

← Back to Series



**Claude Code**  
Features Guide

📖 Part 1 of the Claude Code Series

## How to Use Claude Code Agent Teams: The Complete Guide

Learn how Claude Code Agent Teams coordinate parallel AI agents to tackle complex projects. Real case study: 7 deliverables in 15 minutes across a plumbing franchise.

📅 9 February 2026 ⌚ 11 min read

What if you could clone your best analyst four times, point each clone at a different problem, and have all four deliver their findings simultaneously?

That is essentially what Claude Code Agent Teams does. And after running it on a real client project, we can confirm: it changes the way technical work gets done.

Agent Teams is the flagship feature in Anthropic's latest Claude Code update, and it is the one we have been most eager to test. We ran it against a genuine multi-site SEO project for a plumbing franchise across Sydney and Melbourne - and the results were remarkable enough that we rewrote our internal workflows the same week.

Here is everything we learned.

### Agent Teams Guide

#### Part 1: Multi-Agent Editing

Run parallel AI agents on real projects

## What Are Claude Code Agent Teams?

Agent Teams lets you spin up multiple AI agents that work on different parts of a project simultaneously, coordinated by a single team lead agent.

Think of it like a project manager running a kick-off meeting. The team lead reads the brief, breaks it into workstreams, assigns each workstream to a specialist teammate, then monitors progress and synthesises the results.

## How the Architecture Works

The system has three layers:

1. **Team Lead** - The orchestrator. It reads your instructions, decides what sub-tasks are needed, delegates them to teammates, and compiles the final output.
2. **Teammates** - Independent agents that each receive a focused brief. They can read files, run commands, search the web, and produce deliverables - all within their own context window.
3. **Communication Layer** - Teammates report back to the team lead with their findings. The lead can ask follow-up questions or request revisions before accepting their work.

Each teammate operates in its own thread with its own context. This means they do not pollute each other's thinking or run into context window limits from unrelated information.

## Agent Architecture: Parallel Dispatch



## Real Case Study: Auditing a Plumbing Franchise Network

We did not want to test this on a toy example. We wanted to know if Agent Teams could handle the kind of messy, multi-dimensional work that actually fills our client engagements.

### The Brief

Our client, Mr Splash Plumbing, operates a franchise network with over 20 microsites across Sydney and Melbourne suburbs. Each site targets local keywords like "plumber Parramatta" or "emergency plumber St Kilda."

The problem: the network had grown organically, and nobody had done a comprehensive audit across all sites simultaneously. Some sites were performing well. Others were practically invisible. We needed to understand the full picture - fast.

### The Team We Built

We set up a team of four specialised agents:

Agent	Role	Workstream
GSC Analyst	Google Search Console data analysis	Indexation health, crawl issues, performance trends
Location Optimiser	Suburb page content analysis	Content duplication, local relevance, optimisation gaps
SEO Researcher	Backlink and keyword analysis	Link profile health, ranking opportunities, spam detection
Dev Planner	Technical audit and task mapping	WordPress configuration, site speed, development roadmap

Each agent received a focused brief with access to the project's data files, API credentials, and relevant frameworks.

### What Happened Next

All four agents started working at the same time. While the GSC Analyst was pulling indexation data, the Location Optimiser was comparing content across suburb pages, the SEO Researcher was analysing backlink profiles, and the Dev Planner was crawling site configurations.

Fifteen minutes later, we had seven deliverables on the table.

### The Findings

#### GSC Analyst discovered a critical indexation crisis:

- 300 pages across 7 sites were blocked from Google by a WordPress "Discourage search engines" checkbox that someone had ticked and forgotten about
- The overall index rate across the network was just 56.3%
- Dozens of pages that should have been ranking were completely invisible to search engines

### **Location Optimiser uncovered a massive duplication problem:**

- 97% duplicate content across 43 suburb pages
- The same generic plumbing copy had been pasted across nearly every location page, with only the suburb name swapped out
- An 8-phase remediation plan was produced, prioritised by traffic potential and competitive opportunity

### **SEO Researcher mapped the full link profile:**

- 63,100 total backlinks analysed across the network
- 4 spam domains identified that needed to be disavowed
- 11 keywords sitting in position 2 or 3 - just one spot away from page one, representing quick-win opportunities

### **Dev Planner created a comprehensive roadmap:**

- 12 pending development tasks mapped, totalling 48 to 70 hours of work
- Independently flagged the same WordPress noindex issue the GSC Analyst found
- Prioritised tasks by impact and dependency order

### **The Validation Effect**

Here is what made us sit up: two agents working on completely different workstreams, with no communication between them, independently identified the same critical issue - the WordPress "Discourage search engines" checkbox.

The GSC Analyst found it through indexation data patterns. The Dev Planner found it through WordPress configuration audits. Neither knew what the other was working on.

This kind of independent corroboration is something you normally only get from expensive, time-consuming parallel review processes. Agent Teams delivered it as a natural byproduct of parallel execution.



#### **Each Agent Gets a Clear Scope**

The key to multi-agent success is separation. Each agent owns its own files, its own context, and its own deliverable. No overlap means no conflicts.

## **How to Set Up Your First Agent Team**



Setting up an Agent Team in Claude Code is straightforward. Here is the process we use.

### Step 1: Define Your Workstreams

Before involving any AI, write down the distinct workstreams your project needs. Good workstreams are:

- **Independent** - Each can proceed without waiting for another
- **Focused** - Narrow enough for one agent to handle thoroughly
- **Concrete** - Clear deliverable at the end

Bad workstreams are vague ("look into SEO stuff") or deeply interdependent ("implement the frontend, but wait for the backend agent to finish the API first").

### Step 2: Write the Team Brief

Your prompt to Claude Code should describe:

- The overall project goal
- Each teammate's role and specific task
- What files or data each agent needs access to
- What format you want the deliverables in

Here is a simplified example:

```
Create a team to audit our client website:

Team of 3:
1. Technical SEO agent - crawl the site, check robots.txt,
  sitemap, page speed. Output: technical-audit.md
2. Content analyst - review the top 20 pages for keyword
  targeting, thin content, duplication. Output: content-audit.md
3. Competitor researcher - analyse the top 3 competitors
  for link gaps and content gaps. Output: competitor-report.md

Project files are in shared/projects/client-name/
```

### Step 3: Let the Team Lead Orchestrate

Once you submit the brief, the team lead agent takes over. It will:

- Parse your requirements
- Spawn the teammate agents
- Assign each one their workstream
- Monitor progress
- Compile results

You can watch the progress in real time. Each teammate's work appears in the Claude Code interface as it happens.

## Step 4: Review and Iterate

When the team finishes, review the deliverables. If something needs more depth, you can ask the team lead to send a follow-up task to a specific teammate, or spawn a new agent for additional investigation.

### Parallel vs Sequential: The Time Advantage



## When Agent Teams Make Sense (and When They Don't)

### Ideal Use Cases

- **Multi-site audits** - Like our plumbing franchise case study
- **Codebase refactoring** - One agent handles types, another handles tests, another handles documentation
- **Research projects** - Multiple agents investigating different aspects simultaneously
- **Content production** - We used Agent Teams to write this very blog series you are reading (more on that in a moment)
- **Data analysis** - Different agents analysing different data sources or metrics

### When to Stick with a Single Agent

- **Simple, linear tasks** - If steps must happen in sequence, parallelism adds overhead without benefit
- **Tiny tasks** - Fixing a typo does not need a team
- **Highly interdependent work** - If every step depends on the previous step's output, teammates will spend most of their time waiting

### The Meta Example

We used Agent Teams to produce this blog series. One agent wrote the hub post. Other agents wrote spoke posts in parallel. Each agent had the series brief, the keyword research, and the brand guidelines - and they all worked simultaneously.

The result: a complete content series drafted in a fraction of the time it would take a single writer working through each post sequentially. Every post maintains consistent messaging because every agent started from the same brief.



### Token Cost and Practical Considerations

Agent Teams is powerful, but it is not free. Here is what you should know about the economics.

#### Token Usage Scales Linearly

Each teammate consumes tokens independently. A team of four agents uses roughly four times the tokens of a single agent. There is no magic compression - each agent needs its own context to do its work.

#### When the Cost Is Worth It

For our plumbing franchise audit, the token cost was a fraction of what the equivalent human labour would have cost. Four analysts working for a full day versus 15 minutes of parallel AI execution - the maths speaks for itself.

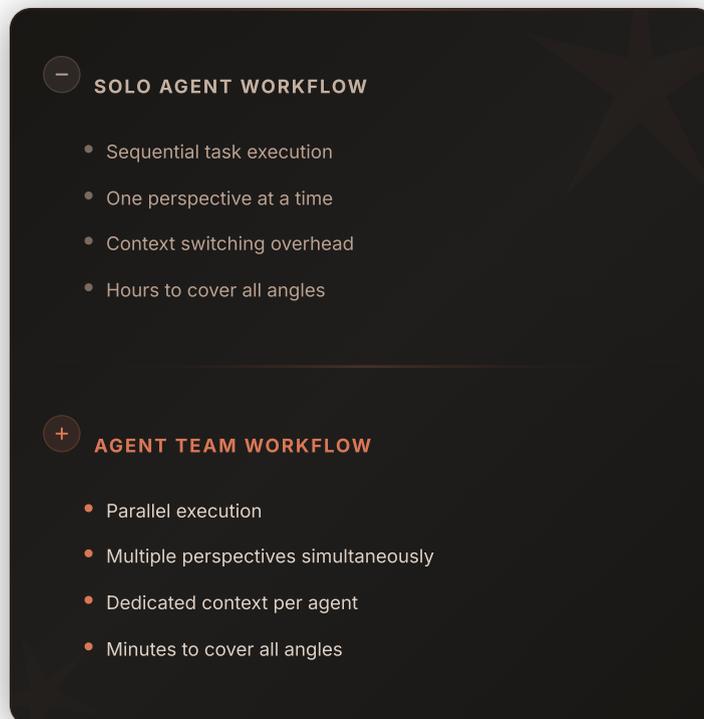
The cost calculus favours Agent Teams when:

- **Time is valuable** - Parallel execution saves hours or days
- **The work is genuinely parallelisable** - Agents are not waiting on each other

- **Accuracy matters** - Independent corroboration catches errors that sequential work misses
- **The project is complex enough** - Simple tasks do not benefit from team overhead

### Tips for Managing Costs

- **Be specific in your briefs** - Vague instructions cause agents to explore unnecessarily
- **Scope teammates tightly** - Each agent should know exactly what to deliver
- **Start small** - Test with 2 agents before scaling to 5
- **Use Plan Mode first** - Map out your workstreams before spawning the team (more on Plan Mode in our [companion post](#))



### What This Means for Agencies and Teams

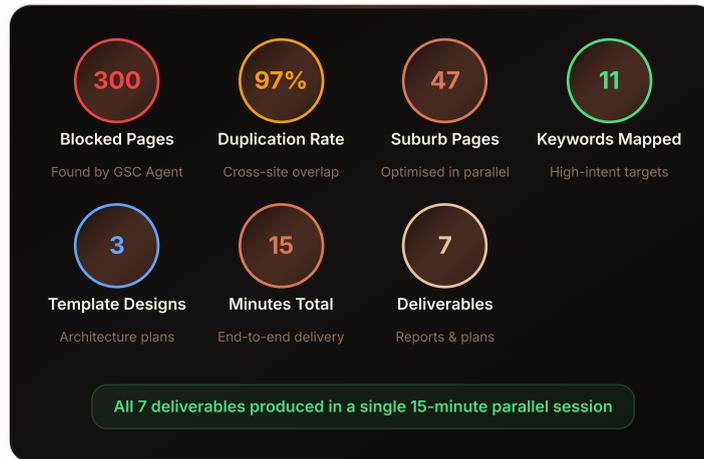
Agent Teams is not just a developer tool. It is a force multiplier for any team doing complex knowledge work.

At Jordan James Media, we have integrated Agent Teams into our audit and research workflows. A project that previously required three team members working across two days can now be scoped, executed, and compiled in under an hour.

This does not replace our team. It amplifies them. Our strategists spend less time on data gathering and more time on the analysis and creative thinking that actually moves the needle for clients.

For agencies evaluating AI tools, Agent Teams represents a step change. It is the difference between having one AI assistant and having a coordinated AI department.

## Mr Splash Audit: Results Dashboard



### Key Takeaways

1. **Agent Teams runs multiple AI agents in parallel**, coordinated by a team lead that delegates, monitors, and compiles results
2. **Real-world tested** - We ran it on a 20+ site plumbing franchise and got 7 deliverables in 15 minutes
3. **Independent corroboration** - Two agents flagged the same critical issue without communicating, providing built-in quality assurance
4. **Token costs scale linearly** - A team of 4 costs roughly 4x a single agent, but the time savings often justify the spend
5. **Best for parallelisable work** - Audits, research, content production, and multi-file refactoring are ideal use cases
6. **Plan your workstreams first** - Use **Plan Mode** to scope the team before you launch it
7. **We used it to write this series** - The content you are reading was produced by an Agent Team working in parallel

#### PRO TIP



#### Setting Up Your First Agent Team

Start with 2 agents on a real task — one for research, one for implementation. Once you see the speed gain, scale to 3-4 agents with distinct roles. Never assign overlapping file scopes.

### Ready to Put AI Teams to Work on Your Projects?

Jordan James Media is a Sydney-based digital marketing agency that uses AI-first workflows to deliver faster, more thorough results for Australian businesses. If you want to see

what Agent Teams and advanced AI tooling can do for your marketing, we would love to chat.

### Talk to Us About AI-Powered Marketing

#### Related Reading:

- [Claude Code's Biggest Update: 7 Features That Changed How We Build](#)
- [Claude Code Plan Mode: Think Before You Build](#)
- [Claude Code Checkpoints: Never Lose Your Work Again](#)

#### Launching Parallel Agents

```
terminal  
  
1 # Terminal 1: GSC Analyst  
2 claude --task "Audit GSC data for mr-splash-plumbing"  
3  
4 # Terminal 2: SEO Researcher  
5 claude --task "Research competitor keywords"  
6  
7 # Terminal 3: Dev Planner  
8 claude --task "Plan location page template"
```

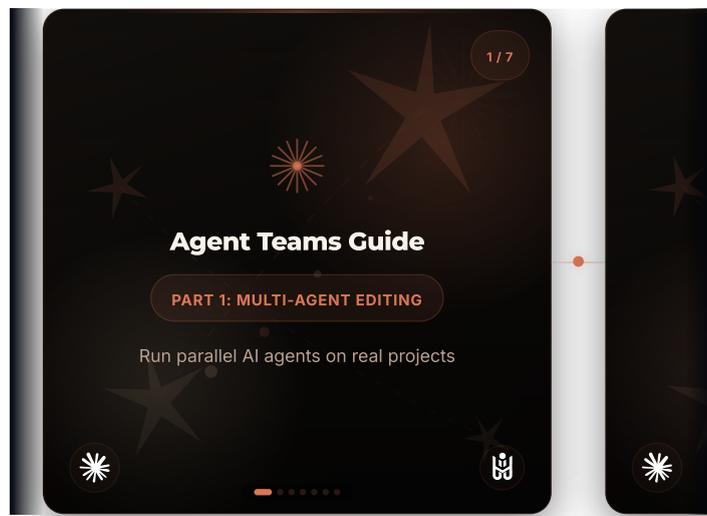
Each agent runs in its own terminal with a focused task. They work simultaneously without interfering with each other's file operations.

#### Social Media Carousel

7 cards — Scroll to browse, click any card to download

PDF for  
LinkedIn

ZIP  
Images



 **Share This Article**  
Spread the knowledge

 Twitter

 LinkedIn

 Copy



## Build With Claude Code

We use Claude Code to ship faster, smarter, and at scale. Let us build your next project.

[Get Your Free AI Strategy](#) →

[App Development](#)



**Claude Code**  
8 parts

Part 0

Part 1

Part 2

Part 3

Part 4

Part 5

Part 6

Part 7

[View All](#)

### Quick Links

- [→ Series Home](#)
- [→ App Development](#)
- [→ Get In Touch](#)

### Claude Code Stats

Model	Opus 4.6
Features	7
Series Parts	8



### Agency Tested

Real projects, real results

Ready to **outperform** your  
competition?

Get your free AI strategy consultation today.





**Jordan James Media**  
GROWTH STUDIO

Australia's AI-first digital marketing agency. We build intelligent systems that drive exponential growth.



info@jordanjamesmedia.com



0475 897 737



Sydney, Australia

● SERVICES

SEO Services

Google Ads

Website Design

Local SEO

Content Marketing

App Development

● COMPANY

About Us

Blog

Case Studies

Pricing

Tools

Contact

● LOCATIONS

Sydney

Melbourne

Brisbane

Perth

Adelaide

All Locations

All Locations →



