

[Skip to main content](#)

r/LocalLLaMA



Search in r/LocalLLaMA

[Log In](#)

r/LocalLLaMA • 1 yr. ago

SomeOddCodeGuy



My personal guide for developing software with AI assistance

So, in the past I've mentioned that I use AI to assist in writing code for my personal projects, especially for things I use to automate stuff for myself, and I've gotten pretty mixed responses. Some folks say they do the same, others say AI can *never* write good code. I ran into a similar mindset among professionals in my field, and it made me realize that maybe folks are simply using AI differently than I am, and that's why our viewpoints are so different on it.

Before I begin, a little about where I'm coming from: I'm a development manager , and I've been in the industry for a while and even went to grad school for it. So when you read this, please keep in mind that this isn't coming from a non-dev, but rather someone who has a pretty solid bit of experience building and supporting large scale systems.

Also, if you read this and think "Why do all this when I can just ask it for code and it works?" This guide is for building large scale systems that are clean, maintainable, and as well written as you can personally muster. Yes, there's redundant work here and yes there's still a lot of work here. But, in my experience, it has not only sped up my personal development but also made it *really* fun for me and allows me to churn out features for hours on end without getting remotely fatigued.

My AI Development Rules

First: The rules I follow when coding with AI to get the most benefit

- Keep context low, because most AI I've found degrade in quality as the context gets larger. Make new conversations often, and rely on editing existing messages to reuse context. For example, if the AI produces a chunk of code and I have a question about it, I might follow up and ask my question. Then, if I see a second, unrelated, question- I might edit the first question that I asked and replace it with my second question, after which I regenerate the AI's response.

[Skip to main content](#)[Log In](#)

reviewing requirements to understand your requirements, make them more concrete, and critique it, refactoring as necessary". I've found that LLMs are too nice when I say I write it, and double down when I say that they wrote it.

- This isn't just about time savings, but mental energy savings. This means creating a workflow that saves the developer as much effort as possible by engaging the dev only at specific moments. There may be times reading this where you think "Why do this extra step BEFORE looking it over?" Because the AI can produce a response in 2 minutes or less, while a human can take 5-10 minutes to do the review, and that is energy spent. It will make you tired. I'd rather burn some AI time to get it right *before* the dev engages
- Do not rely on the AI entirely. Think of the AI as a junior developer- would you task a junior developer with a large scale application and not even review it? Of course not. With AI, you have a junior dev trapped in a little box, writing any code you want. Use that junior dev appropriately, and you'll get a lot of benefit.

Important Note: I always use 2 AI. Always. If you don't have a local AI, then Mistral has le chat for free, and you could use free chatgpt 3.5. If you have high end subscriptions, like Claude Opus and ChatGPT 4 Turbo, even better.

I prefer local AI models for various reasons, and [the quality of some like WizardLM-2 8x22b are on par with ChatGPT 4](#), but use what you have available and feel most comfortable with.

You CAN use just 1, but different models have different training, and may catch things.

Phase 1: Architecture

AI is terrible at architecture, so this is mostly you. You don't have to *deep dive* down to, say, the inner/helper method level, but at a minimum you want to document the following:

1. What is the project about? What are the requirements of the project, in a concise format that you can repeat to the AI over and over again whenever you pose a question to it?
2. What does "Done" look like? This is for your benefit, really. Scope creep is miserable, and you have no one to reign you in as the stakeholder. Trust me; my current project should have been done weeks ago but I won't... quit... adding... features...

[Skip to main content](#)[Log In](#)

-
4. At a high level, what kind of methods should each have? If you have a `LoggingService`, do you want a `"Log(message)"` method? If you have a `FileManagerService`, do you have a `"ReadFile(fileName)"` or `"ReadFile(filePath)"` or something else?

During this phase, you can present the answers to #1 and #2 to your AI and ask it for an architectural breakdown, but *don't just use its answer*. This is just to help you get over mental blocks, give you something to think about, etc. Write your own architecture. A big reason is because you, above all, need to know this project's structure inside and out. It will be harder for you to keep track of your project if you didn't write your own architecture.

Phase 2: The Coding

Below is the workflow I use. I understand that for many people this will feel like an unnecessary number of steps, but for me it has resulted in the highest quality that I've found so far, and has sped my development up massively... especially when working in a language I'm not intimately familiar with (like python. I'm a C# dev lol).

Yes, you can get code from AI far faster than what I'm about to say by simply asking for it and moving on, but the goal for me here is quality, developer understanding of the code, and adherence to the developer's style of coding. I want to write code that is clean, maintainable, scalable, and other developers at least won't want to set fire to if they look at it lol

Note: When making my first coding prompt of a conversation to the AI, I almost always include the answer to #1 from Architecture above- the breakdown of requirements for the full project. That context can sometimes help it better understand what you're trying to achieve.

- Step 1: Look over your architecture and pick a feature.
- Step 2: Present the requirements to the first AI (whichever you want to use first; doesn't matter), as well as the high level overview of the classes and primary methods that you want. I generally formulate a prompt similar to this: "Please write python code to read from a file and present the contents to the user. I'd like the code within a module called 'file_utilities', with a class 'FileManager' that has a method called 'read_file' that takes in a file name. I'd then like this called from a module called 'display_utilities', which has a method called 'display_contents_of_file'. This prints to the console the contents of that file. Please consider these requirements, give any

[Skip to main content](#)[Log In](#)

- Step 3: Copy the requirements and response. Start a new chat. Paste both, telling it that you asked another AI to write the solution, and that was the response. Ask it to please critique and refactor.
- Step 4: Copy the requirements and the new response. Go to AI #2 (if applicable) and ask it the same question as above.
- Step 5: Take the final response and code review it yourself. How does it look? Do you see any obvious flaws? Anything you want to change? Rename any helper methods as necessary. Consider whether any of it looks unnecessary, convoluted, redundant, or simply has a code smell.
- **Final Step:** Take the code, the requirements, and all of your feedback, and start over from step 2, doing the whole flow again if necessary.

While this may seem like it would be exceptionally time consuming, I can tell you that this workflow has worked amazingly for me in saving both time and energy. I'm usually dead tired at the end of a workday, and I simply don't have the mental energy to write code for another 4-5 hours straight. Because of this, I put off personal projects for YEARS. But by doing this, it allows me to get roughly similar quality to my own work when I'm fresh, while pawning off the labor portion of the dev to the AI.

I do the thinking, it does the efforting.

I would expect that steps 2, 3 and 4 will take around 5 minutes total. Step 5 will take 10-20 minutes depending on how much code is involved. Another loop will take another 15-25 minutes. So 1 feature will take around 20-60 minutes or so to produce. But the key here is how much mental energy you, as the developer, conserved while still maintaining tight control over the code.

Also note that this workflow won't work for EVERYTHING. Context limits can make it simply infeasible to engage the AI in some tasks. Say you've got 6 classes that are all working together on a function, and you realize there's an odd bug that you can't figure out where it is in that workflow. More than likely, you won't find an AI capable of handling that amount of context without degraded quality. In those cases, you're on your own.

Anyhow, I know this is lengthy, but I wanted to toss this out there. This workflow has worked amazingly for me, and I intend to continue refining it as time goes.

↑ 375 · ○ 54

[Skip to main content](#)[Log In](#)

actually think this is a great way to do your own pair programming . Plus you get to QA your code as you go.

You also get the benefit of an extra AI looking at the code and verifying what the first model gave you.

Also, it looks like a lot to do, but it really isn't. It's pretty fast once you get the hang of it a couple of times.

⊖ ↑ 32



SomeOddCodeGuy OP • 1y ago

Plus you get to QA your code as you go.

Thank you for mentioning this; I dropped the ball on not bringing that up.

You are the QA here, so test often. After implementing a feature- RUN IT. Make sure that it works lol. That's so important to development in general. Unit tests alone won't tell you if something works well, so be sure to put on your QA hat here.

↑ 25

7 more replies ▾



freedom2adventure • 1y ago

And watch out for strange imports.

↑ 16

1 more reply ▾



deoxykev • 1y ago

Check out <https://aider.chat>, I think it fits really well with your mindset you outlined here. This will just cut down on the mechanical aspects of copy-pasting code, allowing you to iterate faster.

⊖ ↑ 13



SomeOddCodeGuy OP • 1y ago

[Skip to main content](#)[Log In](#)

dawned on me after doing this workflow like 500 times that it was kind of ridiculous that I, as a developer, was sitting there repetitively doing something that code should handle lol

⊖ ↑ 11



RobotRobotWhatDoUSee • 6mo ago

Many months later, but did you end up using Aider? If not Aider, what are you using now? (I see you have a project called Wilmer that I haven't checked out yet, so maybe this question will get answered as I look onto that!)

⊖ ↑ 2



SomeOddCodeGuy [OP](#) • 6mo ago

So I ended up not using Aider at the time, but I've been re-checking out the agentic workflows recently to see where they are.

You're pretty much on the money with Wilmer- my personal codebases tend to be a little bigger, and my usecases a little complex, so I really started to find myself getting frustrated with both agentic workflows and code completion stuff like CoPilot or Continue.dev. They're fantastic, all of them are, but I wanted more granular control over the inputs/outputs and found I could do a lot of things faster myself that way.

So I ended up setting up several Wilmer workflows, some of which automate what I wrote out here, and have gone back to almost exclusively using chat window based AI assistants when coding, as I can get output a bit faster that way.

But since it had been a while since I last tried them, I was actually watching some youtube vids on them yesterday to see where they are and maybe give them a shot.

↑ 2

[Continue this thread >](#)

1 more reply ▾



shroddy • 1y ago

[Skip to main content](#)[Log In](#)

And does it make a difference compared to just writing "analyze the following code" without mentioning where it comes from?

⊖ ↑ 12



SomeOddCodeGuy [OP](#) • 1y ago

I've had better luck with asking it to review the code over analyze, and I've always suspected that was due to reddit and stack overflow.

In terms of the training data on telling it another AI did it... I'm not sure. I feel like it acts differently if I say I did something, as if it's trying not to offend me, and if I say it wrote the code then it seems to start from a position of thinking the answer is more correct than it is.

By saying another AI wrote it, I'm basically saying "This wasn't written by me, by you, or by anyone with feelings, so no one will be made sad if you rip it to pieces", and in general I've had better results getting it to be critical that way.

With that said, that's purely anecdotal and I'm sure a proper study could prove me wrong lol

↑ 12

1 more reply ▾



lolzinventor • 1y ago

Something that helped me with AI generated code was asking it to produce unit tests for the functions generated, using a unit test framework. This makes the AI 'think' more about uses cases and parameters. It also makes testing each function/class easier as you can paste back the failing test results and it can readily identify them from its context. The steps from here to a semi automated agent based work flow are getting clearer every day. Another coding project for an AI perhaps.....

↑ 9



MoffKalast • 1y ago

[Skip to main content](#)[Log In](#)

...entaining by actually helping to maintain and speed up how to process. Easy to implement involves established algorithms that they've seen 100 billion times in the training data and can tailor them perfectly for your use case, while you know what they're a good fit for but have forgotten how to implement them by hand since college/interview and would need a refresher.

They may be meh at architecture, but can still help in forming it by listing all the options that you might be forgetting about or haven't yet considered. The old GPT-4 saved me so much time on a few occasions by just listing some libraries that fit my use case perfectly which I'd never heard about.

I always use 2 AI. Always.

Honestly, I just throw the problem into *all* of the flagship models that are available for free from every company if it's not immediately solved by the first one you tried.



6



tylerjdunn • 1y ago

Thanks for sharing! Super helpful



5



ai_did_my_homework • 10mo ago

Really good, you could sell this as a course lol



4



cryptoguy255 • 1y ago

What models do you use and have the most success with? I only find it useful for creating starting points for a feature. Describe the feature and iterate with small task on top until all falls apart. That is when I fully take over. With larger code calling other code it becomes mostly useless beside fill in the middle to generate some boilerplate code.



3



SomeOddCodeGuy [OP](#) • 1y ago

[Skip to main content](#)[Log In](#)

work down into really small chunks. You'll be applying the exact same mentality here that you would to writing story tickets that can be iterated in a sprint- it needs to be bite sized, and it's the architect's job to keep track of where it falls in the overall flow.

By doing this, I rarely hit a point of scale that I can't keep doing this. I have had to refactor my code once or twice to keep doing that, but I realized this is a good thing, because a really neat feature of this workflow is enforcing modularity. If your code isn't modular, you can't do this after a while. So when I start accidentally creating too many dependencies, my methods grow too large or my methods aren't named in a self-descriptive enough way, then I end up hitting a point where I can't figure out how to cram everything I need into the AI. That's my code smell that I done goofed for other reasons, and I stop and refactor lol.

↑ 15

2 more replies ▾



kex • 1y ago

I find that functional programming practices (especially pure functions) work well with these models since FP inherently keeps context limited

↑ 4



nospoon99 • 1y ago

That's a great guide with some interesting ideas. I like "tell the AI another AI wrote it", so simple! Can't wait to try it out.

↑ 2



SlapAndFinger • 1y ago

[Log In](#)

Also, lots of small, strongly typed functions is the way to go. Get the LLM to write the function and the test in one go, which is pretty doable when they're small.

Finally, I've had decent success with iteratively asking the LLM to update a given piece of code with new features before finally doing a rewrite.



2

**FrankYuan1978** • 1y ago

Thanks, it is help for me.



2

**satoshibitcoin** • 1y ago

Would be nice to have this workflow setup in an IDE



2

**SomeOddCodeGuy** [OP](#) • 1y ago

I've got a project I've been working on the past few months that I'm desperately trying to get patched up enough with duct tape and faith to shove out the door, which will let you do just that. It should work with several existing copilot alternatives like continue for vscode/pycharm. Hopefully in a couple more weeks I'll have something to test out ready to roll.



4

[2 more replies](#) ▾[View more comments](#)