QUICK START GUIDE

# Devchain

AI-powered multi-agent development workflow

v1.0 — February 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:

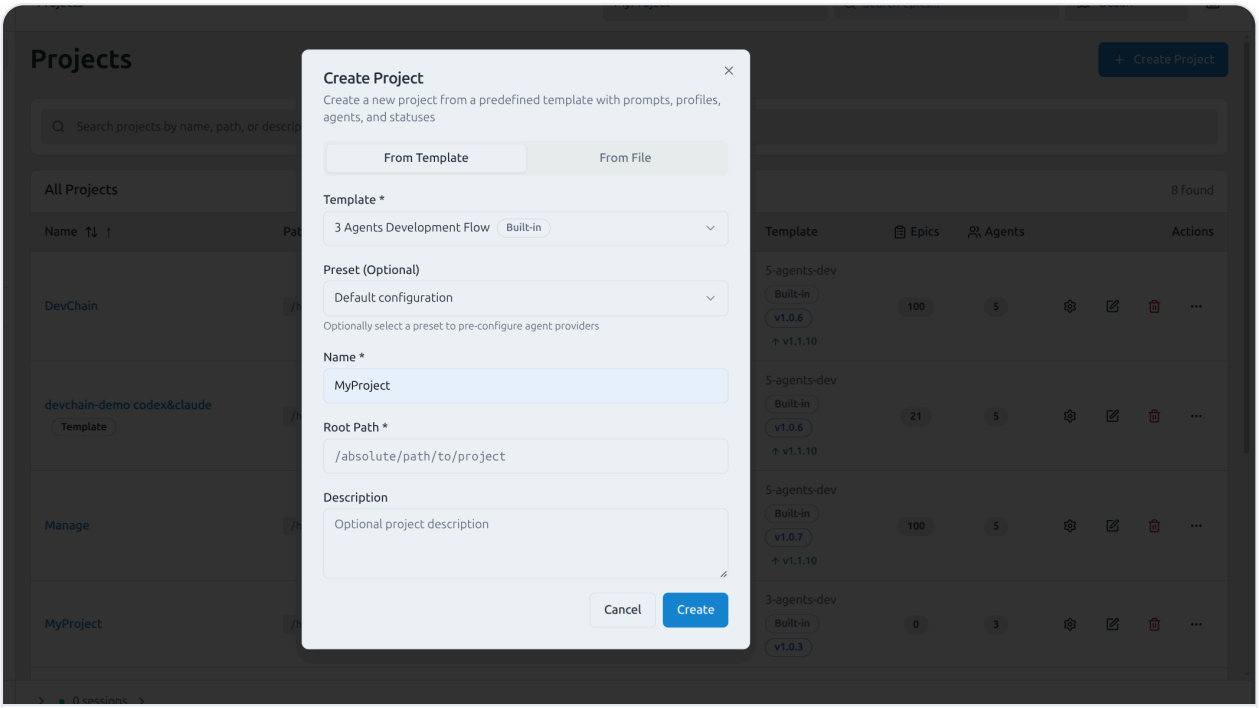
### 3-Agent Template

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

### 5-Agent Template

Best results for complex projects. Includes two planning agents and extra validation steps, but more token-intensive.

*Set the project name and path. Keep the template presets at their defaults.*




Create Project — select a template, set name & path, then click Create.

# 3 Development Flow — 3-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. First sessions can take a bit to build documentation for existing projects.
   - Discusses requirements with user and gets explicit master-plan approval.

2. **Brainstormer decomposes work**
   - Creates a Phase epic ( `Draft` ), a Backlog epic, then sub-epic tasks ( `New` ) with acceptance criteria, prereads, and relevant skills.

3. **Coder executes tasks sequentially**
   - Picks the next lowest `Task:N` item, sets it to `In Progress` , implements only scoped work, adds tests/docs, runs quality checks.

4. **Coder documents evidence**
   - Posts structured completion notes (files changed, tests, verification), then moves task to `Review` .

5. **Coder loops until phase tasks are exhausted**
   - Repeats steps 3–4 for the next `New` task.
   - When no `New` tasks remain, assigns parent epic to Code Reviewer and sets parent to `Review` .

6. **Code Reviewer audits working tree changes**
   - Reviews uncommitted code ( `git status` , `git diff` , `git diff --cached` ) for architecture, DI, errors, security, performance, and code quality.

7. **Review outcome**

> **APPROVED:** Reviewer comments and moves reviewed epics to `Done` (user commits manually).
>
> **ISSUES FOUND:** Reviewer sends a remediation plan to Brainstormer; epic stays in `Review` .

**Remediation cycle (if needed):** Brainstormer creates a new remediation parent epic and decomposes it into new tasks; Coder and Reviewer repeat the loop.

# 4 Chat Page & Provider Configuration

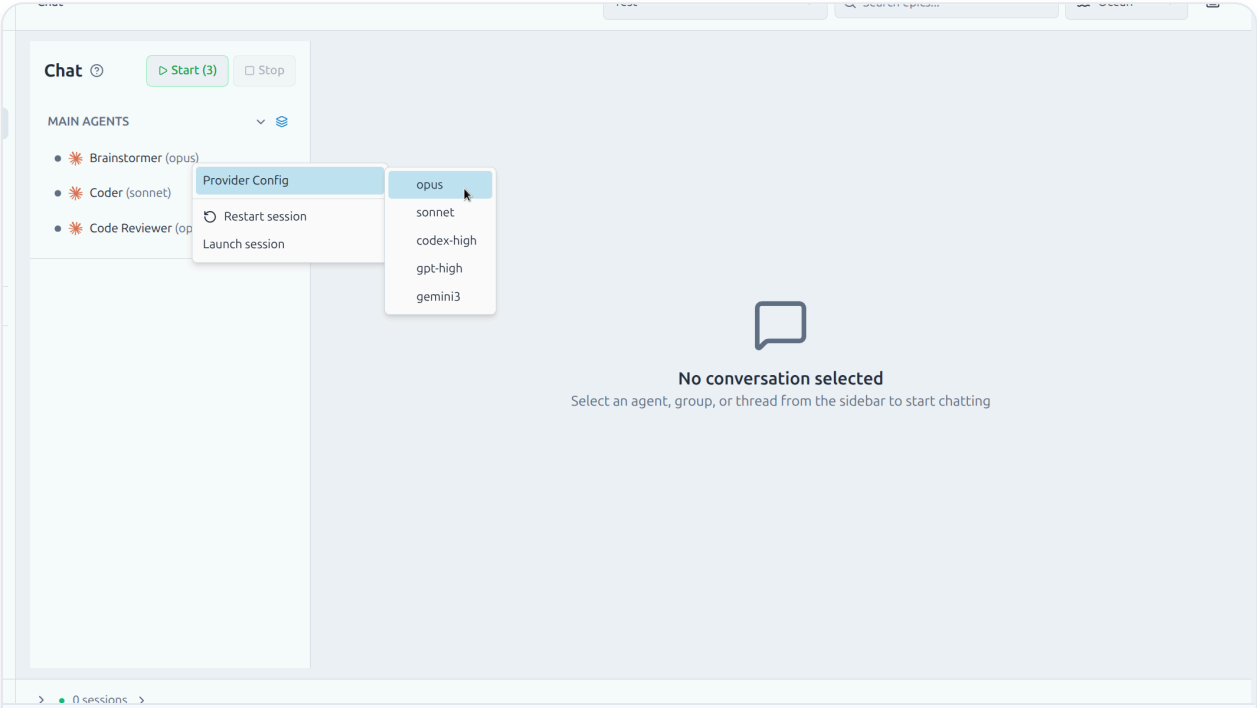
The Chat page shows your agents and their current AI provider assignments.

### Claude + Codex subscriptions

Leave the default setup as-is: **Claude Opus** for planning (Brainstormer), **Sonnet** for implementation (Coder), **Codex 5.3 High** for Code Review.

### Claude-only subscription

Right-click on the Code Reviewer agent and open **Provider Config** to change its provider from Codex to **Opus**.

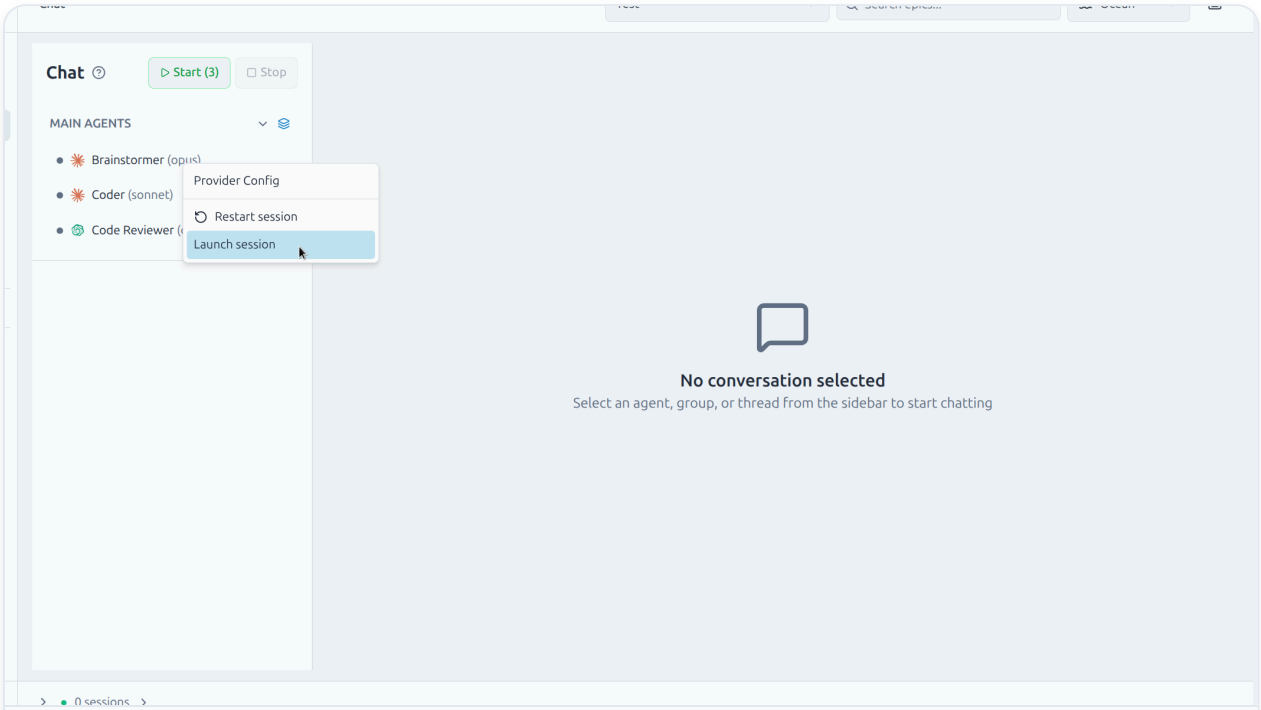



Right-click an agent to access Provider Config and select a different AI model.

# 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.

> Starting only Brainstormer first is a good way to discuss and refine your plan before bringing in the other agents.

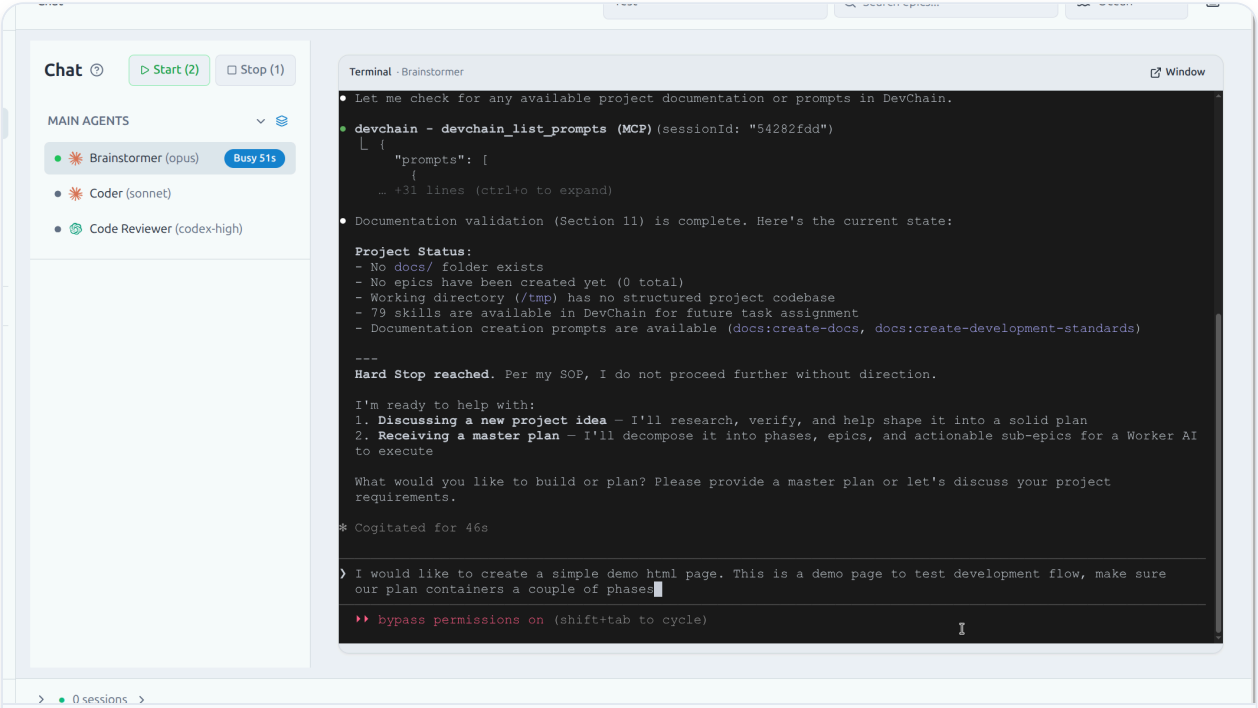



Right-click an agent and choose Launch session to start it individually.

# 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.

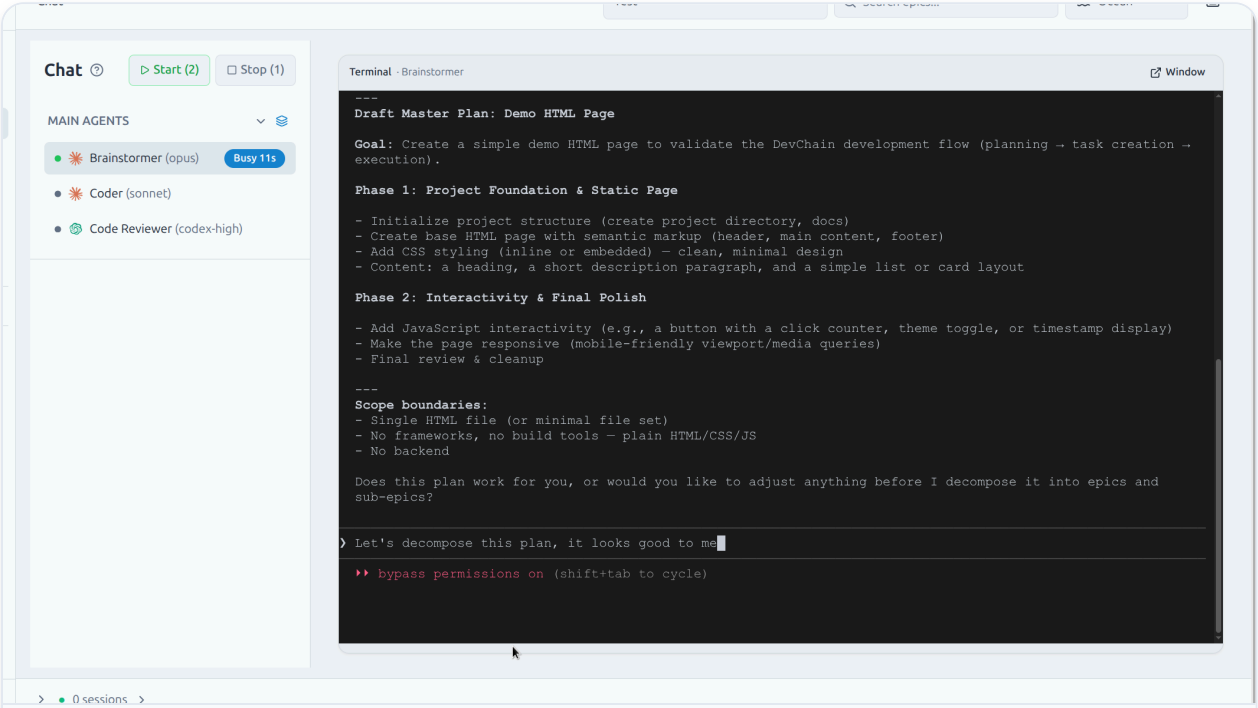



Brainstormer validates documentation, then asks what you would like to build.

# 7 Decompose the Plan

Once the master plan is ready and approved, ask Brainstormer to decompose it into actionable epics and tasks.

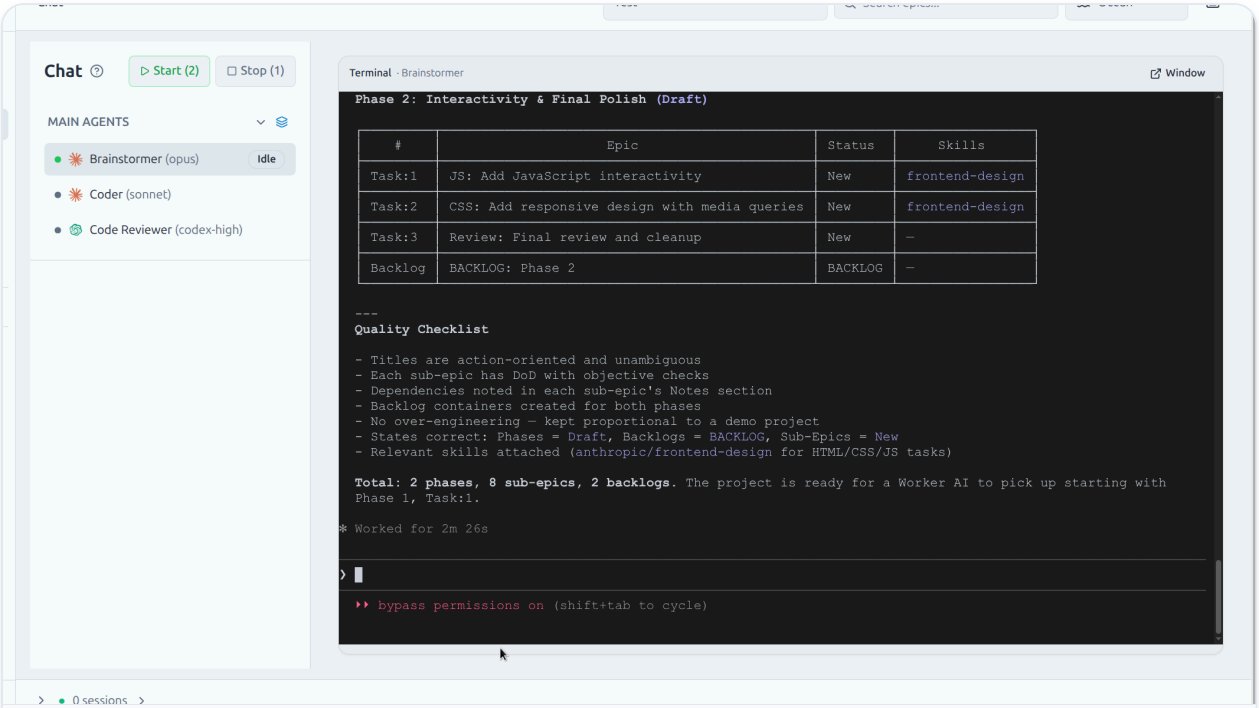
→ Tell Brainstormer: **"Let's decompose this plan"** to break it into phases, epics, and sub-tasks.




Brainstormer presents the master plan; user confirms to decompose it into epics.

# 8 Plan Decomposed into Phases

The plan is decomposed into **Phase epics** (parent epics) and **sub-epics** (individual tasks). Each sub-epic includes acceptance criteria, prereads, and relevant skills.

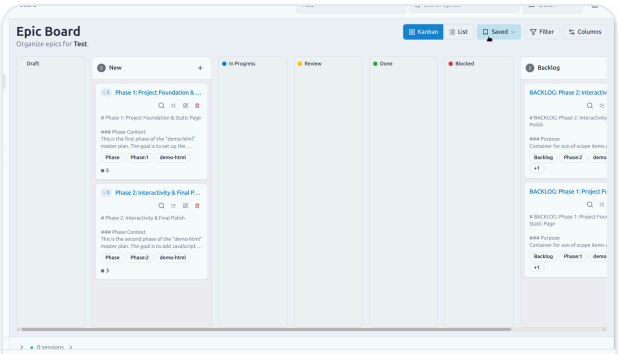



Brainstormer outputs a summary table of phases, tasks, statuses, and skills.

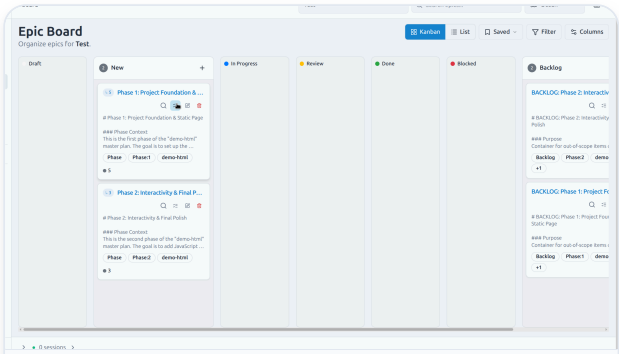
> **i** The project is now ready for a Coder agent to pick up tasks. Head to the Board page to manage epics.

# 9 Board — Browse & Assign Epics

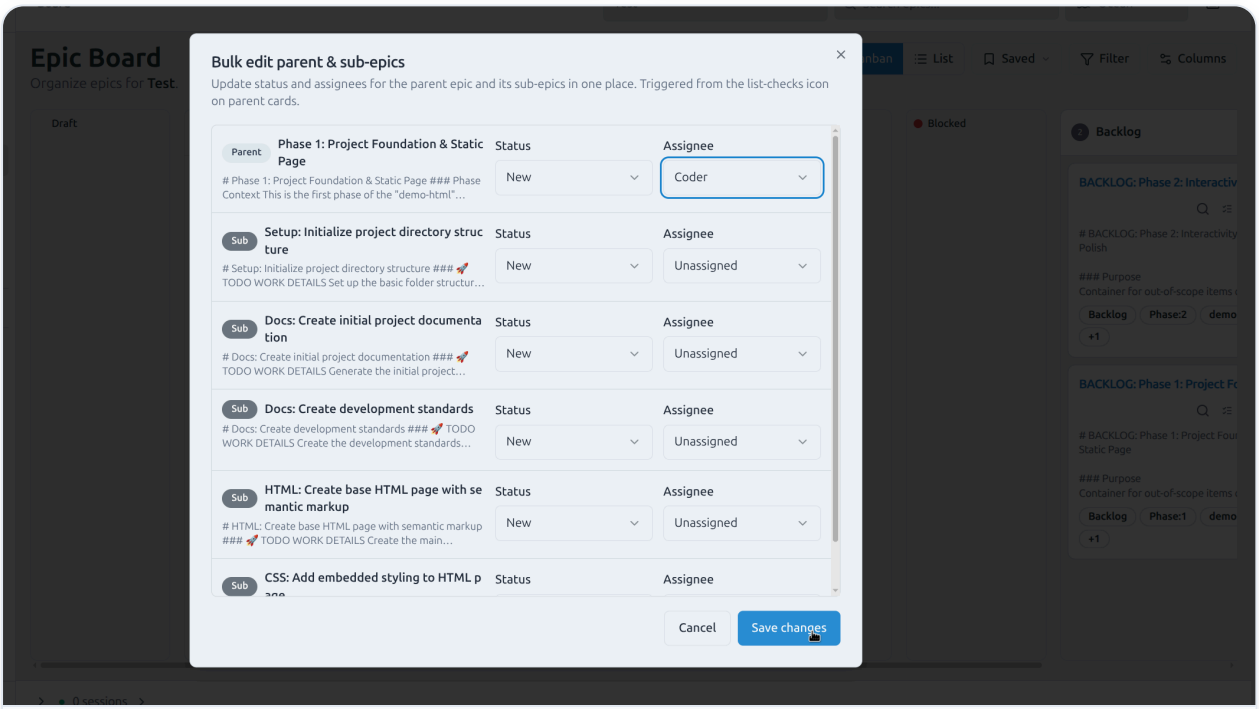
From the **Board page** you can browse created epics in a Kanban view. Drag and drop epics that are ready for implementation from `Draft` to `New` status, then assign the **Coder** agent onto the first Phase.




Board Kanban view with phases in Draft.



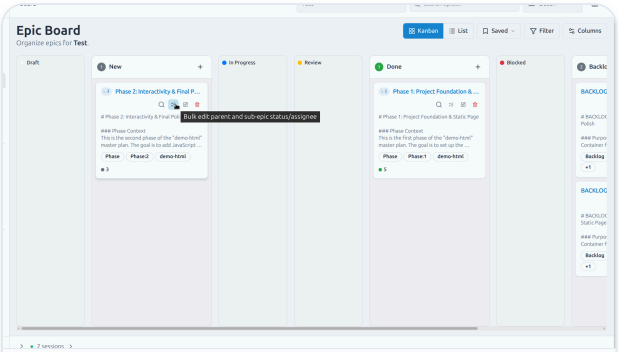

Hover over an epic to see action icons.



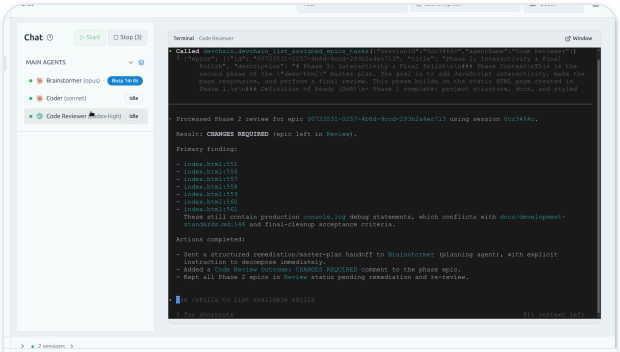

Bulk edit — set status to New and assign the Coder agent to the parent Phase epic.

# 10 Execution, Review & Completion

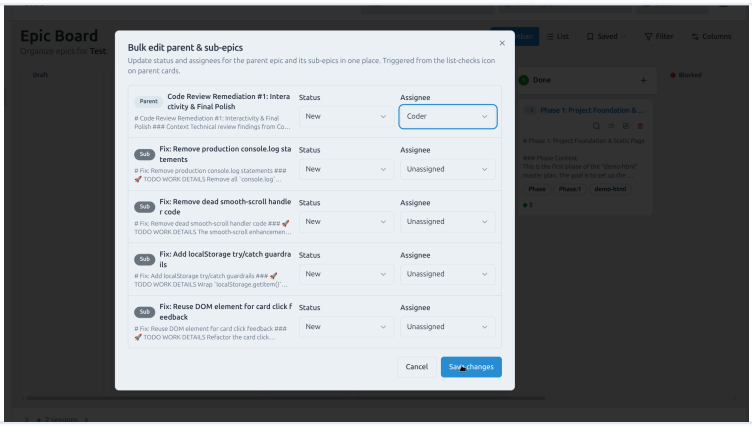
Coder completes all sub-epics and sends work for Code Review. From the Chat page you can watch the coding and reviewing process in real-time.




Phase 1 moved to Done; Phase 2 ready in New.




Code Reviewer finds issues and sends remediation.




Move remediation epics to New and assign onto Coder.

- Code Reviewer **approves** and moves the epic to Done — or sends review details to Brainstormer to create a **remediation epic**.
- Move remediation epics to `New` and assign onto Coder to complete.
- You can be involved in the review process — discuss changes for remediation epics with Brainstormer at any step.
- From the **Review page** you can check code changes and send comments to any agent with additional questions.
- Once happy, **ask Coder to commit** or do it yourself!
- Repeat the flow for new plans, features, and changes.

# ! Note: 5-Agent Development Flow

This guide and its screenshots describe the **3-agent development flow**. Devchain also offers a **5-agent flow** that adds two additional agents for enhanced planning validation and automated epic management.

### 5 Agents = 3-Agent Flow + 2 Extra Agents

**SubBSM** (Sub-Architect)

Technical validator that reviews Brainstormer's draft plans against the actual codebase before they are presented to the user.

**Epic Manager** (Reviewer)

Manages the full epic lifecycle: reviews completed tasks, approves or sends back for revision, assigns next tasks to Coder, and requests code review when all work is done.

## Key Differences from the 3-Agent Flow

### 1 Plan validation with SubBSM

- Before presenting a plan to the user, Brainstormer sends the draft to SubBSM for technical validation.
- SubBSM checks the plan against the actual codebase — verifying file paths, identifying over-engineering, checking dependency conflicts, and validating completeness.
- This validation loop can run up to 10 rounds before the final plan reaches the user.

### 2 Epic Manager controls the execution flow

- In the 3-agent flow, you manually manage epics from the Board (drag, drop, assign). In the 5-agent flow, **Epic Manager automates this**.
- Epic Manager reviews each completed sub-epic, approves or sends it back for revision, and assigns the next task to Coder.
- When all tasks in a phase are done, Epic Manager requests code review automatically.

### 3 Epic assignment goes to Epic Manager

- Instead of assigning Phase epics directly to Coder, assign them to **Epic Manager**.
- Epic Manager will then coordinate task assignments to Coder and manage the review cycle.

i The 5-agent flow produces higher-quality results through additional validation layers, but consumes more tokens. Choose based on your project complexity and subscription tier.