

Claude Code

<https://devSteve.com/blog/posts/claude-code.html>

September 12, 2025

Introduction

If you're a software developer and you haven't tried an agentic AI coding assistant like Claude Code, then now might be the time to give Claude a try. I'm still a relative newcomer to the world of AI "vibe coding" but fortunately it's a pretty short path from zero to hero.

This blog and the parent devSteve.com website were both vibe coded with Claude in less than a couple of hours. The words and content are provided by me, but the HTML, CSS, and JavaScript - along with the devSteve logo and corresponding "dS" favicon - were all generated by Claude.

The results aren't always perfect. When trying to vibe code some more advanced 3D interfaces Claude struggled to provide working code. For things like basic websites, React apps, or backend server API's, Claude seems to do great. It also seems to be good at things like writing unit tests.

This post is going to outline my current workflow when using Claude Code. It is my hope that someone finds this information helpful. This workflow is based on other online tutorials and YouTube videos for which I no longer have direct links (my apologies to the original authors for not linking back to their content).

Workflow

I've used the following Claude Code workflow for my last three projects and it seemed to produce good results. In short, the workflow consists of the following six steps:

1. Create a PRD (Project Requirements Document)
2. Create a CLAUDE.md file
3. Create a PLANNING.md file
4. Create a TASKS.md file
5. Update the CLAUDE.md file
6. Have Claude build the project

Five out of six of the items are just basic prompts with standard templates. The other item just requires making a small text change to the CLAUDE.md file. That's it. It's surprisingly simple.

1. Create a PRD

This step is going to require the most effort on your part. When asking Claude to generate a PRD you'll need to provide a description of the project. This description is the seed of the whole project.

To kick things off, use the following PRD prompt and replace <project-description> with a description of the project you would like Claude to build. If you run through this workflow and you aren't happy with the code produced by Claude, you might want to come back and update your project description and try again.

<prompt redacted>

When writing your project description be as detailed as you'd like. This is also the place where I tell Claude which tech stack I want it to use, and any other technical guidelines. Here is the exact prompt I used to generate the layout for this blog:

<prompt redacted>

Before continuing to step 2 you might want to take a few moments to read through the PRD that Claude generated and make any necessary tweaks. Remember, the PRD is the seed from which the project will sprout. Software development is a chaotic process and chaotic systems are *extremely sensitive to initial conditions*.

2. Create a CLAUDE.md file

The next step is to ask Claude to generate a CLAUDE.md based on the information in the PRD.

<prompt redacted>

No changes need to be made to this prompt. I copy-and-paste it as-is each time I use it. In step 5 we'll make a small change to CLAUDE.md.

3. Create a PLANNING.md file

Next we ask Claude to generate a PLANNING.md file.

<prompt redacted>

Similar to the PRD, you might want to take a minute to review the PLANNING.md file and make any desired changes. This file will also have a large impact on what ultimately gets produced by Claude.

4. Create a TASKS.md file

Next we ask Claude to generate a TASKS.md file.

<prompt redacted>

Claude seems to know to read information from the other files we've created, and it will proceed to come up with milestones and tasks for the project. Again, you might want to take a few moments to review the tasks created by Claude, and make any desired updates before continuing to the next step of the workflow.

5. Update the CLAUDE.md file

Before asking Claude to build the project, add the following text to the top of the CLAUDE.md file (I usually add it just before the "Project Overview" section).

<text redacted>

The primary goal of this workflow is to give Claude as much context as possible, and also to have Claude use files and conventions that will reduce hallucinations and generally keep Claude on track.

6. Have Claude build the project

This is your last chance to review the project files before Claude does its thing. When you're ready, use the next prompt to have Claude build the project.

<prompt redacted>

Claude will begin building the project and will usually pause at the end of each milestone. To continue I simply tell Claude to "Continue to the next milestone." and Claude will get started on the tasks in the next milestone. Repeat the "continue to next milestone" prompt until Claude thinks its finished the project.

In my experience the code isn't always perfect on the first try. When updating an old React app, for example, Claude reported that it was finished even though the code wouldn't build. In these cases my go-to prompt is always:

<prompt redacted>

Even with specific instructions like "fix the build errors", Claude will still sometimes report being done even though the code still won't build. Repeating the "fix the build errors" prompt a few more times usually does the trick, and with enough tries the code will eventually build successfully.

Conclusion

The workflow's simplicity is striking. I've only used it for a handful of projects so far, but this will be my default project workflow for the foreseeable future. The results aren't always perfect, but if Claude can knock out a large portion of the work with just a few prompts, why *wouldn't* I start all new projects this way?