Coding Agents for Expert Work

A practical start for experts, creatives and decision-makers beginning serious work with coding agents.

AUTHOR	PUBLISHED	VERSION	FORMAT
Florian Gitt	13 August 2026	1.0	GUIDE / 10 MIN

Practical guidance only. This resource is not certification, legal advice or a compliance guarantee.

A practical start

The productive shift is not a clever prompt.

A coding agent can research, analyse, write, edit files, run programs and check its own output. The productive shift is learning to give it a bounded working environment, durable context and a clear standard for review.

Use this guide for an initial workflow that is small enough to inspect completely and useful enough to teach you something real.

1. Start with work you understand

Choose a real task where you can recognise a good result and catch a bad one. A source review, dataset check, document transformation, research memo or first prototype is better than a vague request to improve the business.

- Name the decision or artifact the work must support.
- Define what the agent may read, change and communicate.
- Keep the first run small enough to inspect completely.
- Avoid irreversible actions, sensitive systems and autonomous external communication.

2. Give it a workspace, not only a prompt

Serious work needs persistent sources and rules. Put the brief, reference material, terminology, constraints and output location in a workspace the agent can inspect. Record which file or system is authoritative when several versions exist.

A USEFUL MINIMUM	Objective / sources / constraints / permissions / output / review standard

3. Separate exploration from authority

Agents are strong at generating options, tracing sources and producing drafts. They should not silently inherit authority to publish, send, buy, delete or change production systems. Use explicit gates before consequential actions.

STAGE	PURPOSE
Explore	Inspect, search, compare and identify uncertainty.
Draft	Produce the artifact or proposed change locally.
Verify	Test facts, calculations, links, rendering and side effects.
Approve	A responsible person decides whether the result may cross the external boundary.

4. Make evidence travel with the result

Ask for source links, file paths, assumptions and checks close to the claims they support. For data work, preserve raw inputs and reproducible transformations. For writing, distinguish quotation, paraphrase, inference and opinion. For software, run the real build and tests rather than accepting plausible code.

5. Evaluate with cases from real work

Build a small set of representative tasks, including difficult and failure-prone examples. Define the expected qualities before running the agent. The evaluation can be simple, but it should be repeatable.

QUESTION	WHAT TO RECORD
Was the answer correct?	Errors, omissions and unsupported claims
Was the process safe?	Data accessed, actions taken and gates respected
Was the artifact useful?	Expert review, revisions and decision impact
Did the context help?	Sources or instructions that changed the result

6. Keep human judgment visible

Assign an owner for the outcome. The reviewer needs enough domain knowledge and time to examine the material, not merely approve it after the fact. If nobody can evaluate the result, narrow the task or bring in someone who can.

7. Improve the system, not the mythology

When something fails, ask whether the problem came from the task definition, source quality, missing context, permissions, the tool, the model or the review process. Save useful instructions, scripts and evaluation cases. Remove scaffolding that no longer earns its place.

First-session protocol

- Choose one bounded task and write the intended outcome in one sentence.
- Place the relevant sources in a controlled workspace.
- State what the agent may and may not change.
- Ask it to inspect first and explain its proposed approach.
- Let it produce a local draft or artifact.
- Check the evidence, output and side effects yourself.
- Record one failure case and one improvement for the next run.

> **THE AIM IS NOT AUTONOMY FOR ITS OWN SAKE**
>
> It is a working relationship in which the agent expands what an expert can examine and make, while responsibility and correction remain legible.

Canonical HTML: https://wirenet.dev/resources/coding-agents-for-expert-work