AI Development With Spec-Kit:

Turning "prompt and pray" into an actual engineering discipline

Jeff Richley

November 20, 2025

A live demo featuring pandas, penguins, and at least one otter who tried to sabotage my talk last time.

Why AI Development Feels Chaotic

AI projects often go off the rails because:

- Requirements drift faster than your attention span during a Zoom meeting
- All the "context" is scattered across Slack, GitHub, email, sticky notes, and that one brilliant thought you had in the shower
- The AI confidently generates code... for a completely different problem
- There is absolutely *not* a repeatable development cycle — unless "winging it" somehow counts

Goal: Build AI software *without* chaos. (Finally.)

What Spec-Kit Actually Does

Spec-Kit gives you:

- A workflow that gently forces your AI to act like it's seen a project plan before
- A clean separation between *what you want*, *how you want it*, and *how badly the AI will try to misunderstand it*
- Repeatable, auditable steps — so future-you can't yell at past-you
- A simple system that works with any AI coding agent (even the ones with "creative" tendencies)

Not heavy. Not bureaucratic. Just enough structure to keep things sane.

The Four Pillars (a.k.a. The Sacred Scrolls)

Spec-Kit consists of:

1. **/specify** — Define *what* you want and *why*, before the AI goes rogue
2. **/plan** — Decide *how* you want it built, so the AI doesn't freestyle jazz your architecture
3. **/tasks** — Break the project into tiny steps the AI cannot possibly misinterpret
4. **/implement** — Release the AI to generate code... but with *responsible adult supervision*

The Demo: "Pick-a-Zoo"

Previously on... Pick-a-Zoo:
While planning this talk, the otters staged a small yet passionate union rally. They demanded more screen time, better lighting, and dental.
Negotiations are... ongoing.

What we have so far:

- **Live-stream switcher** for real animal cams
- **Cast:** elephants 🐘, hippos 🦛, otters 🦦 (Local 404), penguins 🐧, tigers 🐅
- **Snapshot button** — capturing "evidence" since day one

What we're building today (live):

- **Time-lapse feature** — otter chaos at 60× speed

Cute animals + structured AI dev = today's adventure.

Step 1: /specify — The Spec Before the Chaos

Capturing requirements clearly — you know, like actual engineering:

- **What are we building?**
  A real description, not "something with AI" scribbled on a napkin.
- **Why does it matter?**
  If you can't answer this, congratulations: you've discovered scope creep.
- **What does success look like?**
  Measurable, testable, and the opposite of "vibes."

In real engineering, specifications define behavior, constraints, and acceptance criteria.
/specify does the same thing — just without the 47-page PDF.

No tech. No code. Just clarity before the AI starts free-styling.

Step 2: /plan — The "Big Brain" Phase

This is where we pretend we're doing architecture diagrams on purpose:

- **Architecture**
  What the system should look like, not whatever the AI hallucinated at 2 a.m.
- **Tech stack decisions**
  Picking tools intentionally — not because they appeared in a YouTube thumbnail.
- **Data flow**
  How information is *supposed* to move... instead of whatever mysterious path it took last time.
- **Constraints & options**
  Performance, deployment, security — the adult stuff.

/plan is the AI's blueprint. Without it, the AI builds you a treehouse when you asked for a garage.

Step 3: /tasks — Tiny Steps, Huge Payoff

Tasks exist to prevent chaos and emotional damage:

- **Atomic, buildable steps**
  If a task takes more than one breath to read, break it up.
- **Dependency order**
  Because "surprise prerequisites" are how projects die.
- **Status tracking**
  (Todo / In Progress / Done / "Why did Past-Me do this?")
- **A roadmap the AI can actually follow**
  Think turn-by-turn GPS... not "good luck, buddy."

If a task is unclear, the AI will absolutely produce unclear code — and it will do so with confidence.

Step 4: /implement — Where We Let the AI Cook

Where the "magic" happens... (and by magic, we mean carefully supervised automation):

- **Trigger tasks one at a time**
  Giving the AI *all* the tasks at once is how horror movies start.
- **AI generates code guided by spec + plan**
  Finally using the instructions we so lovingly crafted.
- **Inspect results**
  Trust is great... but verification prevents therapy bills.
- **Iterate cleanly**
  Fix, refine, repeat — not "burn it down and start over."

A simple loop: Specify → Plan → Tasks → Implement.
Rinse. Repeat. Minimal screaming.

The Takeaway

Spec-Kit transforms AI development by:

- **Providing clarity and structure**
  Because "guess what I meant" is not a development strategy.
- **Enabling repeatable workflows**
  You know... like real engineering.
- **Reducing rework and hallucinated code**
  The AI will still hallucinate — just less, and more politely.
- **Making AI coding assistants actually useful**
  Instead of creative writers accidentally doing software engineering.

It's time to retire "prompt and pray." We're doing actual engineering now.

Thank you!

You've been an amazing audience — significantly less chaotic than my AI agents.

Questions?

(Preferably about Spec-Kit... but I am also prepared for otter-related inquiries.)

If your question is "Why did the AI do that?", the answer is: because you trusted it.