

# Why don't we keep our prompts?

Dongjae Lee

2025.06.13

## Abstract

After vibe coding, we lose our prompts, keeping only the code. However, prompts contain user instructions and serve as a kind of specification, containing very important information. Therefore, we should keep prompts as well. Before the era of vibe coding, specifications and code were documented separately, or code was written directly without specifications. However, in the vibe coding era, we need new languages and development environments that can simultaneously represent natural language specifications and code.

Vibe coding is a new way of coding that supervises coding agents using natural language. However, the way of interacting with agents is still chat-style. Chat is not a suitable method of interaction for vibe coding.

In chat, what is said is lost. Once it is lost, it is practically impossible to retrieve. But in vibe coding, what is said to the language model is a kind of code specification, which is very important. Therefore, we should preserve prompts while being able to track which parts were generated by each prompt.

Chat-style interaction makes users write prompts carelessly. Even if the first instruction is vague, it can be corrected in the next instruction. However, this repetitive correction work confuses the language model and makes the code bloated. Refactoring and optimizing such spaghetti code is not easy even for developers.

Instead, vibe coding should be changed to a method of writing permanent documents, and the language model should asynchronously rewrite the code based on the documents and modification requests. This approach removes the temporal dimension, preventing the language model from storing unnecessary change timelines in the context. Furthermore, it allows developers and agents to continuously focus on the core goal while writing code.