Session 4 Handout: Building Reusable Helpers

FAST-02 Generative AI for Everyday Work · 7formations · Instructor: Wadood Chaudhary · All examples fictional

Why this session matters

Once a prompt sequence works, retyping it every week wastes time and invites drift. Saved helpers (agents, GPTs or plugins, Gems or skills, projects) keep your instructions and sources in one place so you and your coworkers get the same rules every time. This session shows how to write helper instructions anyone can run, how to build them in your approved tool, and how to prove (or disprove) that they save time.

Key concepts

1. What a helper is, and isn't

A helper is a saved set of **instructions** plus optional **reference files** that the AI applies every time you use it. It is not a retrained model and it does not "learn" your business; it repeats what you wrote. If the instructions are vague, the helper is consistently vague. And a helper never removes the need to check output.

2. The platforms (as of October 2026: a period of change)

Platform	Helper feature	Status note
Microsoft Copilot (formerly titled Microsoft 365 Copilot)	Agents built with **Agent Builder**, grounded on SharePoint/OneDrive files, or in Copilot Studio for advanced builds	Agents surface only files the *user* can access; over-sharing matters
ChatGPT Business / Enterprise	**Custom GPTs**, being replaced by **plugins** whose instructions are a **skill** plus reference files	OpenAI retires custom GPTs on December 11, 2026 (February 11, 2027 for some workspaces); new GPT creation ended October 26, 2026 for affected Enterprise workspaces. Migrate and retest
Gemini (Google Workspace)	**Gems** with knowledge files; **NotebookLM** for source-grounded notebooks	Google is moving Gems on personal accounts to "skills" from November 2026; check your Workspace admin notice
Claude for Work	**Projects** with custom instructions and project knowledge	Stable concept; check plan limits

Because names and menus are changing, write the **specification** first. A good spec moves between platforms in minutes.

3. Instructions a coworker can reuse

A helper spec has seven parts:

1. **Job and reader:** one sentence. "Extract supplier quotes into the comparison table for the purchasing lead."
2. **Sources:** which reference files, and "use only these plus what the user uploads."

3. **Format:** exact columns, sections, or length.
4. **Missing-fact rule:** "Leave blank" or "write 'not in source.'"
5. **Limits:** what it must not do ("no totals or recommendations").
6. **Check step:** a line the helper adds to every output reminding the user what to verify.
7. **Refusal rule:** what to do if someone pastes restricted data.

The cold test: a partner runs the helper on a new example without asking you anything. Every question they wanted to ask is a missing instruction.

4. Measuring value honestly

Time the task by hand at the same quality bar. Then time it with the helper, **including checking and fixing**. Count errors you caught and errors found later. Write a quality note. Decide: keep, adjust, or stop.

Example: by hand 45 minutes; helper 6 minutes plus 25 minutes of audit = 31 minutes. Honest saving: 14 minutes, with an audited table. If checking had taken 40 minutes, the right answer would be "stop": that is a good outcome, not a failure.

Worked example: helper spec (paper version)

```
NAME: Bayside Quote Extractor
JOB: Extract supplier quotes into the 8-column comparison table for the
purchasing lead.
SOURCES: Use only the quotes the user uploads in this chat and the
"Comparison template" reference file.
FORMAT: Exactly the template's 8 columns; one row per quote; units as
in the header; Source column = quote letter + line.
MISSING: Leave the cell blank. Never estimate.
LIMITS: No totals, no rankings, no recommendations.
CHECK STEP: End every reply with "Audit every cell against the quotes;
do the math in the spreadsheet."
REFUSAL: If the user pastes bank details or personal information, say
it must be removed and do not process it.
```

Checklist: helper ready to share

- ☐ Built in the employer-approved tool with a work account.
- ☐ All seven spec parts present.
- ☐ Reference files contain no restricted data.
- ☐ Partner cold test passed; instructions improved.
- ☐ Before-and-after log includes checking time.
- ☐ Owner named for updates and for migration if the vendor changes the feature.

Do and don't

Do	Don't
Write the spec before clicking "create"	Build by trial and error with no written spec
Keep instructions positive and concrete	Write a single line like "be accurate"
Put rules in instructions, facts in reference files	Mix policy rules into uploaded files
Retest after any vendor migration or model change	Assume a migrated helper behaves the same
Count checking time	Report only the AI's run time

Try this at work

1. List three recurring tasks. Pick the one with the clearest source.
2. Write the seven-part spec on paper.
3. Build it in your approved tool and have a coworker run it cold.
4. Log two real runs with checking time. Decide keep, adjust, or stop.
5. If you use ChatGPT GPTs or Gemini Gems, check this month whether your helpers need migrating.

Glossary

Term	Meaning
Helper	Saved instructions plus reference files reused across chats
Agent (Microsoft)	A Copilot helper built with Agent Builder or Copilot Studio
Plugin / skill (OpenAI)	The replacement for custom GPTs; a skill holds reusable instructions
Gem (Google)	Gemini's saved helper; moving to "skills" on personal accounts
Project (Anthropic)	Claude workspace with custom instructions and knowledge
Cold test	A partner runs the helper without help from its creator
Before-and-after log	Record of time and errors with and without AI, including checking

Key takeaways

- A helper repeats your rules; it doesn't replace your check.
- Spec first: job, sources, format, missing-fact rule, limits, check, refusal.
- Measure honestly, including checking time.

Vendor details verified against OpenAI, Google, and Microsoft help pages on October 29, 2026. These features are changing quickly; confirm before each use.