

Wayfind

The Debugging Copilot

LLMs are changing software development

- You can generate new projects in seconds
 - v0.dev
 - replit
- And “vibe code” all the way to a functioning project ([tweet](#))

Software development has two modes

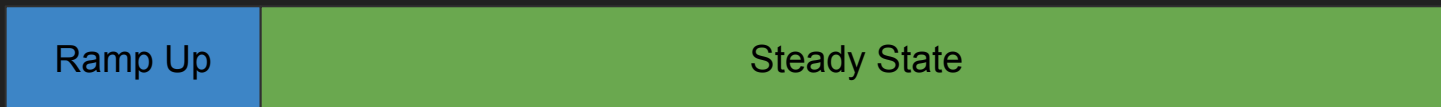
- Ramp Up
 - Writing / generating a new project
 - Understanding unfamiliar codebases
- Steady State
 - Fixing bugs
 - Adding new features

LLMs are changing how we spend our time

Before:



After:



LLMs help you go from 0 to 1 *much* faster, but they start to struggle as the codebase gets larger...

Programmer time goes increasingly towards whatever LLMs are worst at

LLMs introduce new problems

- Code turns over faster within companies
 - Net result: faster progress
 - Side effect: harder to keep an up-to-date mental model of the codebase
- LLMs sometimes get stuck 90% of the way there
 - Gets worse the further out of the training distribution you are
- LLMs are less useful on large codebases
 - Can't fit everything in the model context anymore
 - Code would be hard to run / debug without access to runtime data

LLMs *do* help with debugging, just less

- Have an LLM read files and summarize them to quickly understand the code
- Brainstorm hypothesis
- Write code to test theories more quickly

The language model has no access to the runtime context of the program (except logs)!

Can pdb / lldb/ gdb help LLMs transform debugging?

- Debuggers (pdb, lldb, gdb, etc.) allow you to run a program, evaluate variables at runtime, and step through functions
 - This is the missing runtime context!
- What if we exposed these through tool calls?
 - Programmer and LLM collaborate to plan a debugging strategy, and then execute it together
 - Tool call to check how a variable looks a few times at runtime > LLM guess about what will happen
- Additional benefit: this cuts down on the context that needs to be sent at the model

Takeaways

- The success of Cursor / Windsurf / Replit creates tailwinds for a new type of tool
- Steady-state programming can be transformed along the same lines



- The opportunity is massive! Most work is not greenfield.