

★ Member-only story

Claude Code and Git Worktrees

Scaling Agentic Coding with Parallel Claude Code

3 min read · Jul 15, 2025



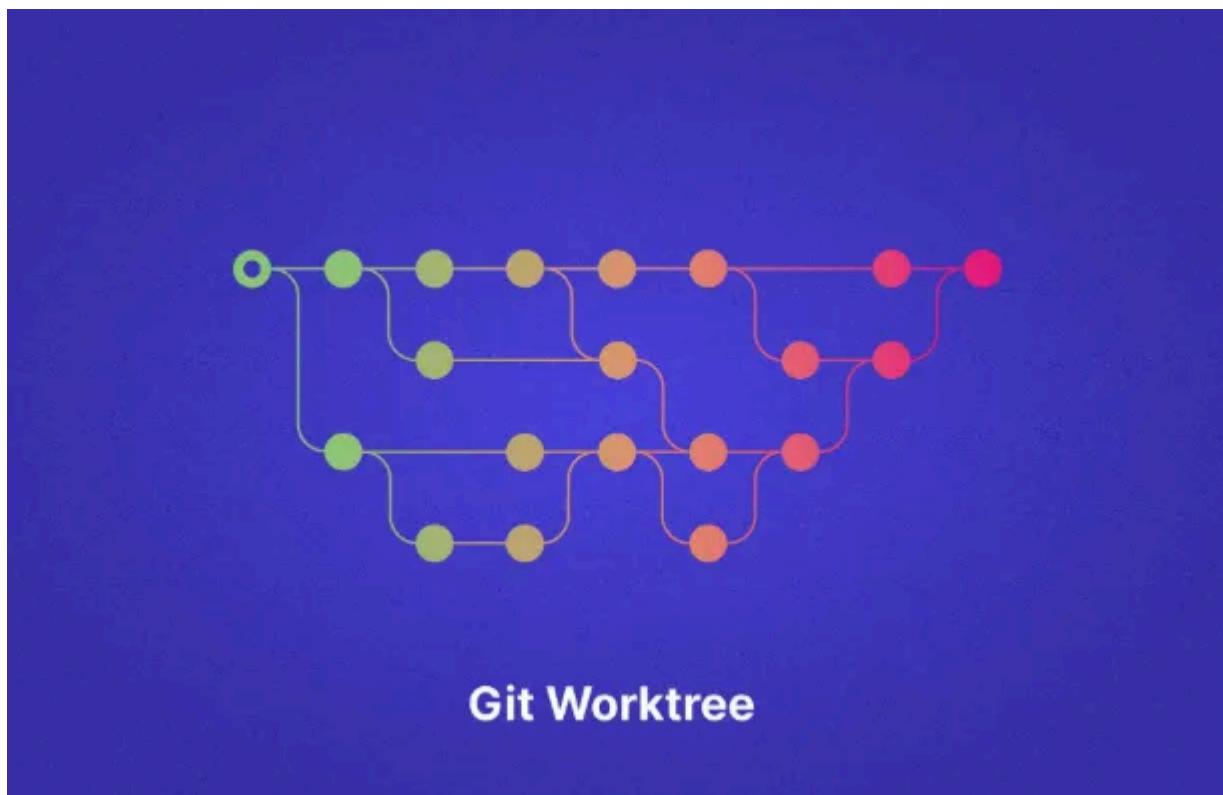
Kenji

Follow

Listen

Share

More



<https://devdynamics.ai/blog/understanding-git-worktree-to-fast-track-software-development-process/>

When I first experimented with Claude Code alongside my favorite IDE, I noticed something. While a single agent could tackle complex tasks, I often wished for multiple “opinions” on a design or implementation.

What if I could generate different takes on the exact specification, side by side? That is actually documented in [Anthropic's Claude Code best practices](#) — it's about using Git worktrees — and it has reshaped how I approach non-deterministic AI workflows.

• • •

Why Parallel Agents Matter

Claude Code breaks down engineering tasks into clear plans. Yet, because large language models produce varied outputs, running the same prompt twice can yield distinct results. Rather than fight that variability, we can embrace it — launching multiple agents in parallel and selecting the version that works best.

“I’d rather review three implementations than roll the dice on a single outcome.”

However, a simple `git checkout` approach won’t work: all agents would compete for the same branch, causing merge conflicts and lost progress. That’s where Git worktrees come in.

Git Worktrees Explained

Worktrees provide physical isolation for each agent:

- **Setup:** From your main directory create a sibling folder called `trees`.

```
git worktree add -b task-1 ../trees/task-1
```

Repeat for as many agents as you need (e.g., `task-2`, `task-3`).

- **Advantages:** Each agent works independently — you can update ports, install dependencies, and run servers (e.g., on ports 3001–3003) without interfering with one another. Once they finish, merge via standard Git.

Automate worktree creation with this `create_worktree.sh` script:

```
#!/bin/bash
BRANCH=$1
mkdir -p ../trees
```

```
git worktree add -b $BRANCH ../trees/$BRANCH  
cd ../trees/$BRANCH  
echo "Worktree created for $BRANCH"
```

A Claude command can streamline this further: it creates worktrees, copies `.env` files, updates configs, and preps each environment. Then fires off the detailed plan (your task specifications) to each agent on Claude 4 simultaneously.

• • •

Parallel Execution in Action

Let's see how this works on a generic front-end project — **DemoApp**:

Plan: Define the task clearly. For DemoApp, the goal might be: “Convert the main dashboard to a compact, information-dense layout with collapsible panels and updated typography.”

Agents: Spin up three parallel agents in separate worktrees:

```
for i in {1..3}; do  
  git worktree add -b task-$i ../trees/task-$i  
done
```

Runtime: Each agent consumes tokens and completes. Because each worktree is isolated, they can install dependencies, run local servers (e.g., on ports 3001, 3002, 3003), and apply changes without conflict.

Testing: Launch all three instances at once:

```
./start-tree-clients.sh 3001 3002 3003
```

In Claude Code, define a custom slash command in `.claude/commands/exe-parallel.md`:

```
Execute the plan in parallel agents: for i in {1..3}; do cd ..../trees/ui-rewrite
```

Run with /exe-parallel.

This leverages LLMs' probabilistic nature: Different agents produce unique perspectives, ideal for ambiguous tasks like UI design.

• • •

Merging and Strategic Use

Post-execution, review summaries. Commit changes per worktree, then merge the best (e.g., the modern version) into main: git merge task-rewrite-2.

Resolve conflicts as usual. Use this bash snippet to merge safely:

```
git checkout main
git merge task-rewrite-2 --no-ff -m "Merge best UI revamp from agent 2"
git push origin main
```

Use this for:

- Ambiguous/hard tasks.
- When failures are likely – multiple agents increase success odds.
- With clear plans, iterative prompting suits single branches.

This isn't cheap (high token costs), but it maximizes compute: Generate n futures, pick the best, ship faster. As agentic tools like Claude Code evolve, parallel workflows position you to orchestrate, not just code.

In 2025's AI landscape, scaling agents isn't optional — it's essential.

Artificial Intelligence

Technology

Innovation

Productivity

Programming



Follow

Written by Kenji

421 followers · 372 following

I explore how AI & tech shape our lives, from innovation to personal growth. Writing helps me learn, and I'm curious about where tech is taking us.

Responses (1)



Bgerby

What are your thoughts?



Nicolae Rusan

Aug 17

...

I like this approach of parallel implementations, how are you trying to get the best variety of possible solution spaces?

We're exploring things like this at Magnet (www.magnet.run) - would love your feedback if you give it a spin!

1 [Reply](#)

More from Kenji



In AI Unscripted by Kenji

Getting Started with Claude Code Subagents

How they work, and why they're worth using in your coding workflow.



Jul 29



92



...



Kenji

My Experience with OpenWebUI API Access: Overcoming Common Integration Challenges

Tips and Solutions for Solving Common OpenWebUI API Access Issues

Sep 17, 2024

70

1



...

In AI Unscripted by Kenji

A Beginner's Guide: Exploring Subagents in Claude Code

Claude Code tool can spawn *subagents*—lightweight instances of the model that execute tasks in parallel. When you live in Bangkok and...

♦ Jul 24 99



...

In AI Unscripted by Kenji

How to Build Effective AI Agents: Lessons from Anthropic

A Practical Guide to Creating Smart, Reliable AI Systems: From Workflows to Dynamic Problem Solvers

◆ Jan 3 🙌 195 💬 3



[See all from Kenji](#)

Recommended from Medium

 In AI Software Engineer by Joe Njenga

Anthropic Just Solved AI Agent Bloat—150K Tokens Down to 2K (Code Execution With MCP)

Anthropic just released smartest way to build scalable AI agents, cutting token use by 98%, shift from tool calling to MCP code execution

◆ 4d ago 🙌 417 💬 33



 The Atomic Architect

You Haven't Seen AI Until You Try Claude Sonnet 4.5's New Feature—It Redefines Insane

I built a working expense tracker in eight minutes while my coffee was still hot, and it remembered every receipt when I closed my laptop...

 Oct 26  464  20

...

 Bhavyansh

My Favorite 8 CLI Tools for Everyday Development (2025 Edition)

Replace your old Unix tools with modern alternatives that actually respect your time

 5d ago  53  5



...

In Predict by Tasmia Sharmin

The Man Who Invented AI Just Admitted What Tech CEOs Won't Say!

Geoffrey Hinton: They're spending \$420 billion on AI. It only pays off if they fire you

 Nov 2  1.5K  133



...

In Towards AI by Teja Kusireddy

We Spent \$47,000 Running AI Agents in Production. Here's What Nobody Tells You About A2A and MCP.

Multi-agent systems are the future. Agent-to-Agent (A2A) communication and Anthropic's Model Context Protocol (MCP) are revolutionary. But...

star Oct 16 3.4K 116



...

In Coding Beauty by Tari Ibaba

This insane new coding model is 13 times faster than Claude Sonnet 4.5



Just wow.

star Nov 2 233 3



...

See more recommendations