

Devchain

AI-powered multi-agent 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 a pre-built agent template:

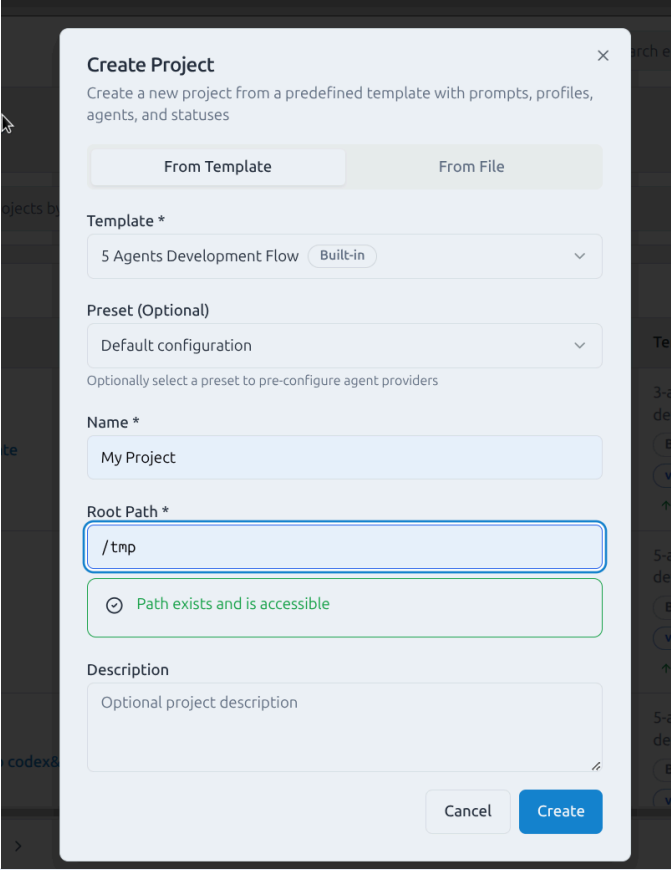
3-Agent Template

Best for basic subscriptions. Most token-efficient setup with Brainstormer, Coder, and Code Reviewer.

5-Agent Template ←

Best results for complex projects. Includes two additional agents for plan validation and automated epic management.

Select the 5 Agents Development Flow template. Set the project name and path. Keep the template presets at their defaults.



Create Project ×

Create a new project from a predefined template with prompts, profiles, agents, and statuses

Template *
5 Agents Development Flow Built-in ▼

Preset (Optional)
Default configuration ▼

Optionally select a preset to pre-configure agent providers

Name *
My Project

Root Path *
/tmp

Path exists and is accessible

Description
Optional project description

Create Project — select the 5-Agent template, set name & path, then click Create.

3 Development Flow — 5-Agent Setup

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

- 1 Initialize all agents or just Brainstormer** to discuss your plan/feature implementation.
 - › Validates/creates project docs first, then discusses requirements and helps shape the plan.
- 2 Brainstormer drafts plan and sends to SubBSM**
 - › SubBSM validates the draft against the actual codebase — checking file paths, dependencies, completeness, and flagging over-engineering. Can run multiple validation rounds.
- 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**
 - › From the Board, move the phases you want completed in the next cycle from **Draft** to **New** and assign to **Epic Manager** (not Coder).
- 5 Epic Manager coordinates execution**
 - › Assigns tasks to Coder sequentially. Reviews each completed sub-epic (approves or sends back).
 - › When all tasks in a phase are done, 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 coordinates the fix with Coder and triggers another Code Review.

4 Chat Page & Provider Configuration

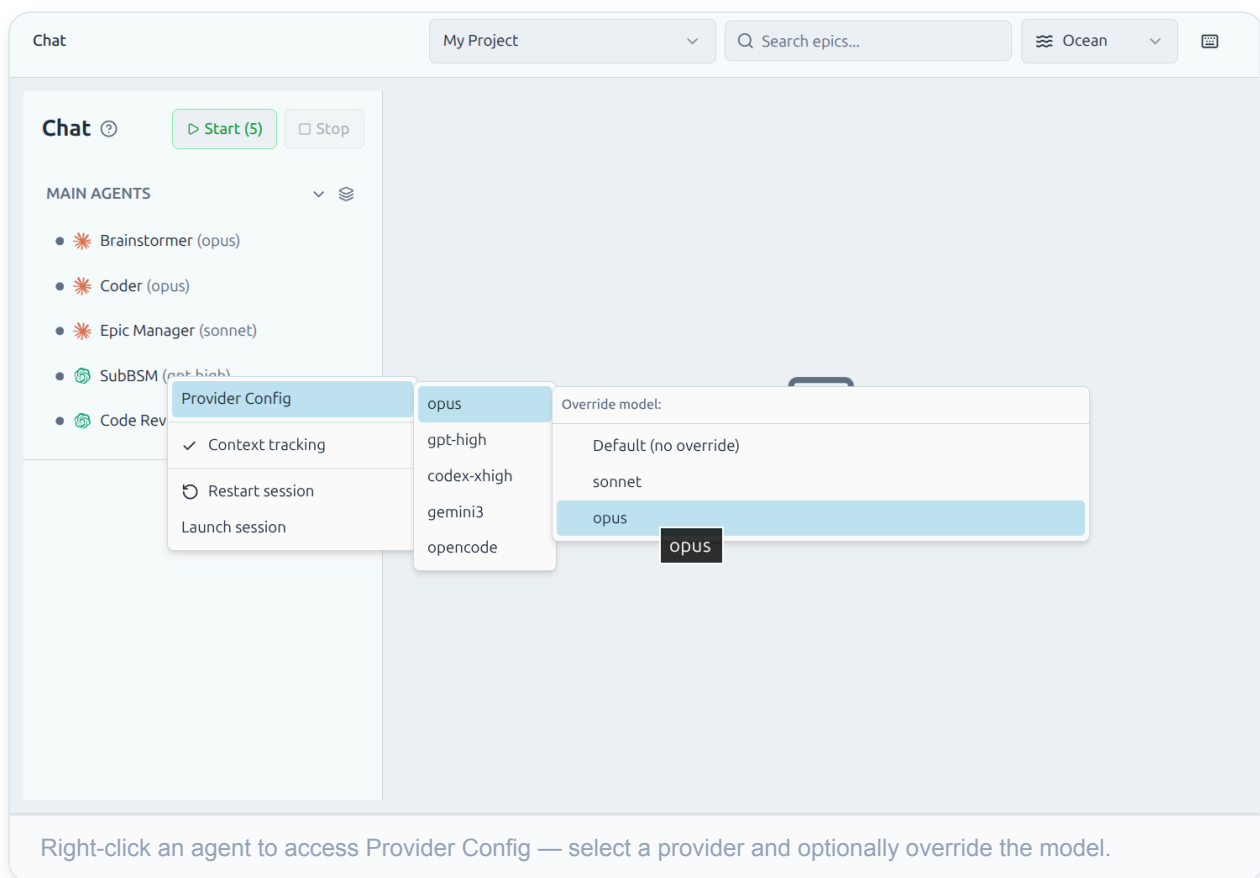
The Chat page shows all five agents and their current AI provider assignments.

Claude + Codex subscriptions (default)

Leave the default setup as-is: **Claude Opus** for Brainstormer and Coder, **Opus with Sonnet override** for Epic Manager, **Codex / GPT-High** for SubBSM and Code Reviewer.

Claude-only subscription

Right-click on SubBSM and Code Reviewer agents and open **Provider Config** to change their provider from GPT-High to **Opus**.



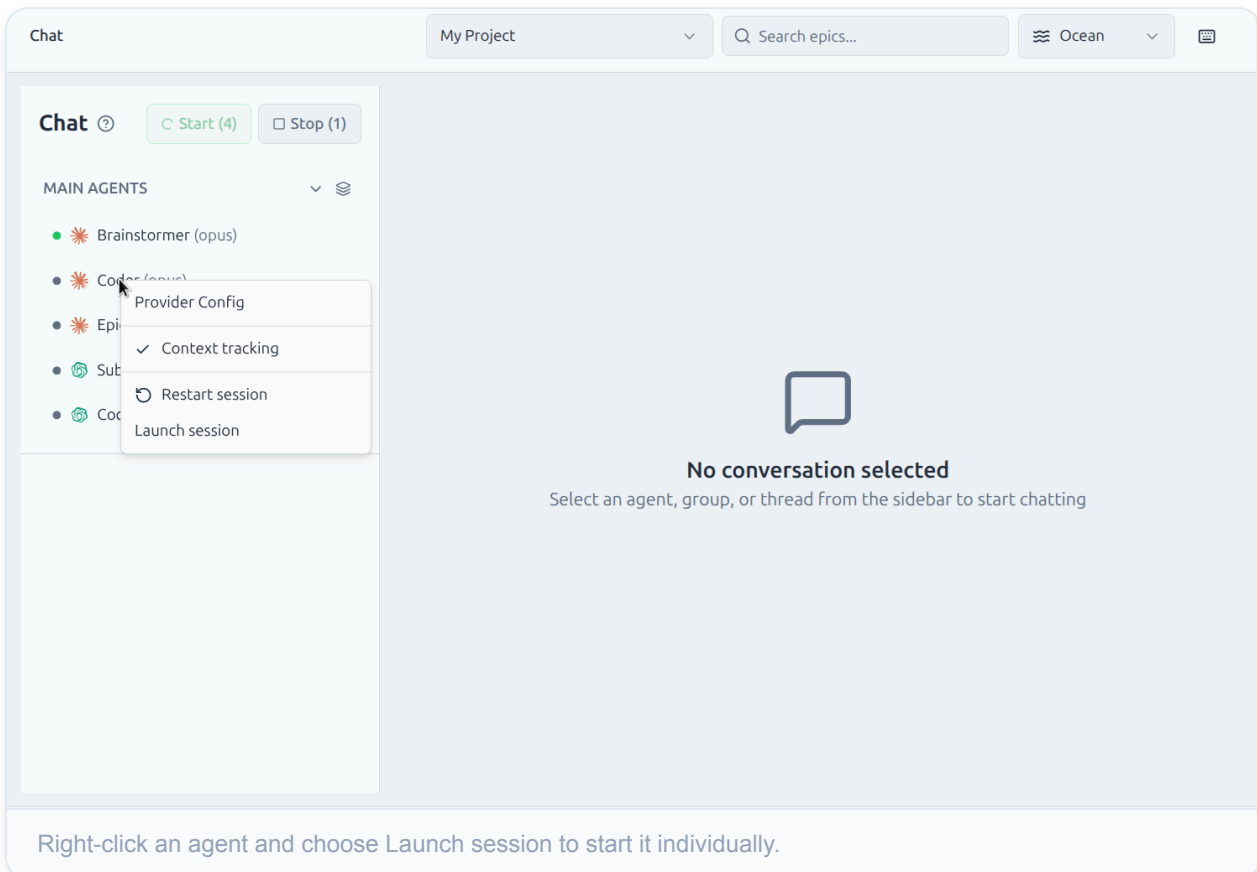
The screenshot shows the Chat page interface. At the top, there's a header with "Chat", "My Project", a search bar "Search epics...", and a user profile "Ocean". Below the header, there's a "Chat" section with a "Start (5)" button and a "Stop" button. A list of "MAIN AGENTS" is shown: Brainstormer (opus), Coder (opus), Epic Manager (sonnet), SubBSM (gpt-high), and Code Rev (gpt-high). A context menu is open over the "Code Rev" agent, showing options: "Provider Config", "Context tracking", "Restart session", and "Launch session". The "Provider Config" option is selected, opening a sub-menu with provider options: "opus", "gpt-high", "codex-xhigh", "gemini3", and "opencode". The "opus" option is selected, opening an "Override model:" dialog with options: "Default (no override)", "sonnet", and "opus". The "opus" option is selected in the dialog.

Right-click an agent to access Provider Config — select a provider and optionally override the model.

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 bringing in the other agents.



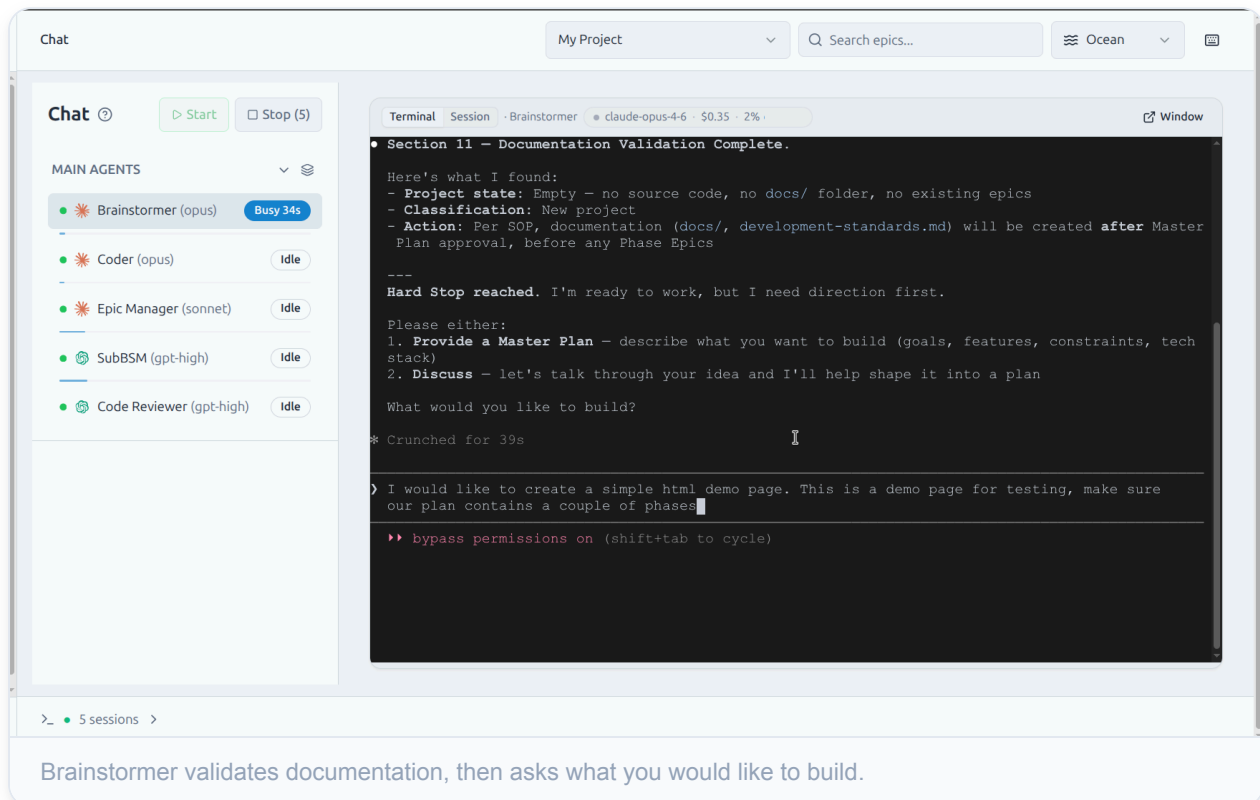
The screenshot shows the Chat interface with a sidebar on the left and a main chat area on the right. The sidebar is titled 'Chat' and has a 'Start (4)' button and a 'Stop (1)' button. Below these are 'MAIN AGENTS' with a list of agents: Brainstormer (opus), Code (opus), Epi, Sub, and Coc. A context menu is open over 'Code (opus)', showing options: Provider Config, Context tracking (checked), Restart session, and Launch session. The main chat area displays 'No conversation selected' and a message: 'Select an agent, group, or thread from the sidebar to start chatting'.

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, Busy 34s), Coder (opus, Idle), Epic Manager (sonnet, Idle), SubBSM (gpt-high, Idle), and Code Reviewer (gpt-high, Idle). The main area shows a terminal window titled 'Terminal Session · Brainstormer · claude-opus-4-6 · \$0.35 · 2%'. The terminal output includes:

```
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)
```

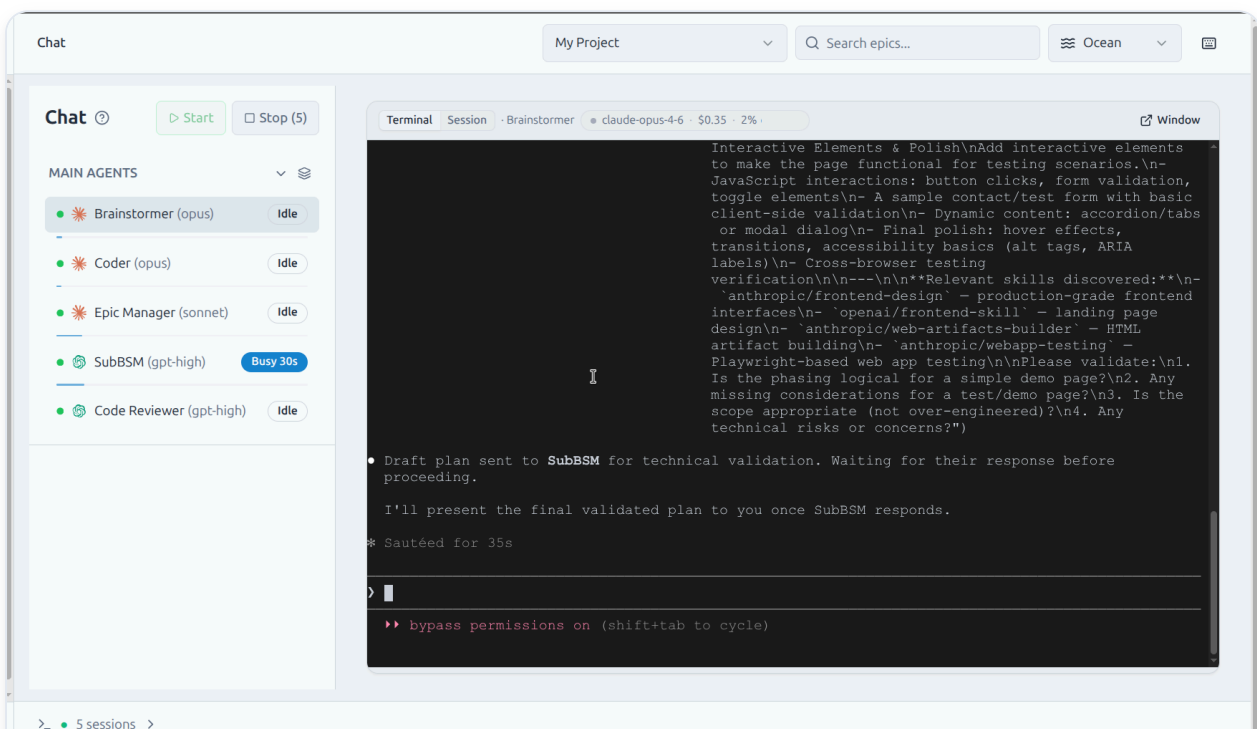
Below the terminal window, a caption reads: 'Brainstormer validates documentation, then asks what you would like to build.'

7 SubBSM Validates the Plan

After discussing and refining the plan, Brainstormer automatically sends the draft to **SubBSM** for technical validation before presenting it to you. This is a key difference from the 3-agent flow.

What SubBSM checks:

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



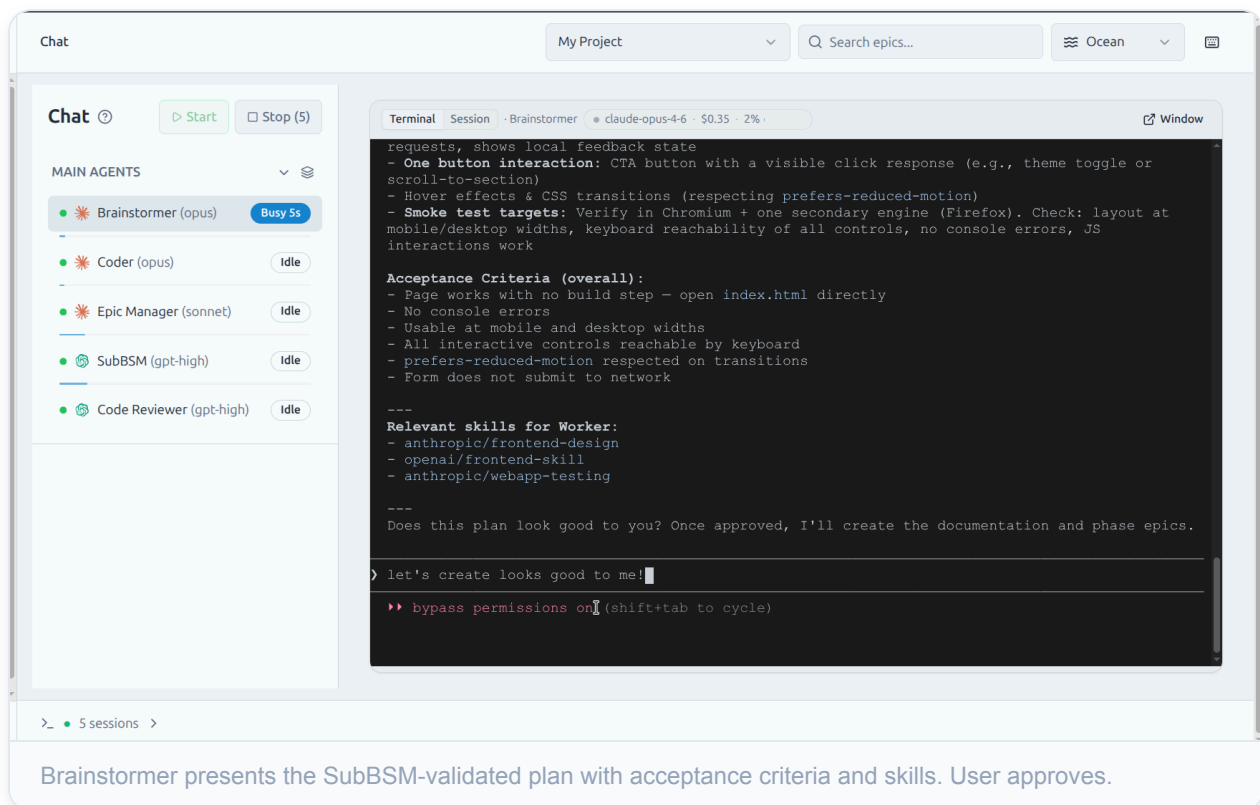
Brainstormer sends the draft plan to SubBSM for technical validation and waits for a response.

i You do not need to interact with SubBSM directly. The validation happens automatically between Brainstormer and SubBSM.

8 Approve & Decompose the Plan

Once SubBSM 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 sidebar on the left listing "MAIN AGENTS": Brainstormer (opus) - Busy 5s, Coder (opus) - Idle, Epic Manager (sonnet) - Idle, SubBSM (gpt-high) - Idle, and Code Reviewer (gpt-high) - Idle. The main chat area displays a terminal window with the following content:

```
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.

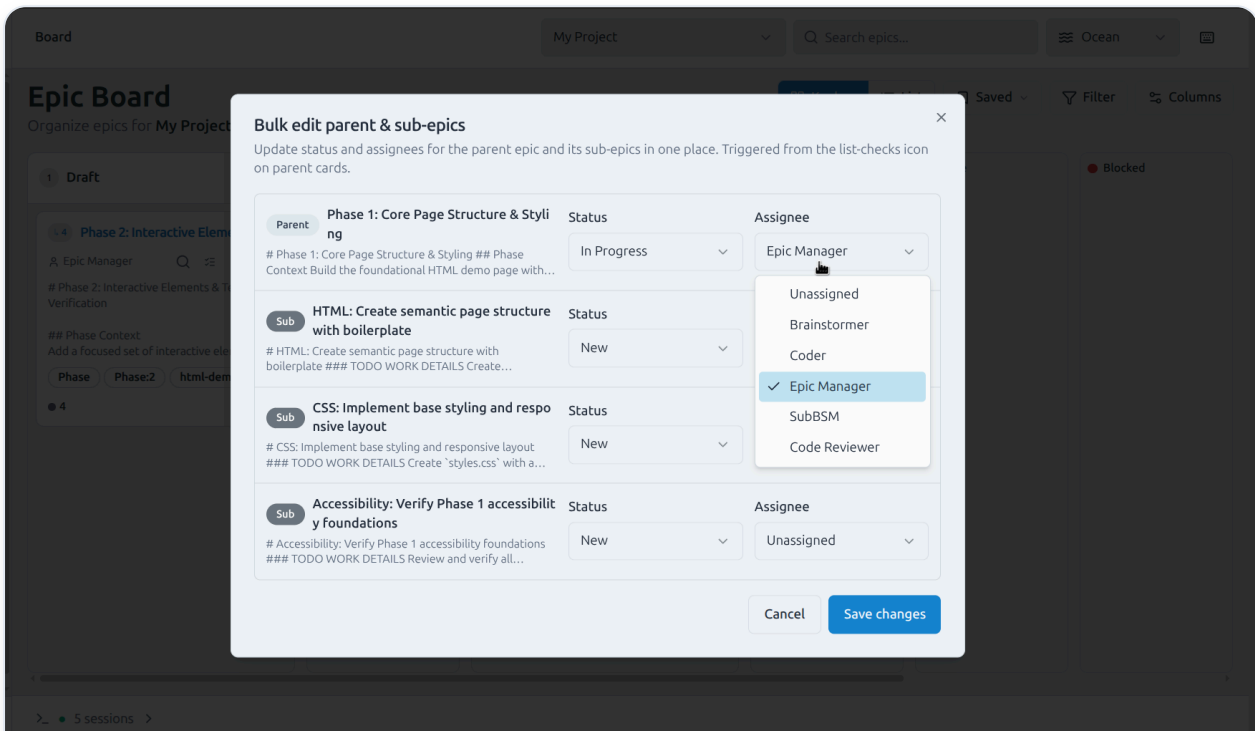
> let's create looks good to me!
>> bypass permissions on (shift+tab to cycle)
```

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

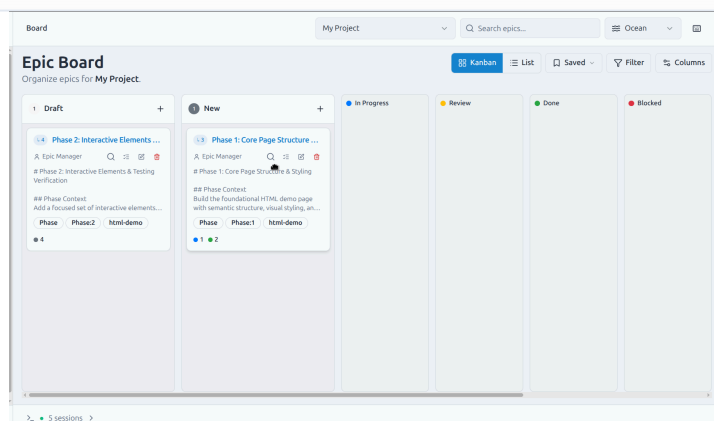
- i After approval, Brainstormer creates the documentation and decomposes the plan into Phase epics and sub-epic tasks on the Board. Head to the Board page to manage epics.

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** — not directly to Coder. Epic Manager will then coordinate task assignments automatically.



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

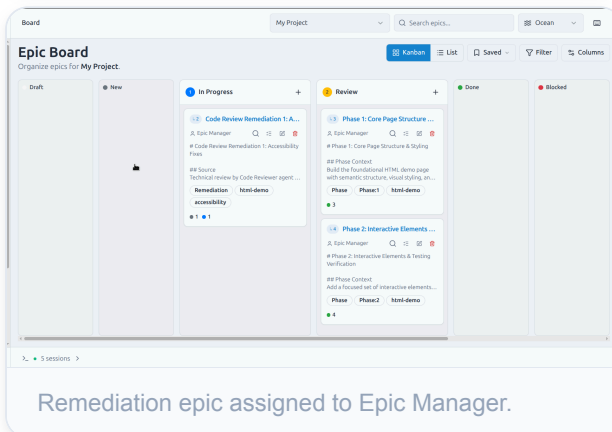
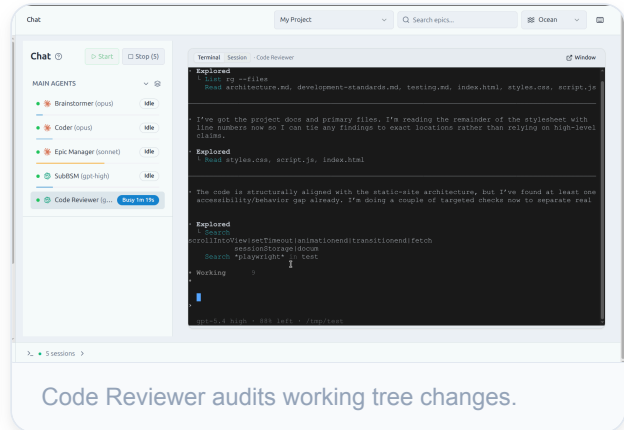
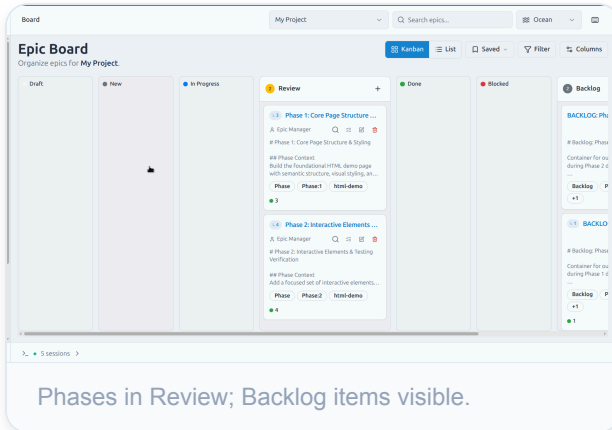


Kanban view — Phase 1 assigned to Epic Manager and moved to In Progress.

i Unlike the 3-agent flow where you assign directly to Coder, here **Epic Manager** handles task-level assignments, reviews, and Code Review requests.

10 Execution, Review & Completion

Epic Manager coordinates the full cycle: assigning tasks to Coder, reviewing completed work, and requesting Code Review when all phase tasks are done.



- 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 coordinates the fix with Coder and triggers another Code Review.
- From the **Review page** you can check code changes and send comments to any agent.
- Once happy, **ask Coder to commit** or do it yourself! Repeat the flow for new plans and features.

! Note: 3-Agent Development Flow

This guide and its screenshots describe the **5-agent development flow**. Devchain also offers a **3-agent flow** that is simpler, more token-efficient, and well-suited for smaller projects or basic subscriptions.

3 Agents = 5-Agent Flow – 2 Agents

No SubBSM

Plans go directly from Brainstormer to the user without an intermediate technical validation step.

No Epic Manager

You manually manage epics from the Board — dragging, dropping, and assigning Phase epics directly to Coder.

Key Differences from the 5-Agent Flow

1 No plan validation step

- › Brainstormer presents the plan directly to you without SubBSM verification.
- › You are the sole reviewer of the plan before decomposition.

2 Manual epic management from the Board

- › In the 5-agent flow, Epic Manager automates task assignments and reviews. In the 3-agent flow, **you manage this manually**.
- › Drag and drop epics between columns, assign Phase epics directly to Coder.

3 Phase assignment goes to Coder

- › Instead of assigning Phase epics to Epic Manager, assign them directly to **Coder**.
- › Coder picks up tasks sequentially and sends completed work to Code Reviewer.

i The 3-agent flow is more token-efficient and gives you more direct control over the process. Choose it for smaller projects or when you want hands-on epic management.