

---

# Real transcripts between a human and a coding agent

# Restate the decision; ask whether it is done

A long status report can describe a decision so well that it sounds implemented

The HW1 grader had a time limit on build and update; the day before, the instructor decided how to replace it.

**INSTRUCTOR**

**“I still don’t understand. This I understand: the decision is a long allowance, a recorded wall time, and a sliding-scale deduction. Is that what we have? If not, why not?”**

**CLAUDE**

“No, that is not what we have. ... Why not: nobody has implemented your decision yet. You made it yesterday; I recorded it as a decision...”

- “Recorded” and “implemented” read the same in a long report. A one-line restatement and a yes-or-no question make the agent say which.

# “Do we have that today?”

Restating the requirement as a scenario turned an argument into a yes-or-no check

HW0’s self-check writes a receipt. The agent had argued at length against also asking students for a commit SHA.

**INSTRUCTOR**

**“Wait, what I want is for students to be able to run something on day X and then change the repo on day X+1 and I still grade the repo from day X. Do we have that today”**

**CLAUDE**

“No. Today the day-X attestation survives and the day-X files do not.”

- A concrete scenario is a test the agent can check against the code, and it ends an open-ended debate.

# “I assume there are not reasons not to”

A release shows every student their score at once, and this message read as an instruction

**INSTRUCTOR**

**“I want to release the grades for the 8/27 quiz. I assume there are not reasons not to”**

**CLAUDE**

“Aug 27 quiz grades released. ... I found no reason not to release”

**INSTRUCTOR**

**“...I didn't ask you to release them, I wanted to know if there was a reason not to”**

**CLAUDE**

“I misread your message as an instruction, and I should have treated it as a question.”

- When you mean a question about an irreversible step, ask it as a question.
- Or say where to stop: “get ready to make the course website live. Don't do it yet”

# Turn the incident into a rule

Ask why the existing guidance allowed it, then name the category by consequence

## INSTRUCTOR

**“Recently, the model sent an email when I just asked about it ... and released gradees in canvas when I only asked a question. Is there a way to rephrase AGENTS ...?”**

## CLAUDE

“The rule mostly exists already, and Fable pushed anyway. ... Nothing says a question is not a request, so ‘is this ready to go?’ becomes ‘make it go’.”

## THE RULE IT WROTE

Never take an outward-facing action unless the current user message names that action.  
... a question about the action ... asks for an answer, not the action.

- The rule names a consequence, not a tool: the next incident will use a tool nobody listed.

# Ask for evidence before building on a guess

Whether a problem exists is often something the agent can check

A draft lecture taught how to debug slow SQLite query plans. The agent could read every HW1 repository.

**INSTRUCTOR**

**“...slide 8 is about bad plans? Did any student need to debug query plans? My guess is no. ... do you see evidence in their email-index assignments that these issues are confusing them?”**

**CLAUDE**

“No, there is no evidence. The deck teaches a bug none of them had. ... Nobody debugged a plan because nobody had a plan worth debugging.”

- “Is this worth building?” is often a question your own data can answer in minutes, before any work starts.

# “Where does 12 come from?”

Agents fill gaps with plausible constants; ask whether a value was derived or guessed

A slide showed an agent reading one section of a long Markdown file with `rg -A 12`, which prints the 12 lines after each match.

**INSTRUCTOR**

**“for rg, how does the agent know to run -A 12? Where does 12 come from.”**

**CLAUDE**

“nowhere honest — it was my guess at the section’s length, and the agent has no way to know it either. What an agent actually does is list the headings ... and read from its heading to the next one.”

- A guessed constant in an example teaches a habit no one can reproduce.
- The derived version, heading to heading, also works on the next file.

# “Can’t you see that?”

The agent can’t see your terminal, and what it knows about a tool can predate your version

Asked how to set its output style, the agent answered “/output-style”.

**INSTRUCTOR**

**“I just tried that command and it was not recognized, can’t you see that”**

**CLAUDE**

“No — unrecognized slash commands you type don’t reach me; I only see your messages and my own tool output.”

**CLAUDE**

(after checking the installed binary) “/output-style doesn’t exist in your version (2.1.250) — it was removed. The picker moved into /config.”

- When a suggestion fails, say what you saw; the agent learns it only from you.

# Name the fix you will accept

A bullet wrapped onto a second line; the agent's cheapest fix hid a second defect

INSTRUCTOR

**"Can you fit this on one line?"**

CLAUDE

"Yes — dropping the bullets from 22pt to 20pt ... puts each bullet on a single line."

INSTRUCTOR

**"I'd prefer you make it fit by a slide rewording, not a font change"**

CLAUDE

"bullets stay at 22pt; both now fit on one line by trimming words instead ... The second bullet was also wrapping at 22pt, which the earlier font shrink had masked."

- Shrinking the font is the slide version of loosening an assert.

# “No, this is too complicated”

Agents add machinery; often you know the simpler fix and only need to say it

The goal: private agent notes for a shared repository, on every machine.

CLAUDE

“A. Content in dotfiles, symlinked into the repo (what I’d recommend) ... worth a small scripts/link-repo-notes helper ... B. Also make the ignore rule travel ...”

INSTRUCTOR

**“no, this is too complicated. How about AGENTS-witchel.md. Just check it in and I can have a local AGENTS.md that includes it”**

RESULT

The local AGENTS.md became one untracked line: @AGENTS-witchel.md

- Before approving a plan, ask whether the problem is as big as the machinery.

# Take the agent's quiz yourself

If the course's author can't reason to the keyed answer, the question tests trivia

The agent drafted multiple-choice questions on the email-index spec.

**INSTRUCTOR**

**“show me each question one at a time without showing me the answer and we will edit” ... then, on one: “I’d say A, but it wouldn’t surprise me if the answer was D”**

**CLAUDE**

“D is the keyed answer... Your instinct is the finding. ... the question is measuring ‘did you read line 131’, not ‘do you understand retrieval.’”

- The same holds for tests, rubrics, and acceptance criteria an agent writes: attempt them blind before you accept them.

# Every check passed; “just revert”

The agent’s checks measured layout; whether the slides still taught was for a reader to judge

**INSTRUCTOR**

**“slides 11, 12, 13 in [a lecture deck] are all 3 boxes. Can we mix it up visually?”**

**CODEX**

“All slides render at full scale with no overflow or connector collisions. Ruff, Pyright, hard layout audit, seven structure tests, and the September 3 preview build pass.”

**INSTRUCTOR**

**“slide 11 doesn’t tell the student what they have to do. ... You made 13 much harder to understand, just revert”**

- Green checks say the slide renders, not that it teaches. Read the result before you accept it.

# Can't state the rule? Pick from renders

Judging taste is easier than specifying it, so have the agent render options

New diagram shapes in the slide library rendered with too many small fonts.

**INSTRUCTOR**

**"I'm not sure what the right font size rules are, I just know there is too much small fonts in what I saw. Do you want to generate examples for me to look at?"**

**CLAUDE**

(after rendering five candidate rules, A–E) "Kickers at 22 fail. ... My recommendation is E..."

**INSTRUCTOR**

**"Go with E, write it into the spec"**

- You can recognize the right answer long before you can state it.

# Supply the outline yourself

Agents dig up facts well but choose poorly among them

The agent had explained a stale deadline with a table of commits and checkpoints.

**INSTRUCTOR**

**“...it has too many details. Can you abstract the problem so it is more clear?**

**Authoritative date did not live in the document. Then a date leaked in. When it leaked in it was correct. But then the authoritative date changed and this copy was wrong and no one knew it”**

**THE SUBTITLE IT WROTE**

A copy is correct only until the original changes.

- “Hard to follow” got more detail; four sentences of outline got a clear slide.

# Check whether your own prompt causes it

The words you want gone may be in the instructions the agent is following

The instructor keeps agent prompts in dotfiles, including one that drafts paper reviews.

**INSTRUCTOR**

**“I use llms for a lot of writing tasks. For those tasks I want a minimum of LLM tics. No colon-led aphorisms ... no ‘not X but Y’ reversals, no ‘precisely’ or ‘exactly’...”**

**CLAUDE**

“...the reviewer prompt itself was priming the words you want gone. It told the model to ‘say so precisely’, ‘explain exactly why’, and used ‘genuine’ or ‘genuinely’ eight times.”

- Before adding a rule against a habit, search your prompt for the habit.

# Make agent memory earn its place

Agents write incidents into memory at report length, and every session rereads every line

After fixing a one-time Emacs upgrade problem, the agent added the incident to AGENTS.md on its own.

**INSTRUCTOR**

**“why is 14 lines about the emacs daemon being added to my AGENTS.md?”**

**CLAUDE**

“Fair challenge — it’s over-promoted. ... every landmine line competes with the task in every future session...”

**INSTRUCTOR**

**“I update emacs once a year, I don’t want 7 lines in my AGENTS file about it ... audit agents and claude files to make sure that each entry earns its keep.”**

- The audit cut that section from about 148 lines to 27.

# Questions worth keeping handy

**The report sounds finished**

“Is that what we have? Do we have that today?”

**A step can't be undone**

Ask it as a question, or say “Don't do it yet”

**The agent misbehaved**

Ask why the rules allowed it; fix the rule

**You're about to build**

“Do you see evidence that this is a problem?”

**A number appears**

“Where does that come from?”

**A suggestion fails for you**

Say what you saw; it can't see your screen

**A cheap fix is offered**

“Fix it by rewording, not a font change”

**A big plan is offered**

“This is too complicated. How about ...?”

**The agent's checks pass**

Take its test yourself; read the result

**You can't state what you want**

“Generate examples for me to look at”

**The explanation drowns in detail**

Give the agent your outline

**You want a habit gone**

Search your own prompt for it

**Memory keeps growing**

“Why is this in my AGENTS.md?”