.

My role: [YOUR ROLE]
The task I repeat: [DESCRIBE THE REPEATING TASK]
What I can reliably supply each time: [LIST YOUR REAL INPUTS]
What I can never supply: [LIST WHAT YOU WILL NOT HAVE]
What a good result looks like: [DESCRIBE A GOOD OUTCOME]

Now, create a solid prompt for me, structured as:
  1. Input sources
  2. Output rules
  3. Don'ts
  4. Success criteria

Then tell me what information you need from me that I have not
already given you. Do not fill those gaps with assumptions. Ask.
```

SYSTEM PROMPT 2

The One-Prompt Agent Build

COPY WHOLE

```
Can you help me create a custom agent that [WHAT THE AGENT CONSUMES]?
I want to use this agent [WHEN YOU WILL RUN IT]. I will enter the
information you need, and the agent will produce [THE EXACT OUTPUTS].

First, write a prompt with clear inputs, outputs, rules, don'ts,
files needed and success metrics that I can paste into instructions.

Second, create the skills this agent will need to do that job, by
interacting with me and gathering the information from me.

Third, tell me what to add to knowledge.

Fourth, come up with three suggested prompts for me.

Finally, name this agent [AGENT NAME].
```

SYSTEM PROMPT 3

The Five-Run Correction Loop

COPY WHOLE

```
You are reviewing your own output from a live run of this agent.

Here is what I gave you as input this time:
[PASTE THE REAL INPUT YOU SUPPLIED]

Here is what you produced:
[PASTE THE OUTPUT]

Here is what I had to change before I could use it:
[DESCRIBE YOUR EDITS]

Based on that gap, tell me which single change would have prevented
it: a change to the instructions, a missing knowledge file, or a
change to the skill. Name one, not three. Then write the exact
replacement text for whichever one you named.
```

05 Variable Callouts

VARIABLE	WHAT IT IS	WORKED EXAMPLE
[YOUR ROLE]	Enough context for the model to pitch the output correctly	Social media manager at a mid-market manufacturer, reporting to the marketing director
[DESCRIBE THE REPEATING TASK]	The cyclical job, named with its real trigger	Writing social posts after every trade show our sales team attends
[LIST YOUR REAL INPUTS]		

	What genuinely arrives each cycle, not what you wish arrived	Three to five phone photos, a booth number, and sometimes a voice note from a rep
`[LIST WHAT YOU WILL NOT HAVE]`	The honest gaps, so the model asks instead of inventing	No attendee names, no quotes, no permission to show customer faces
`[DESCRIBE A GOOD OUTCOME]`	Your own standard, in your own words	Posts a rep would recognise as the event they actually attended
`[WHAT THE AGENT CONSUMES]`	The raw material the agent scans or reads on each run	Scans images from our sales reps and event managers
`[WHEN YOU WILL RUN IT]`	The cadence, which is what makes it an agent rather than a prompt	Right after each event
`[THE EXACT OUTPUTS]`	Count and destination, stated precisely enough to be checked	Three social media posts, for LinkedIn, Facebook and Instagram respectively
`[AGENT NAME]`	A name you will recognise in a list six months from now	Post Event Social Agent
`[PASTE THE REAL INPUT YOU SUPPLIED]`	The genuine input from one test run	Four photos from the Chicago show plus a 20-second voice note
`[PASTE THE OUTPUT]`	What the agent returned, unedited	Three drafts, one of which named a company that was not at the booth
`[DESCRIBE YOUR EDITS]`	What you fixed by hand, which is the signal	Removed the invented company name and rewrote the Instagram caption to drop a hashtag we do not use

06 Human Quality Control

Three checks before anything ships. Each one catches a specific failure this workflow produces.

Check 1. Every name, quote, handle, link and photographed face in the output traces back to something you actually supplied.

CATCHES Invented attribution. The agent is working from thin input by design, which is exactly the condition under which a model fills gaps with plausible detail. Copilot wrote this check into its own success criteria, including flagging unverified permissions, which is the one that matters when a customer's face is in the frame.

Check 2. The generated instructions describe your workflow, not a generic version of it, and you have corrected at least one thing in them.

CATCHES Accepting a confident spec for a job the model has never watched you do. A first pass that needs no corrections usually means it was not specific enough to be wrong, which is the failure that only surfaces several runs later.

Check 3. You have run the agent against five real cycles and folded each correction back into the configuration before treating it as finished.

CATCHES The abandoned agent. Building it is the satisfying part and the part people stop at. Pam is explicit that testing is the owner's job and that the instructions, knowledge or skill will need changing based on what the first five runs produce.