

Devchain

AI-powered multi-team development workflow

v1.0 — April 2026

1 Install & Launch

Install the CLI globally, then start Devchain from the root of your project:

TERMINAL

```
$ npm i -g devchain-cli  
# Start from your project  
$ devchain start
```

i The browser UI opens automatically after running `devchain start`.

2 Create a Project

Create a new project using the Teams template. This template ships with two purpose-built teams that manage themselves.

Teams Template ← recommended

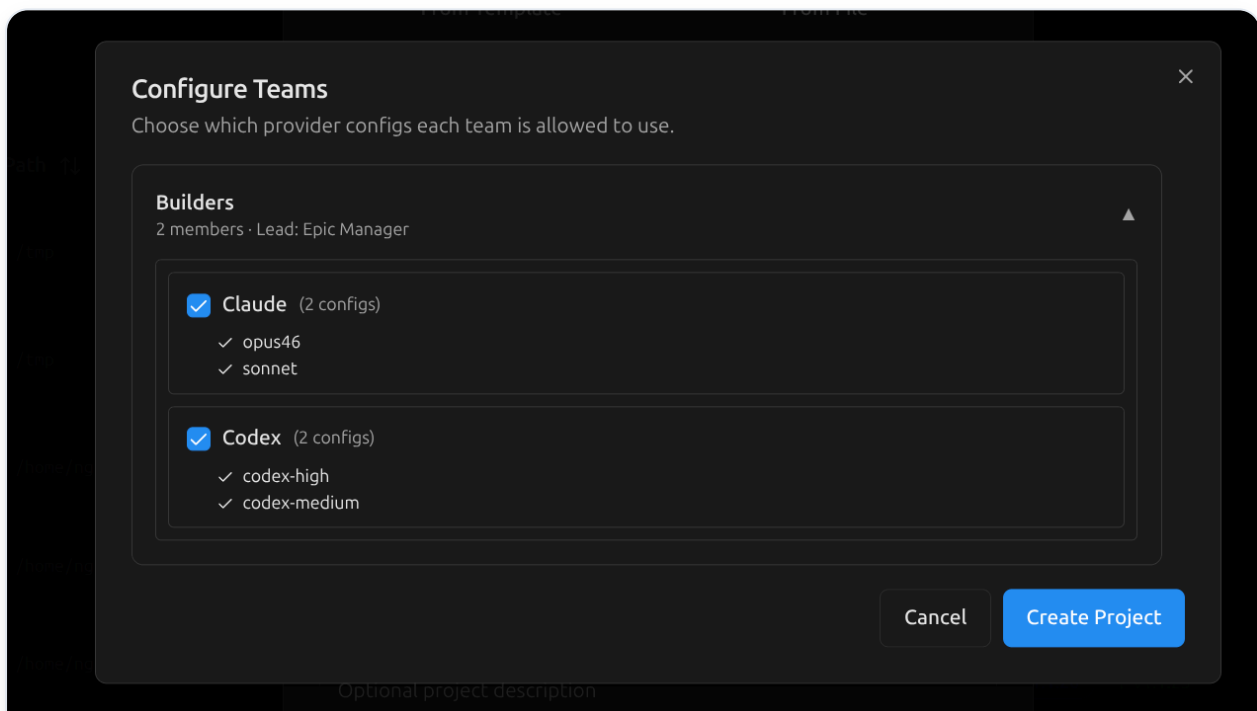
Two teams that manage themselves.

Planning team (Brainstormer + Architect) drafts and validates the plan. **Builders team** (Epic Manager + Coders) auto-scales and picks the right model per task.

5-Agent / 3-Agent Templates

Classic flat workflows still bundled and supported. Pick these if you prefer a fixed agent roster — see the final page.

Select the Teams Development template. Set the project name and path. A **Configure Teams** dialog appears as part of the create flow.



Configure Teams — pick which provider configs the Builders team is allowed to use (Claude, Codex, or both).

i Pick at least one provider. The team lead is bounded to your selection when it spawns workers — selecting both Claude and Codex gives it more room to mix model tiers across tasks.

3 Development Flow — Teams Setup

New → In Progress → Review → Done (or Blocked when necessary)

1 Initialize Brainstormer to discuss your plan or feature.

- › As the Planning team lead, Brainstormer validates project docs first, then discusses requirements and helps shape the plan.

2 Brainstormer drafts plan and the Planning team validates it

- › Brainstormer hands the draft to the rest of the Planning team. They check it against the actual codebase — file paths, dependencies, completeness, over-engineering — and reply with feedback. Multiple validation rounds are possible.
- › *Optional:* for ambiguous or architecturally novel work, Brainstormer can ask the Planning team to research the user's request **independently in parallel**, gathering several framings before drafting.

3 Validated plan presented to user; Brainstormer decomposes on approval

- › Plan includes acceptance criteria and relevant skills. Once approved, Brainstormer creates Phase epics ([Draft](#)) and sub-epic tasks ([New](#)).

4 User assigns Phase epics to Epic Manager (Builders team lead)

- › From the Board, move the phases you want completed in the next cycle from [Draft](#) to [New](#) and assign to **Epic Manager**.

5 Builders team auto-scales and executes

- › Epic Manager evaluates each sub-epic and picks the right model for the job — cheaper models for routine tasks, top-tier models for harder work. It reuses idle Coders before spawning new ones, and batches related tasks onto the same Coder to keep context loaded.
- › When all tasks in a phase are done, Epic Manager requests Code Review automatically.

6 Code Reviewer audits working tree changes

- › Reviews uncommitted code for architecture, errors, security, performance, and code quality.

7 Review outcome

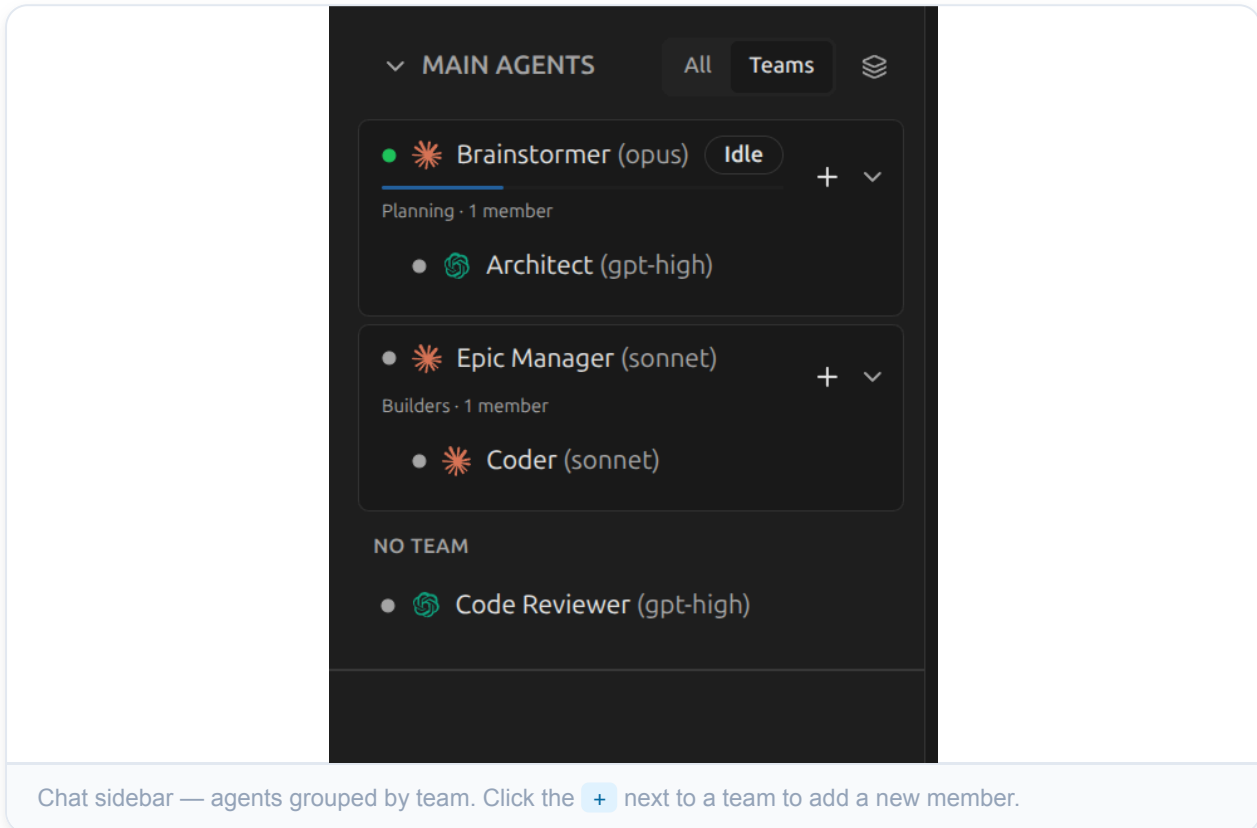
APPROVED: Reviewer moves epics to [Done](#) (user commits manually).

ISSUES FOUND: Reviewer sends remediation plan to Brainstormer; epic stays in [Review](#) .

Remediation cycle (if needed): Brainstormer creates a new remediation parent epic. Assign it to Epic Manager, who routes the fix through the Builders team and triggers another Code Review.

4 Chat Page — Teams View

The Chat page sidebar shows your agents grouped by team. Toggle **Teams** at the top to see the team layout: Planning under the Brainstormer, Builders under the Epic Manager, and any agents that don't belong to a team listed at the bottom.



Provider Config (per agent)

Right-click any agent and open **Provider Config** to switch its provider or override the model. The team lead respects each member's individual config when routing tasks.

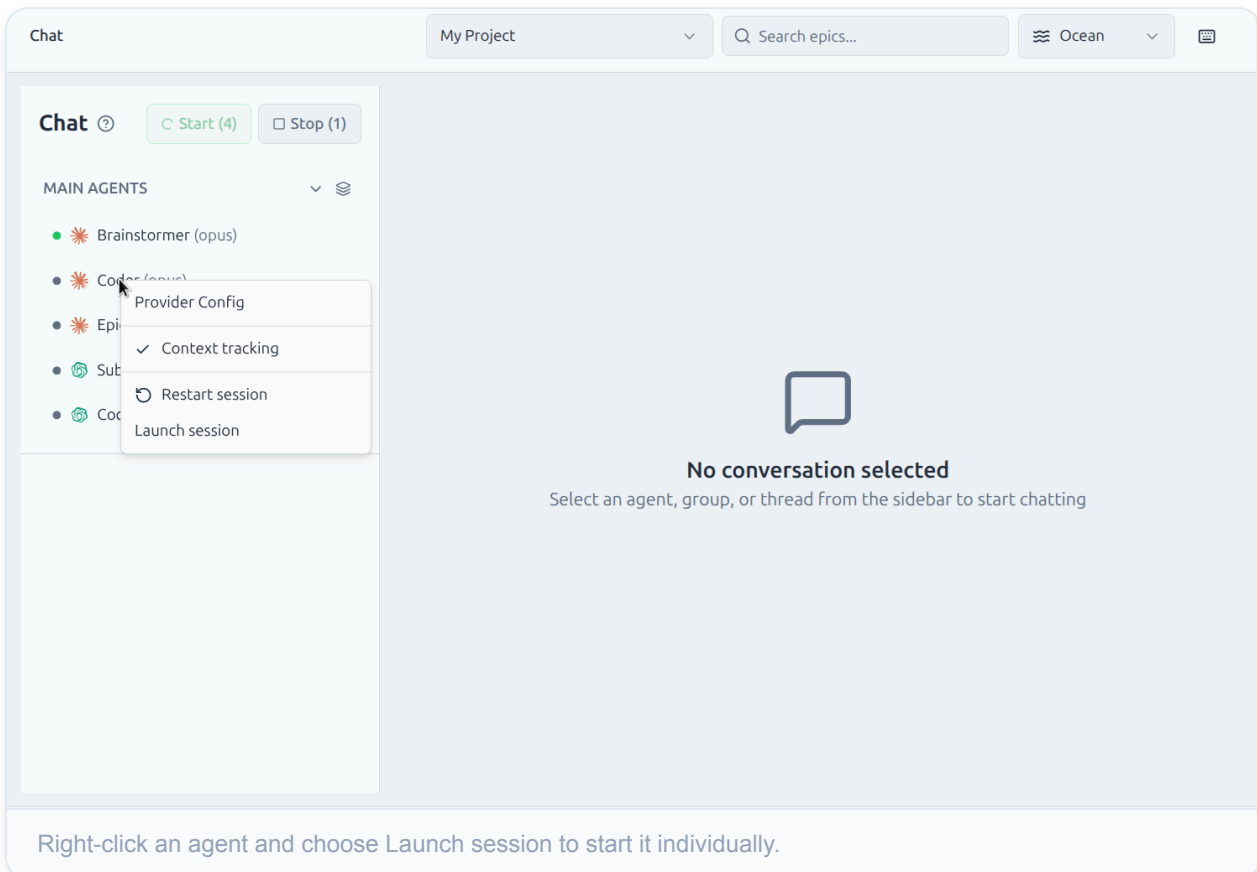
Configure Teams (per team)

The provider configs you picked at project creation gate which configs the team lead is allowed to use when spawning new workers. You can adjust this at any time from the team's edit dialog (see Page 11).

5 Start All or Launch Brainstormer

From the Chat page, click **Start** to launch all agents at once, or right-click an individual agent and choose **Launch session** to start just that one.

i Starting only Brainstormer first is a good way to discuss and refine your plan before the rest of the Planning team and the Builders join in.



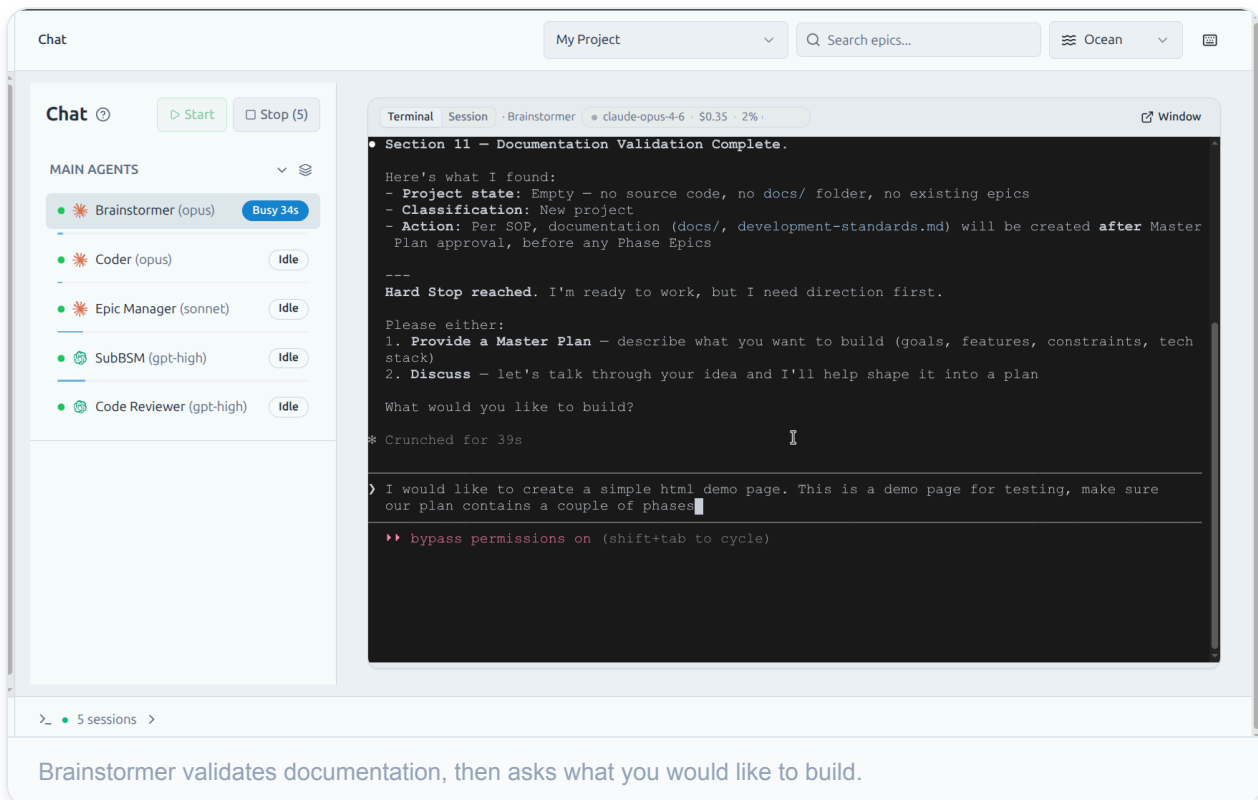
The screenshot shows the 'Chat' interface for 'My Project'. The sidebar on the left is titled 'Chat' and contains a 'Start (4)' button and a 'Stop (1)' button. Below these are 'MAIN AGENTS' with a list of agents: 'Brainstormer (opus)', 'Codes (opus)', 'Epi...', 'Sub...', and 'Coc...'. A context menu is open over the 'Codes (opus)' agent, showing options: 'Provider Config', 'Context tracking', 'Restart session', and 'Launch session'. The main chat area is empty and displays a message bubble icon with the text 'No conversation selected' and 'Select an agent, group, or thread from the sidebar to start chatting'. The top of the interface includes a search bar for 'Search epics...' and a 'Ocean' dropdown menu.

Right-click an agent and choose Launch session to start it individually.

6 Discuss Your Plan with Brainstormer

Open the Brainstormer terminal. Once Brainstormer finishes its initial steps (validating project docs, checking skills), it will prompt you to discuss your plan, change, or research.

- Describe what you want to build or change. Brainstormer will research, verify, and help shape it into a solid plan.



The screenshot displays the Brainstormer chat interface. On the left, a sidebar lists 'MAIN AGENTS' including Brainstormer (opus), Coder (opus), Epic Manager (sonnet), SubBSM (gpt-high), and Code Reviewer (gpt-high). The main area shows a terminal window titled 'Terminal' with the following content:

```
Section 11 - Documentation Validation Complete.
Here's what I found:
- Project state: Empty - no source code, no docs/ folder, no existing epics
- Classification: New project
- Action: Per SOP, documentation (docs/, development-standards.md) will be created after Master Plan approval, before any Phase Epics

---
Hard Stop reached. I'm ready to work, but I need direction first.

Please either:
1. Provide a Master Plan - describe what you want to build (goals, features, constraints, tech stack)
2. Discuss - let's talk through your idea and I'll help shape it into a plan

What would you like to build?

* Crunched for 39s

> I would like to create a simple html demo page. This is a demo page for testing, make sure our plan contains a couple of phases
  >> bypass permissions on (shift+tab to cycle)
```

Below the terminal window, a caption reads: "Brainstormer validates documentation, then asks what you would like to build."

7 Planning Team Validates the Plan

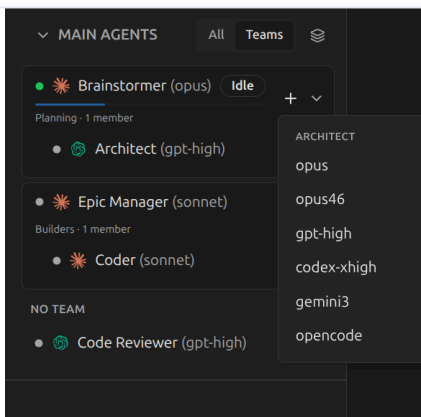
After discussing and refining the plan, Brainstormer hands the draft to the rest of the **Planning team** for technical validation before presenting it to you.

What the Planning team checks:

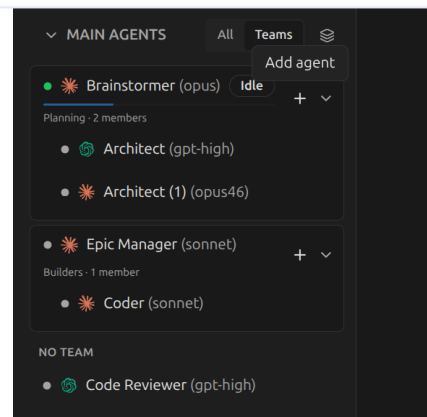
- Verifies file paths and project structure against the actual codebase
- Identifies over-engineering or missing dependencies
- Validates completeness of the plan
- Multiple validation rounds run automatically before the plan reaches you

Optional — scale the Planning team for parallel research

For ambiguous or architecturally novel work, you can add more Architects to the Planning team — each on a different model (Opus, Codex, Gemini, OpenCode...). Brainstormer can then ask them to research the request **independently in parallel**, gathering several framings before drafting. Diverse perspectives, fewer blind spots.



Click **+** on the Planning team to add an Architect on any provider config.



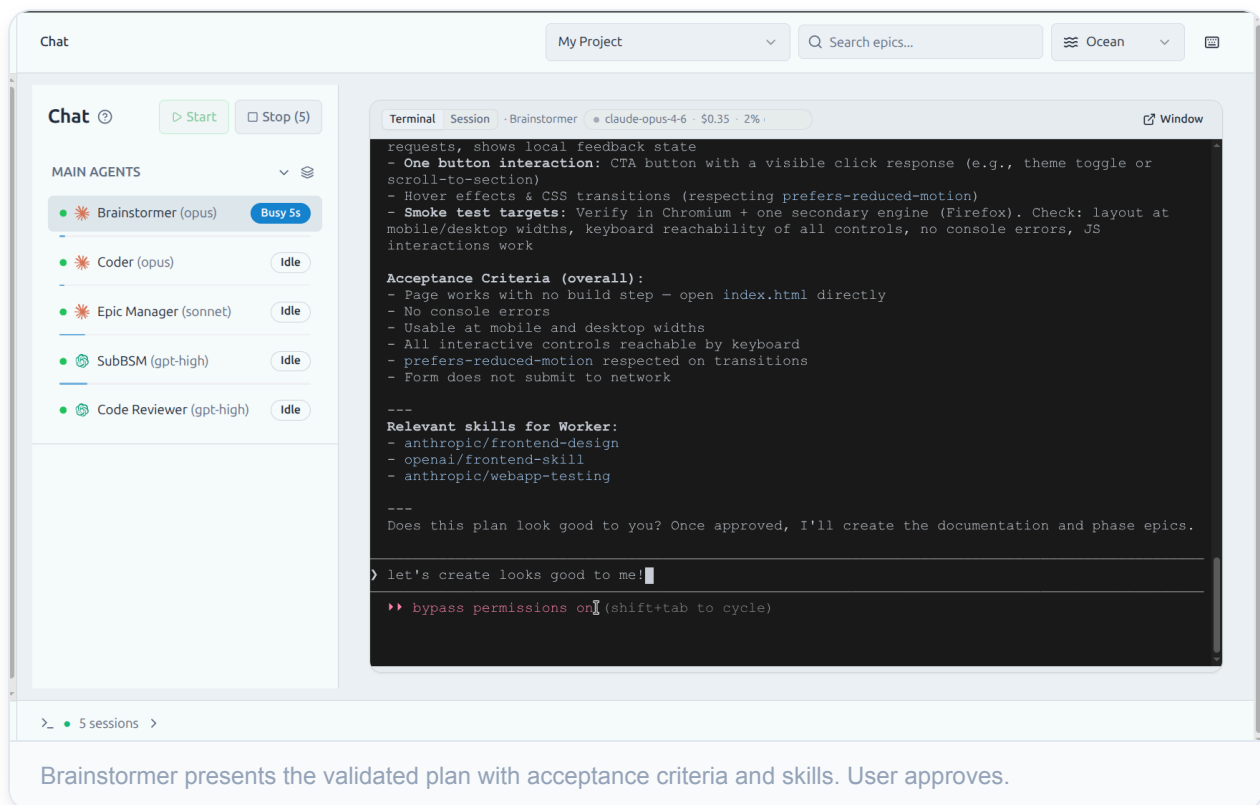
Two Architects on different models — two independent framings.

i You do not need to interact with the Planning team directly. Validation happens automatically between Brainstormer and the team members.

8 Approve & Decompose the Plan

Once the Planning team has validated the plan, Brainstormer presents the final version to you with **acceptance criteria** and **relevant skills**. Approve it to trigger decomposition into phases, epics, and sub-tasks.

- Review the validated plan. When ready, confirm to Brainstormer to create the documentation and decompose into phase epics.



The screenshot shows a chat window titled "Chat" with a dropdown menu for "My Project" and a search bar for "Search epics...". The chat area displays a message from "Brainstormer (opus)" which is currently "Busy 5s". The message content is as follows:

```
requests, shows local feedback state
- One button interaction: CTA button with a visible click response (e.g., theme toggle or scroll-to-section)
- Hover effects & CSS transitions (respecting prefers-reduced-motion)
- Smoke test targets: Verify in Chromium + one secondary engine (Firefox). Check: layout at mobile/desktop widths, keyboard reachability of all controls, no console errors, JS interactions work

Acceptance Criteria (overall):
- Page works with no build step - open index.html directly
- No console errors
- Usable at mobile and desktop widths
- All interactive controls reachable by keyboard
- prefers-reduced-motion respected on transitions
- Form does not submit to network

---
Relevant skills for Worker:
- anthropic/frontend-design
- openai/frontend-skill
- anthropic/webapp-testing

---
Does this plan look good to you? Once approved, I'll create the documentation and phase epics.

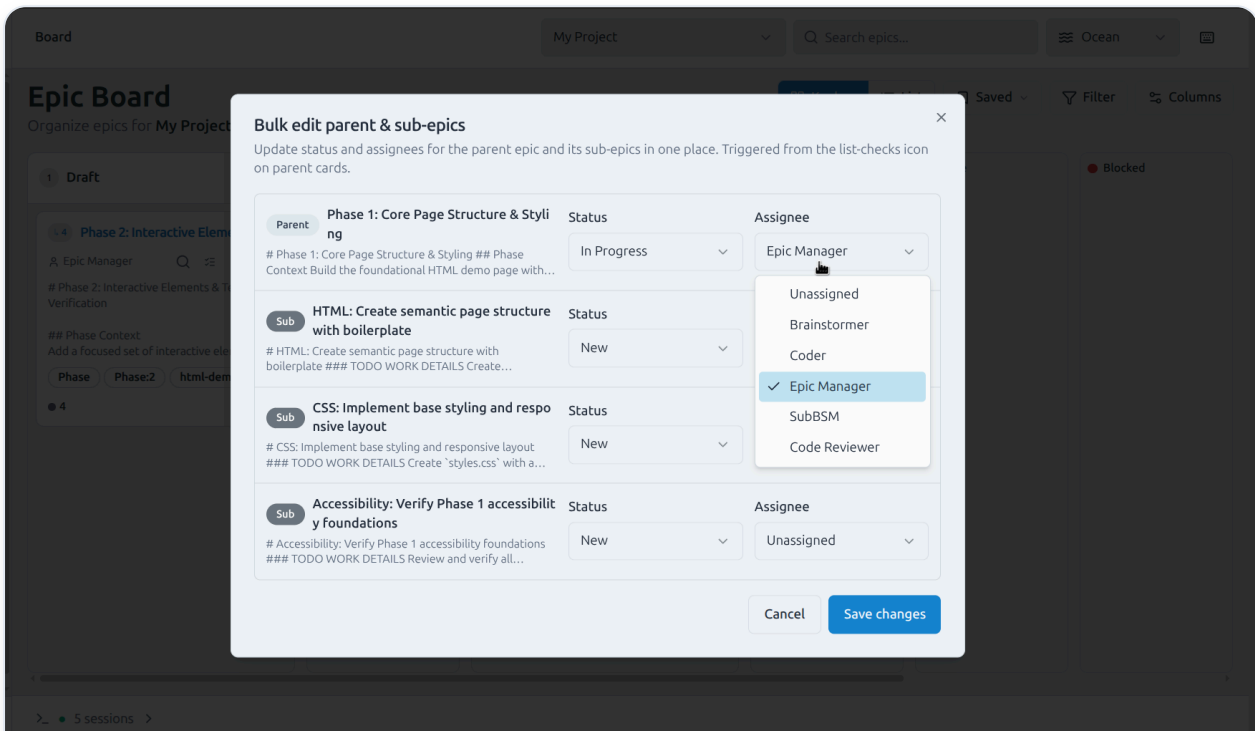
> let's create looks good to me!
>> bypass permissions on (shift+tab to cycle)
```

Below the chat window, a caption reads: "Brainstormer presents the validated plan with acceptance criteria and skills. User approves."

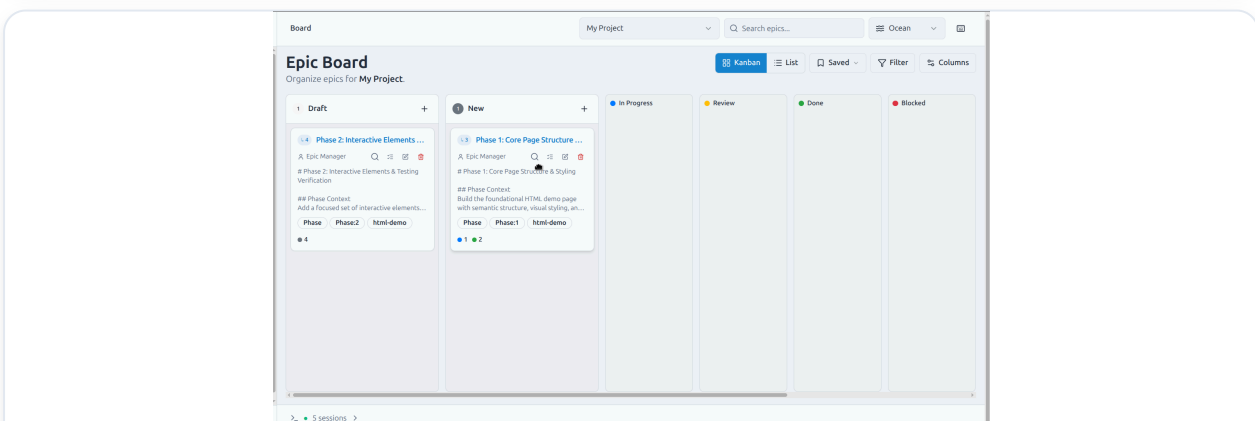
- i After approval, Brainstormer creates the documentation and decomposes the plan into Phase epics and sub-epic tasks on the Board. Head to the Board page to manage epics.

9 Board — Assign Phases to Epic Manager

From the **Board** page you can browse created epics in a Kanban view. Move the Phase epics you want completed in the next cycle from **Draft** to **New** and assign them to **Epic Manager** — the Builders team lead. Epic Manager will then route tasks across the Builders team automatically.



Bulk edit — set status and assign the Phase epic to Epic Manager.

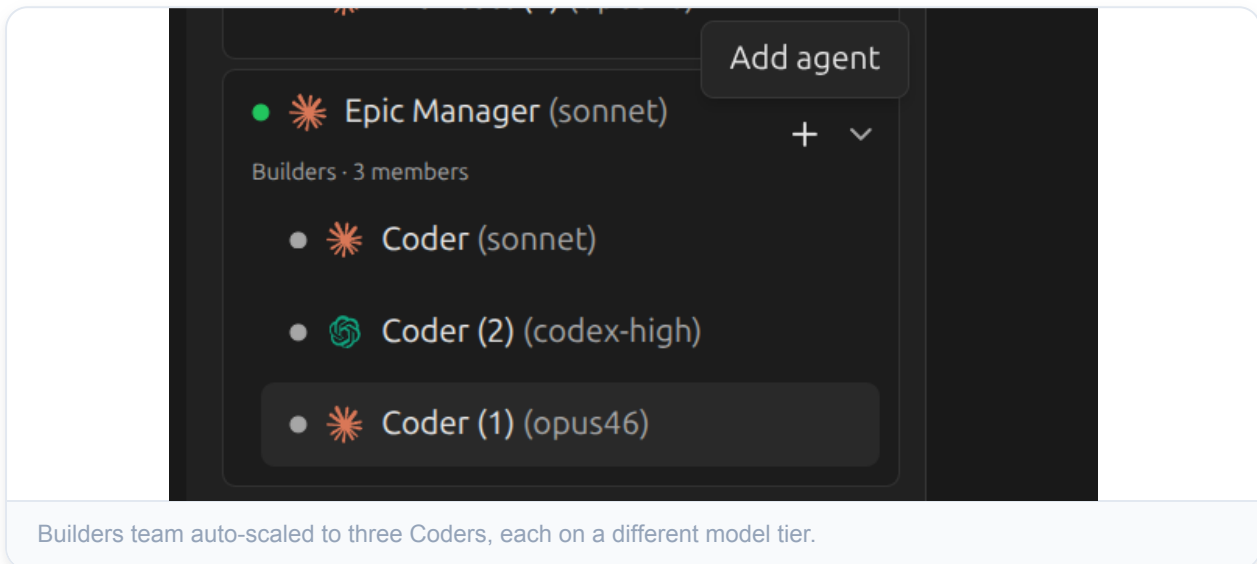


Kanban view — Phase 1 assigned to Epic Manager and moved to In Progress.

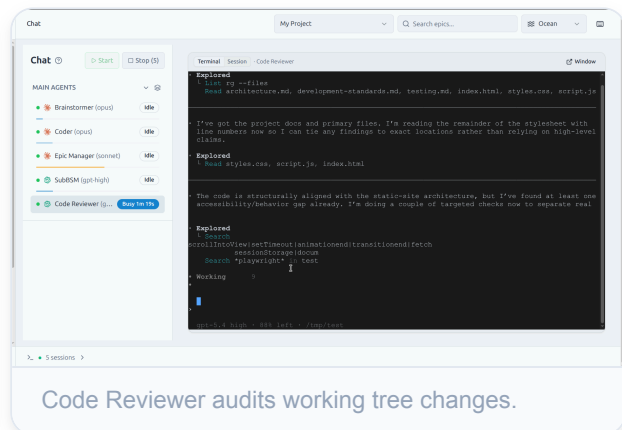
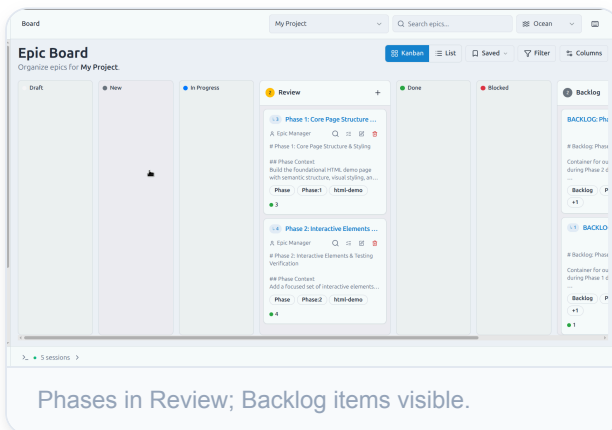
i You assign at the **Phase** level, not at sub-task level. Epic Manager handles the routing inside the Builders team.

10 Builders Auto-Scale & Code Review

Epic Manager evaluates each sub-epic and routes it to the right Coder — cheaper models for routine work, top-tier models for harder changes. As workload grows it adds new Coders up to the team's capacity caps; idle Coders are reused before new ones are spawned.



Once all tasks in a phase are done, Epic Manager triggers Code Review automatically.



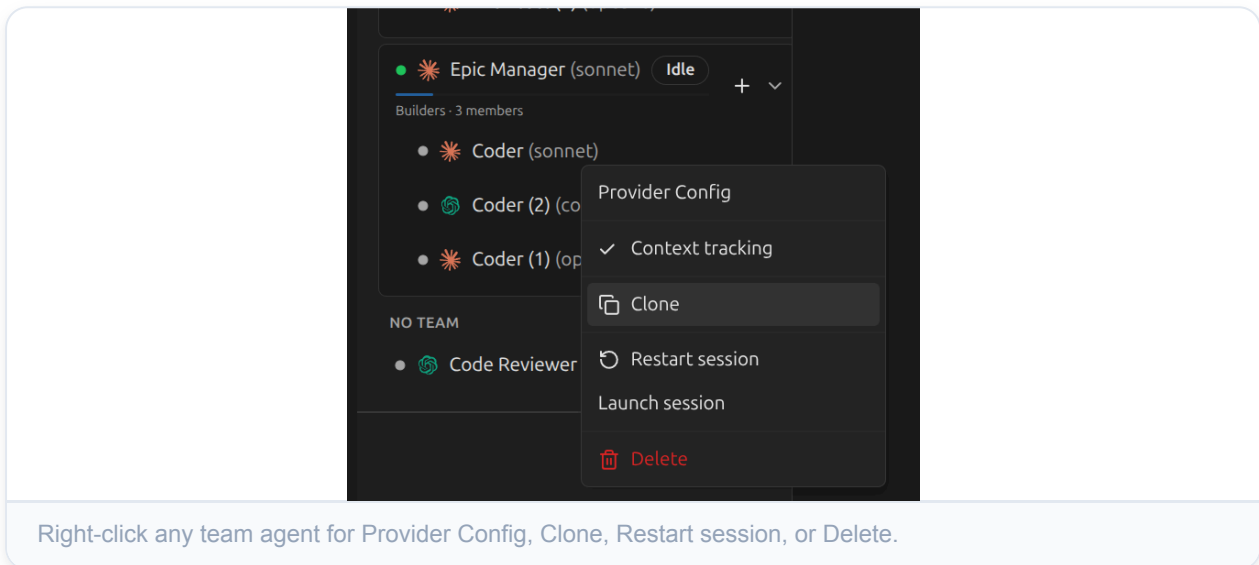
- Code Reviewer **approves** → epic moves to Done. Or sends review details to Brainstormer to create a **remediation epic**.
- Assign remediation epics to **Epic Manager**, who routes the fix through the Builders team and triggers another Code Review.
- From the **Review page** you can check code changes and send comments to any agent.
- Once happy, **ask a Coder to commit** or do it yourself! Repeat the flow for new plans and features.

11 Manual Team Management

Auto-scaling is a default, not a lock-in. Every team can be steered manually from the Chat sidebar.

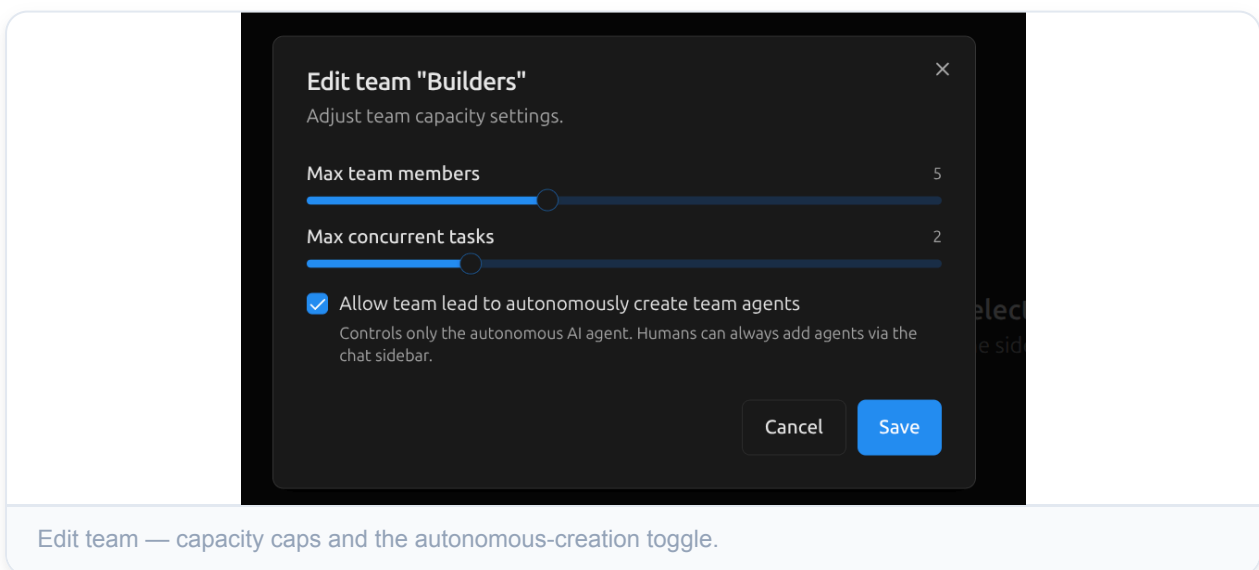
Clone, restart, or delete any agent

Right-click any team member to **Clone** it (carries the same role and provider), **Restart session**, or **Delete**. Team membership is no longer fixed by the template — it's a live roster.



Edit team capacity and disable autonomous scaling

Right-click a team header and pick **Edit team** to adjust **Max team members**, **Max concurrent tasks**, and toggle **Allow team lead to autonomously create team agents**. Turn the toggle off and the team lead stops spawning workers entirely — you decide every roster change yourself.



i Disabling autonomous creation is useful when you want hands-on control over the Builders roster — e.g. to keep token usage strictly bounded for a small project.

! Note: Other Templates Available

This guide describes the **Teams template** — the recommended flow for new projects. Devchain also bundles two flat-roster templates that remain fully supported.

Flat-roster alternatives

5-agents-dev

Brainstormer, SubBSM, Epic Manager, Coder, Code Reviewer. The classic flat workflow with a fixed roster — no auto-scaling, no parallel planning.

3-agents-dev

Brainstormer, SubBSM, Coder. The most token-efficient flow — you manage the Board manually and assign Phase epics directly to Coder.

Why Teams is the recommended default

1 Cheaper on average

› Builders pick the right model per task — trivial work no longer pays top-tier prices.

2 Faster on parallel work

› Builders scale with the workload, and Planning can research from several angles in parallel for better master plans.

3 Cleaner mental model

› Two team leads with clear ownership replace a flat 5-agent matrix — you assign at the team level, the team handles the rest.

i Existing projects on 5-agents-dev or 3-agents-dev keep working — nothing breaks. New projects should start from the Teams template.