99% of Developers Haven’t Seen Claude Code Sub Agents (It Changes Everything)

The gap between solo devs and teams just shrunk

For the last month, I’ve been loving using Claude Code. Building my open source project Mentis and shipping features at work. Claude Code finally feels like a true developers AI tool, mimicking the vibes I felt when I first started developing.

But recently I have been noticing something very disruptive to my pattern of thinking. Well, Claude Code may just have a massive solution to the problem.

Not a Medium member? Keep reading for free by clicking here.

The Disruptive Problem without Sub Agents

The thing about development that no one mentions, we are constantly switching between completely different mindsets without even realising it.

When I’m building Mentis, I have really noticed my brain goes through these different modes:

- **Code Architecture Mode:** *Balancing DRY principles without creating complexity problems.*
- **Project Architecture Mode:** *Figuring out if anything is getting out of hand and making sure I won’t need an insane refactor.*
- **Project Planning Mode:** *Translating features into technical requirements, making sure the next feature I work on isn’t completely thrown away by an upcoming feature.*
- **Testing Architecture Mode:** *Looking for testing gaps, looking and thinking about edge cases, figuring out if my tests have gaps.*
- **Documentation Mode:** *Focusing on current documentation to see if needs to be updated and looking for gaps that need to be filled in.*

The list is actually much longer, we switch between these mindsets frequently everyday.

Claude Code is an amazing advancement compared to the rest, it got the basic flow very right. Plan, code, test. Other AI tools do this too but Claude Code really feels like it has

nailed this.

The Big Frustration I Always Have

Last week, I was working on a single test for Mentis. I was in my "Testing Mindset", focused and just wanting to cover one edge case.

I asked Claude Code to help me write this test.

Instead, it decided to refactor a bunch of other tests, modify code, then finally it made a test suite instead of a single test.

My focus was gone, for a moment I completely forgot about my test case, now I was reading random changes. *Later on I added memory to help with this but I don't always want it to be applied.*

Suddenly because of AI, my mindset had shifted context and was now in the "Project/Test Architecture Mode" without me intentionally being in that mindset.

This honestly happens a lot:

- *"Suggest changes, don't code them"*, only for Claude to code anyway.
- *"Fix this test"*, only for Claude to fix the logic not the test case.
- *"Fix this bug"*, only for Claude to implement a new feature, instead of finding the root cause.
- *"How can I refactor this"*, only for Claude to start implementing a crazy approach.

It feels like working with a Junior Developer, who changes context way too often which you have to remind to stay focused.

Claude Code Dropped Sub Agents

Claude Code quietly released something called Sub Agents. I would have completely missed it, if it wasn't for reading through their docs.

The concept follows mixture of experts, instead of one AI trying to be everything, you create specialized agents for tasks.

The concept made me super excited instantly, instead of these confusing mindsets we can specialise agents into these "mindsets".

> *Custom sub agents in Claude Code are specialized AI assistants that can be invoked to handle specific types of tasks. They enable more efficient problem-solving by providing task-specific configurations with customized system prompts, tools and a separate context window. — Anthropic on Sub Agents.*

Each sub agent gets a context window.

No more context unintentional switching mindset, each specialist maintains their own focus.

The setup is very simple:

```
/agents
```

Make sure you have updated Claude with `claude update`.

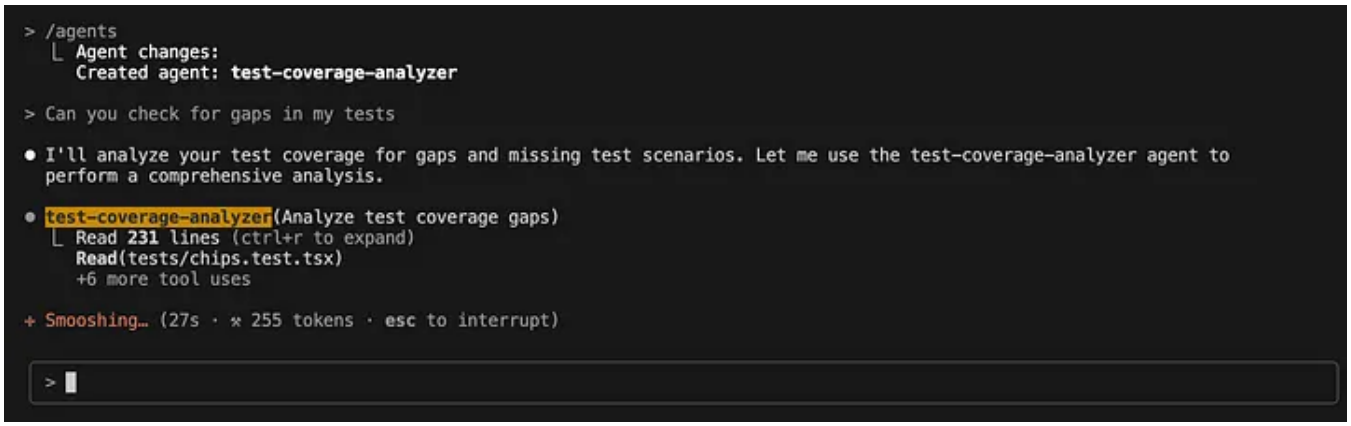
Claude Agents terminal selection

How I Make My Agents

I currently like adding the Agents to the current project, as I like to have them version control. I use the “Generate with Claude” as it’s recommended.

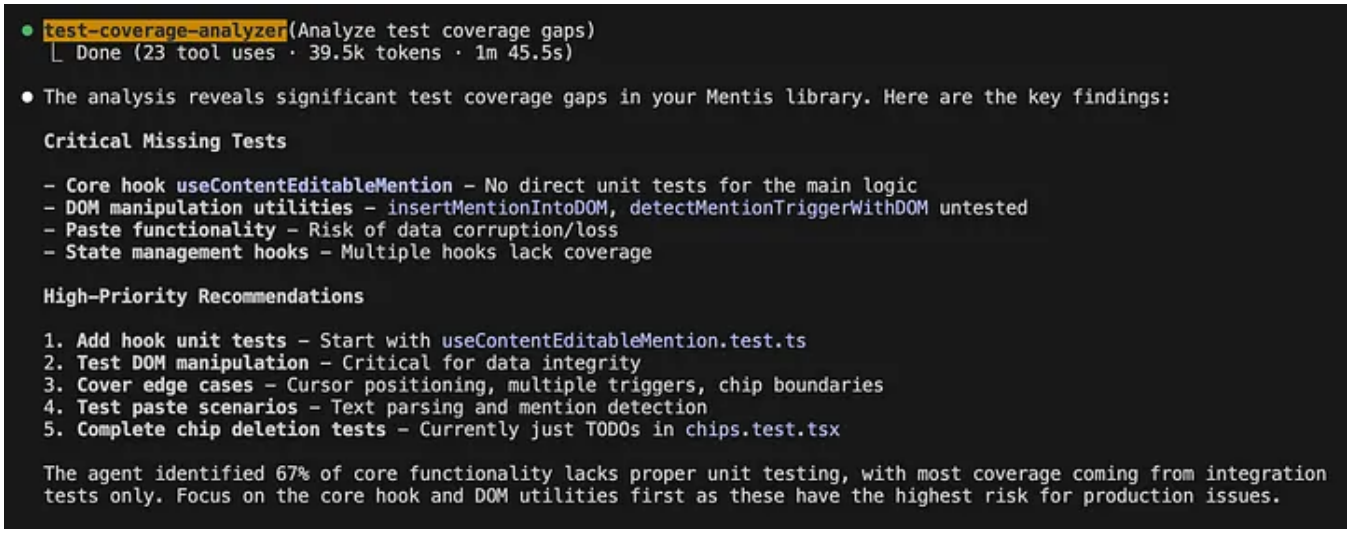
Select Tools only when you need them

Here is an example of my architect agent tool access, I don't want to give them access to editing as I only want advice not implementation, instead I'll hand that to another agent.



test-coverage-analyzer agent prompt

> *Expert focused purely on test coverage, gaps, and quality. Use proactively when testing anything.* — The input I provided to made the agent.



test-coverage-analyzer agent response

This Has Completely Changed How I Think About AI

That response might not seem very major but for me it solves an insanely large frustration I have.

This response is perfect and exactly what I wanted, now I have control of how Claude Code actually works, so it can better mimic my mindset and workflow.

What is really happening:

- **Context Isolation:** *Agents aren't mixed up between context.*

- **Automatic Agent Delegation:** *Claude Code will route tasks.*
- **Explicit Invocation:** *Use specific agents by calling out to them.*
- **Tool Restriction:** *Only give access to the tools you need.*

This workflow has completely changed my daily development.

The Mindset Revolution

This change isn't about pure coding speed, it's about removing one of our biggest problems in development (context switching).

Before sub agents there were hidden costs:

- *Will it do what I asked?*
- *Will it go off and confuse me?*
- *Will I spend more time fixing something, than just building it myself?*

Now I can define what Claude Code actually does and how it does it.

It's like having a team, where you get to choose the team.

What You Should Do This Weekend

- **If you have 30 minutes:** Update and run `/agents` create and try agents out.
- **If you have 2 hours:** Build a 5 agent team for a project.
- **If you have a weekend:** Use global agents vs project agents.

Use Claude's recommendations when it comes to generating and using agents.

IMO, we are at the beginning of another massive shift in software development.

I wonder what development mindsets you find yourself switching between the most? If you have tried sub agents, do you find they solve or improve this?

I'm not affiliated with Claude or Anthropic. All opinions shared are from my own experiences as a software engineer.