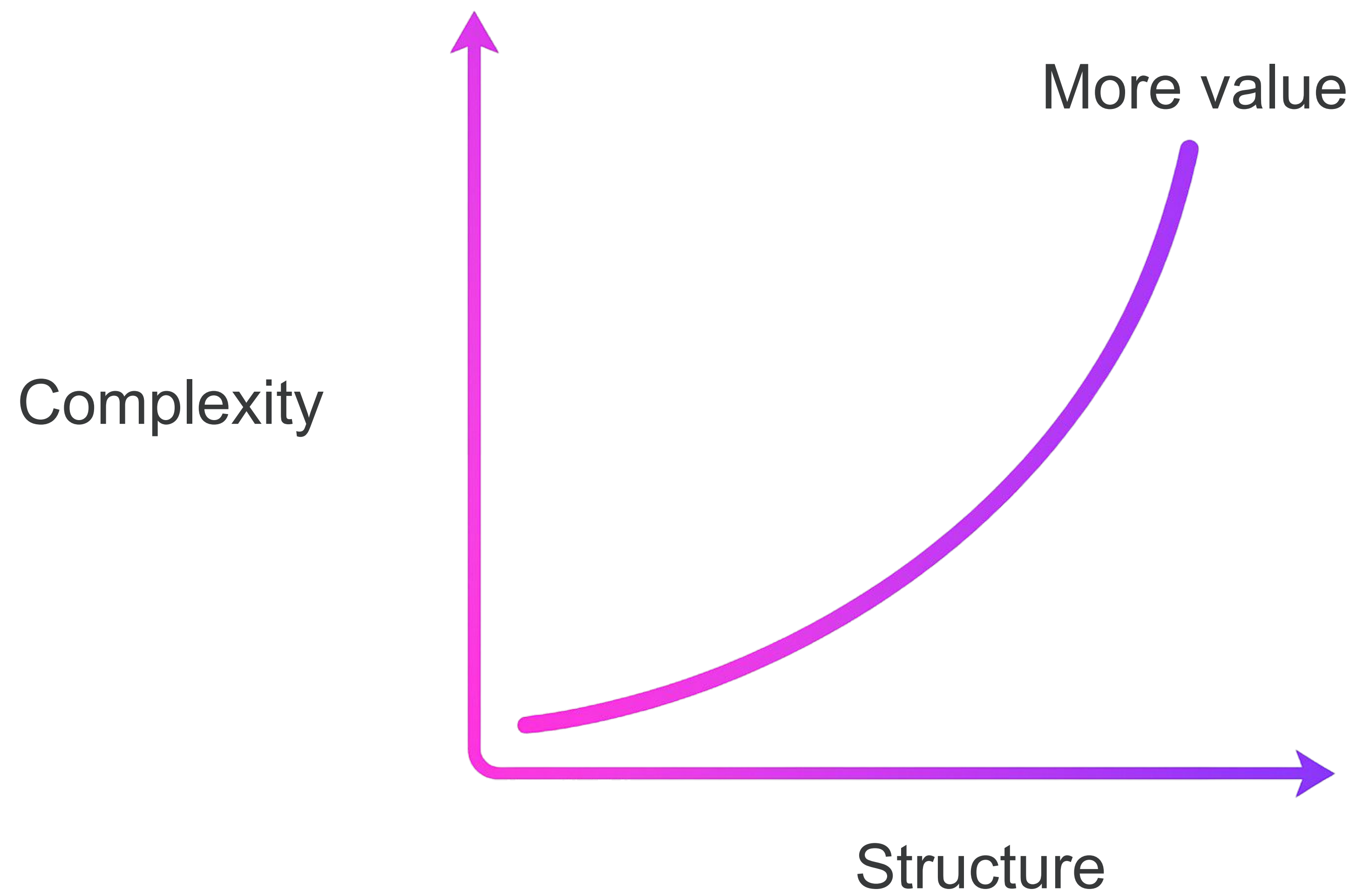


Vibes



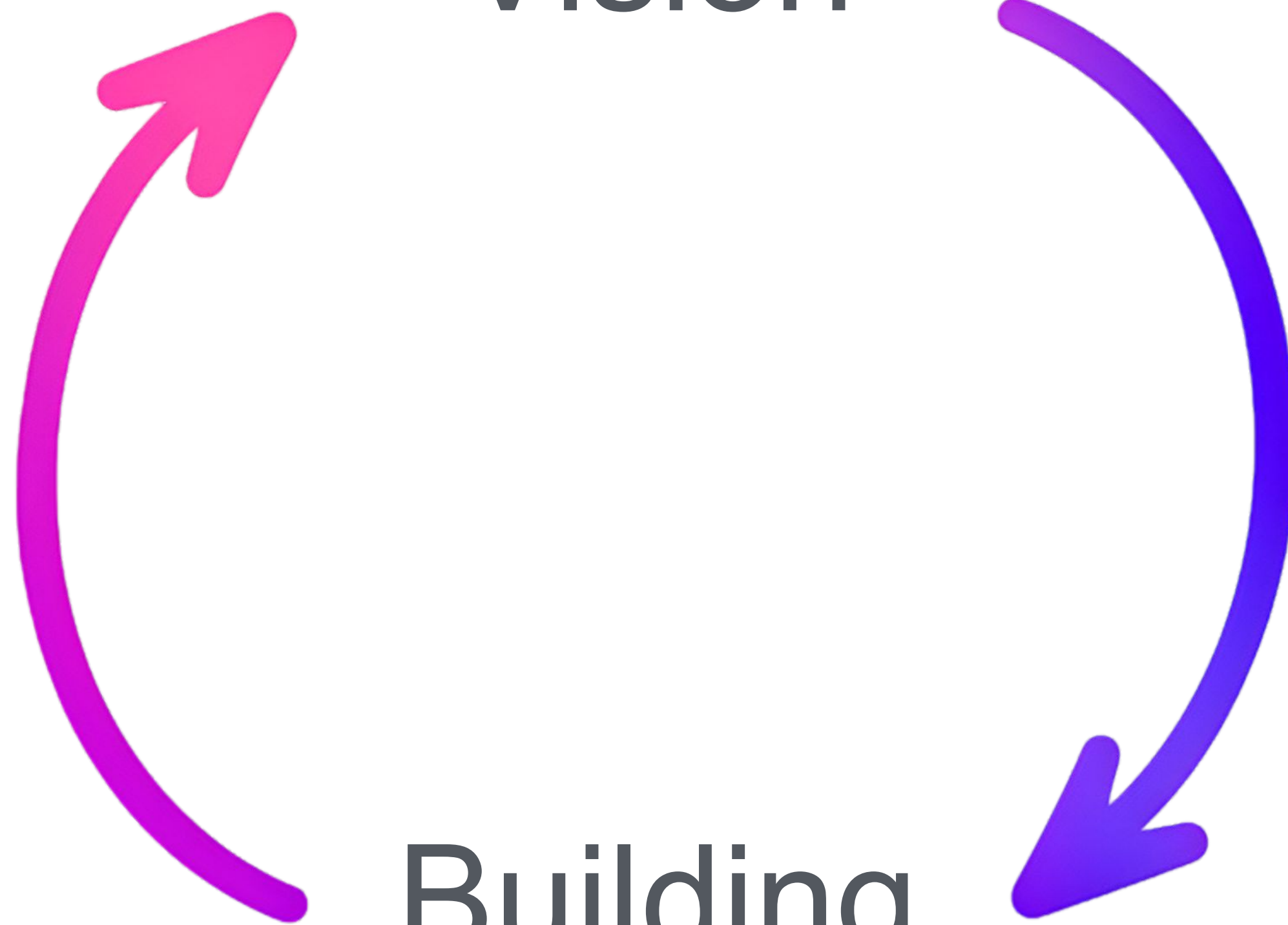
Less

More



Vision

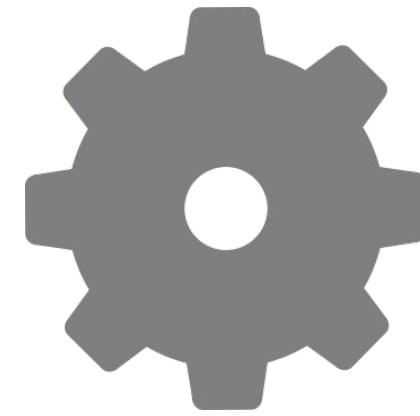
Building



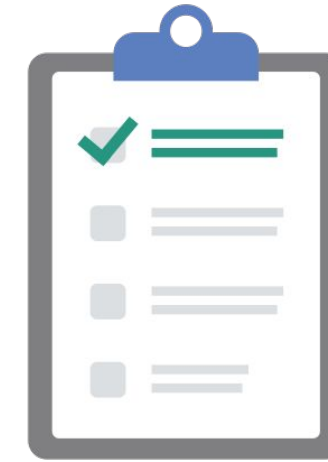
Specification



Blueprint



Roadmap





PRIORITY FLOW

The image features a light beige rounded rectangle containing a faint grey hourglass shape. A bright orange diagonal line with a slight glow runs from the bottom-left to the top-right. Three glowing cyan-blue dots are positioned along this line: one in the lower-left section, one in the narrow central neck of the hourglass, and one in the upper-right section. The text "PRIORITY FLOW" is centered in a bold, italicized black font, overlapping the central part of the orange line.

Fast & Smart



Context

Description

User

Job to be done

Research

APIs

Code libraries

Models

Ask me **one question at a time** so we can develop a thorough, step-by-step spec for an idea. Each question should **build on my previous answers**, and our end goal is to have a detailed specification I can hand off to a developer. Let's do this iteratively and dig into every relevant detail.

Remember, only one question at a time.

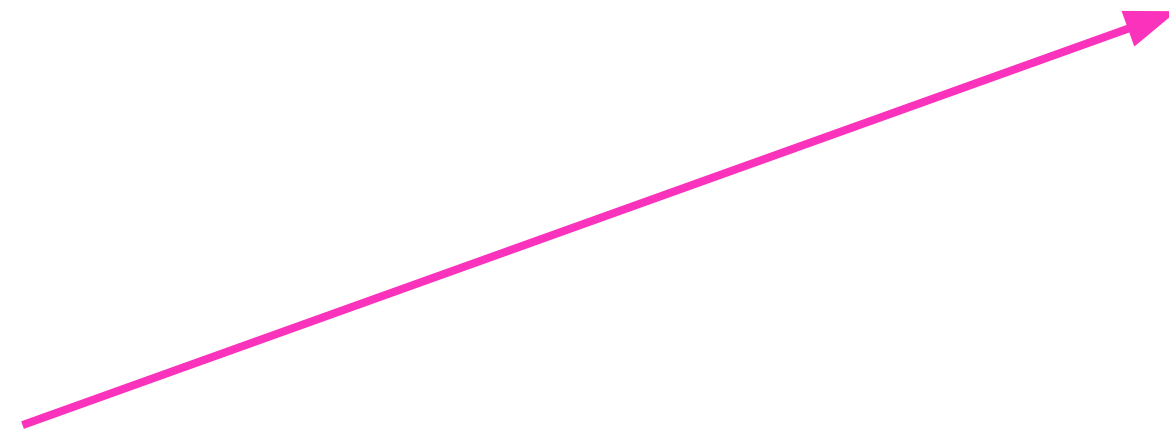
Here's the idea:

<IDEA>

Spec Demo



GPT o4-mini (high)



GPT o3

Now that we've wrapped up the brainstorming process, can you compile our findings into a comprehensive, **developer-ready specification**? Include all relevant requirements, architecture choices, data handling details, error handling strategies, and **a testing plan** so a developer can immediately begin implementation.

OpenAI o3



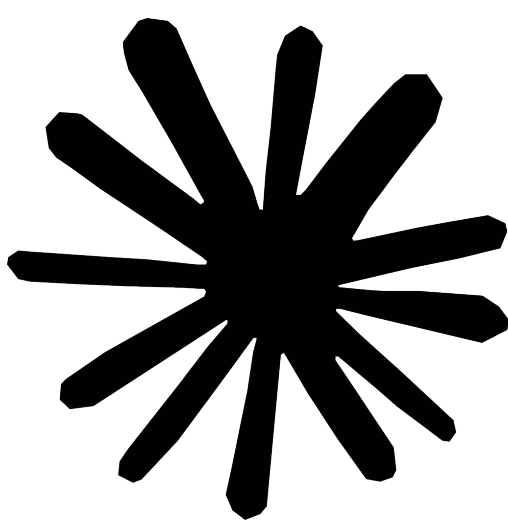
Gemini 2.5 Pro



Grok 4



Opus 4



Draft a detailed, step-by-step blueprint for building this project. Then, once you have a solid plan, break it down into small, iterative chunks that build on each other. Look at these chunks and then go another round to break it into small steps. Review the results and make sure that the steps are small enough to be implemented safely with strong testing, but big enough to move the project forward. Iterate until you feel that the steps are right sized for this project.

From here you should have the foundation to provide a series of prompts for a code-generation LLM that will implement each step in a test-driven manner. Prioritize best practices, incremental progress, and early testing, ensuring no big jumps in complexity at any stage. Make sure that each prompt builds on the previous prompts, and ends with wiring things together. There should be no hanging or orphaned code that isn't integrated into a previous step.

Make sure to separate each prompt section and use prompting best practices. Use markdown. Each prompt should be tagged as text using code tags. The goal is to output prompts, but context is important as well.

Make sure to write out all the prompts. Each prompt should stand alone and not reference other prompts

<SPEC>

Blueprint Demo



2.5 Pro (65k output)

Roadmap Demo

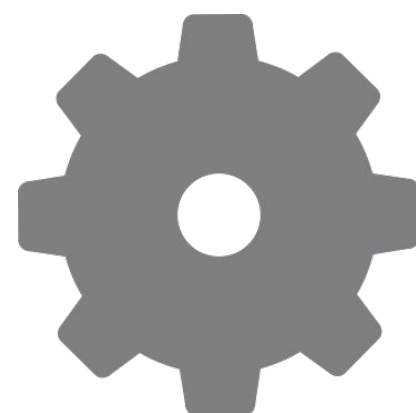


2.5 Pro (65k output)

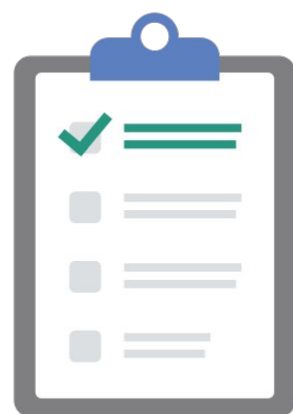
Specification



Blueprint

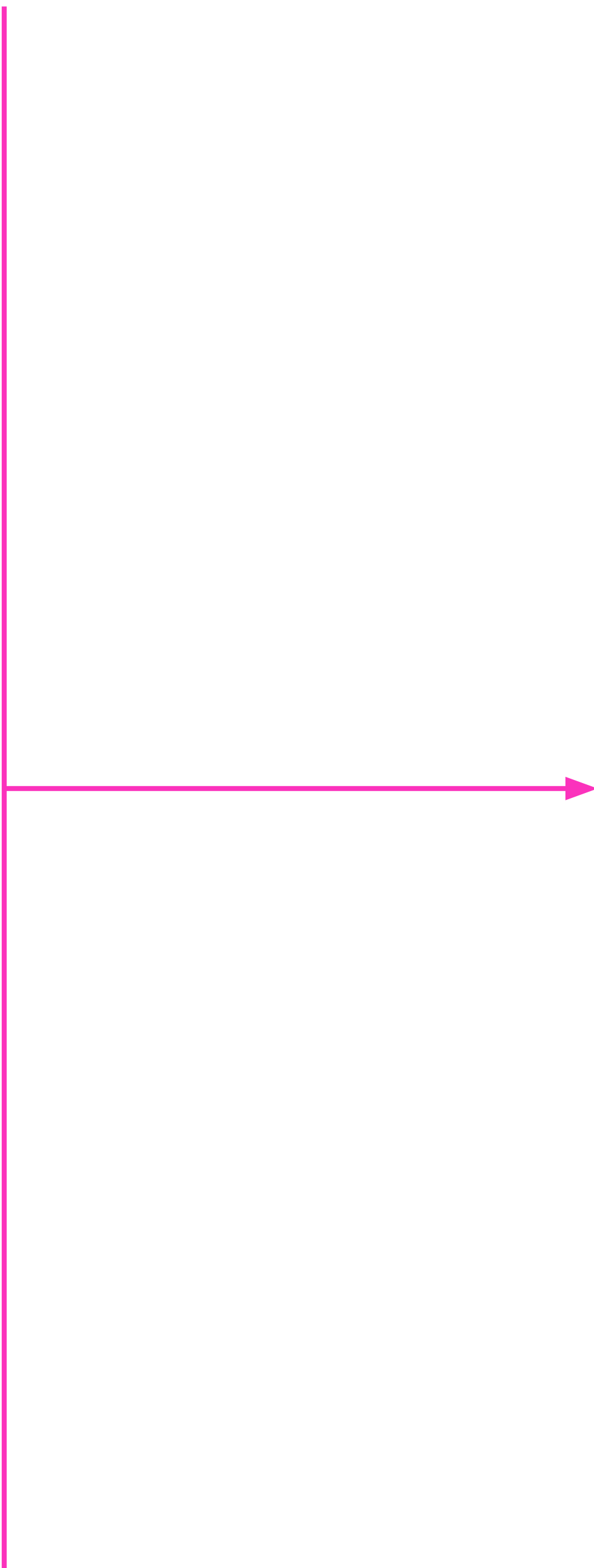
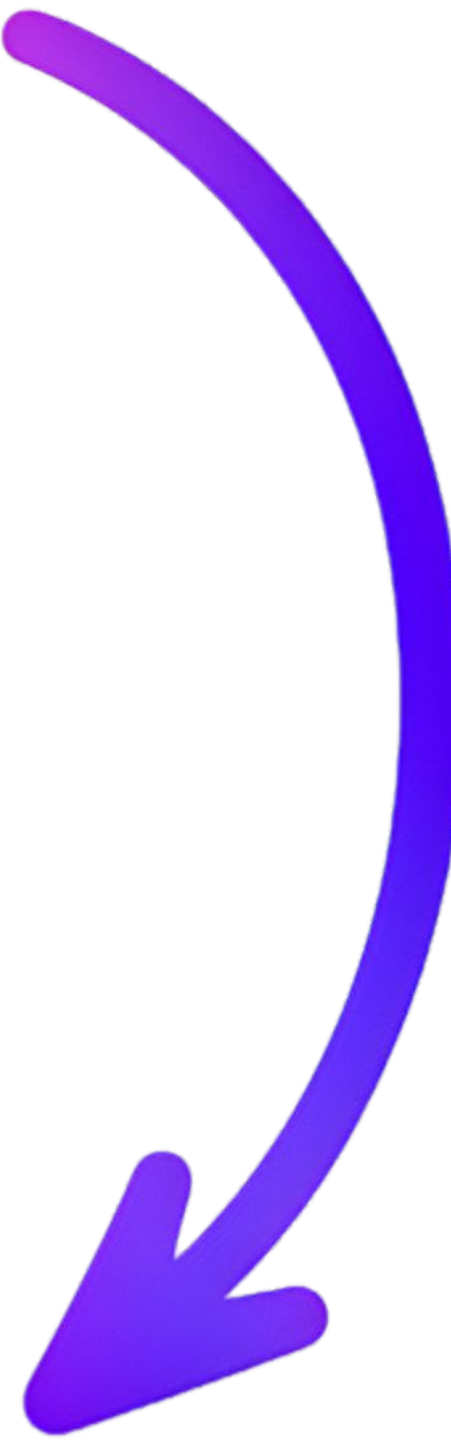


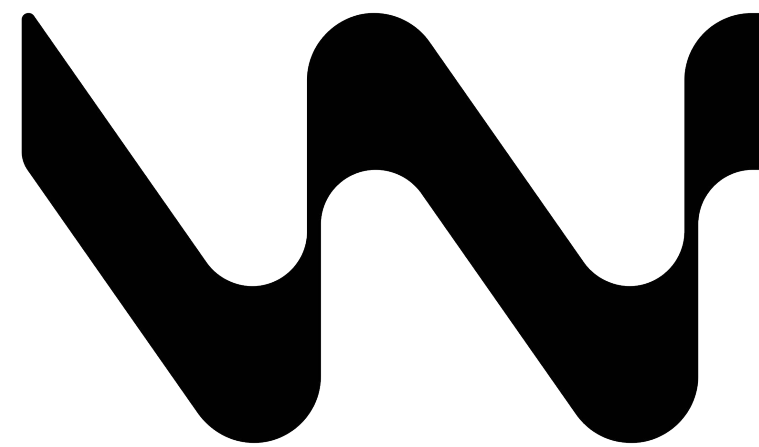
Roadmap



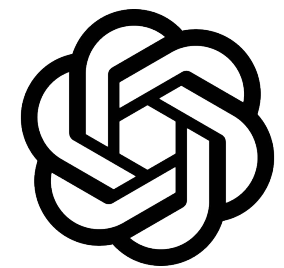
Vision

Building





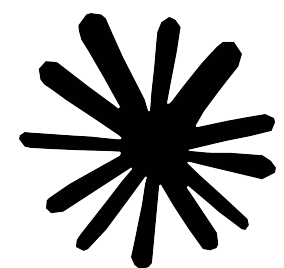
Plan



o3

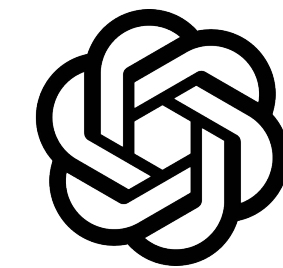
Gemini[◆]

2.5 Pro

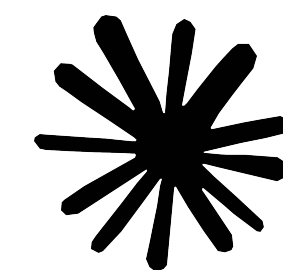


Opus/Sonnet 4

Code



GPT 4.1



Opus/Sonnet 4

Gemini[◆]

2.5 Pro