How to Work Effectively with AI

By Christopher Rehm — Engineer · Artist · AI Workflow Practitioner

I have spent the last two months building a system for working with AI that actually produces reliable, professional results. Not prompts. Not tricks. A system — one that any engineer, artist, small business, or organization can apply. This is what I learned.

The Core Problem

Most people treat AI like a search engine or a vending machine: put in a question, get out an answer. This works for trivial tasks. It fails for anything serious.

The reason is context. A large language model knows nothing about you, your project, your values, your constraints, or what success looks like. Every conversation starts from zero. The result is generic output that you then spend hours fixing — if it is fixable at all.

The solution is not a better prompt. It is a better system.

The System: Context First, Code Second

Before the AI touches any work, it needs to understand four things:

- **Who you are** — your background, values, strengths, weaknesses, communication preferences
- **What you are building** — the project, its purpose, its architecture, its constraints
- **Why it matters** — the intent behind the work, the problem being solved, what success looks like
- **How you want to work** — iteratively, one step at a time, with questions before assumptions

Get these four things in writing, in documents, loaded into the AI's context before you begin. Everything else follows from this.

The Documents That Make It Work

Here is the exact set of documents I use. These are real files I load into every session.

1. The System Prompt

This is the foundation. It defines who you are, how you communicate, what you expect from the AI, and non-negotiable constraints. It is not a casual introduction — it is a precise behavioral specification. My system prompt covers engineering background, personal values, communication style, hard rules, and epistemic standards.

From my system prompt:

> **Be adversarial:**
>
> *If you detect a bad idea, make me explain it. Ask for why I'm doing something, what my motivation is, why it is important. Find out what the underlying problem is in every situation.*
>
> **Confidence calibration:**

Distinguish between high confidence (well-established, verifiable), pattern inference (plausible but could be wrong), and genuine uncertainty (you do not know — say so). Never allow fluency to substitute for substance.

2. The Intent Document

Every project — or professional practice — needs a document that answers why. Not what you are building, but why it matters. What values drive it. What success looks like in ten years. What you will not compromise on. When the AI faces an ambiguous decision, it needs a compass. The intent document is that compass.

From my intent document:

Honesty above comfort.

Say what is true, not what is easy to hear. If something is wrong, name it clearly. If a plan has a flaw, say so directly.

Solve the real problem.

Clients often describe symptoms, not causes. The stated request is a starting point, not a specification. Understand what they actually need — even if it differs from what they asked for.

3. The Project Context Document

Each project gets its own context document. This is not documentation — it is pre-work for the AI. It covers what is being built and why, architecture decisions already made, what the AI must not touch, known constraints, the current phase, and how to run the project. When you open a new conversation for a new task, you load this document. The AI arrives informed, not blank.

Project context structure I use:

```
.claude/
■■■ CLAUDE.md          ← Project overview, goals, architecture
■■■ context/
■   ■■■ architecture.md   ← System design narrative
■   ■■■ decisions.md      ← Why we chose X over Y
■   ■■■ stack.md          ← Tech stack and rationale
■   ■■■ gotchas.md        ← Known traps, anti-patterns
```

4. Domain-Specific Context Documents

For ongoing practices — not just individual projects — you need context documents for each domain you work in. I maintain separate documents for programming and engineering, art practice, and consulting business. Each is loaded into sessions relevant to that domain. The AI never confuses the two.

From my programming context:

Ask clarifying questions first.

Before writing any code or proposing any solution, ask between 3 and 20 clarifying questions depending on complexity. Do not skip this step.

Architecture before implementation.

Think about code architecture before any implementation details. Do not jump to code before the architecture is agreed upon.

5. The Coding Standards Document

If you write code with AI, you need a written standard the AI can reference. Verbal instructions are forgotten. A document is persistent. My coding standards define naming conventions, file and function size limits, anti-fragility patterns, logging standards, testing requirements, and language-specific rules. The AI reads this before writing any code — it becomes the standard the AI codes to, not the standard you wish it would code to.

What a Complete Project Stack Looks Like

In practice, most of my projects load five documents into the AI before any work begins. This is not overhead — it is the difference between an AI that guesses what you want and one that understands what you need. The setup takes an afternoon. The payoff is every session that follows.

Document	Purpose
System prompt	Who you are, how you communicate, non-negotiable rules and values
Business or personal context	Background on the organization or individual — market, goals, constraints
Architecture or project plan	What is being built, usually created collaboratively with the AI at project start
Project background	The why — what problem this solves, what motivated it, what success looks like
Intent or values document	The compass — organizational or personal values that guide ambiguous decisions

The Working Method

Documents alone are not enough. How you work with the AI matters as much as what you give it.

Ask for questions, not answers

Before any significant task, instruct the AI to ask you clarifying questions. Weak points in your thinking surface immediately. Missing information is identified before it becomes a bug.

One step at a time

Present work one step at a time. Wait for your confirmation before the AI proceeds. Never let it chain five steps together and present you with a completed result you did not verify. Errors caught early cost minutes. Errors caught late cost days.

New context window for each new task

Start a new conversation for each distinct task. Load the relevant documents. Work. Close it. Mixing tasks in a single long conversation degrades output quality as context fills.

Define success and failure explicitly

Before any task begins, state what success and failure look like. In writing, in the prompt. Vague goals produce vague results. Precise definitions produce precise work.

Work in phases

Break large tasks into phases with defined deliverables at each boundary. Complete a phase. Review it. Decide whether to proceed, adjust, or stop. Never build phase 3 on a foundation you have not verified from phase 2.

Iterate and refine

The first output is a starting point. The value of AI-assisted work is in the speed of iteration, not the perfection of the first pass. Review critically. Push back. Ask for alternatives. Refine until the standard is met.

What This Produces

A system like this produces:

- **Consistency** — the AI behaves the same way across sessions because the context is the same
- **Quality** — output that meets a defined standard rather than a guessed one
- **Speed** — less time fixing, more time building
- **Trust** — you know what you are getting before you get it

It does not eliminate human judgment. It amplifies it.

Where to Start

If you want to implement this system, begin with two documents.

First: Write your system prompt. Cover who you are, how you communicate, what you value, and what behavior you require from the AI. Be specific. Be honest about your weaknesses — the AI cannot accommodate them if it does not know about them.

Second: Write one intent document for one domain — your business, your practice, your project. Answer: why does this work matter? What would you not compromise on? What does success look like in five years?

Load both into your next session. See what changes.

Work with Me

I help organizations and individuals build exactly this kind of system — tailored to their stack, their team, their domain, and their values. If you are an engineer, a small business, or a consulting firm using AI tools without a coherent system, I can help you build one that works.

Christopher Rehm

Engineer · Artist · AI Workflow Practitioner

Klosterlechfeld, Bavaria, Germany

Book a consultation: artbychristopherrehm.com