

Why we should pilot AI CodeOps

AI CodeOps (formerly CodeOps Studio) is a browser-based platform where AI agents plan, build, review, test and ship code, and nothing merges until quality gates and a human approve it. It works with GitHub, GitLab, Bitbucket, Gitea and Forgejo, is free during its open beta, and is operated by TechInnoHub.

What it changes

- **You want AI to ship changes, not only suggest them.** AI CodeOps agents plan the change, write it against your repository, run your tests in a sandbox, review the diff and open the pull request. You review a finished change, not a suggestion.
- **Nothing should merge on the AI's word.** Every AI change passes a release gate: tests that ran on that version, a review with no open high or critical findings, and a person's approval. The person who asked for the change can't approve it alone when someone else can.
- **You need to show how code shipped.** Each change keeps a signed evidence pack: the task, models, approvals, findings, tests and audit trail, which anyone can verify without an account.
- **Your code isn't only on GitHub.** GitHub, GitLab, Bitbucket, Gitea and Forgejo, including your own servers, all work the same way.
- **You want spend under control.** Every session shows what it cost, and budgets stop a session before it passes the limit you set.

A two-week pilot

1. **Day 1:** Sign up, connect your Git host and create a project for one real repository. Invite one teammate so approvals need a second person.
2. **Days 2–3:** Run three or four small, real tasks you'd otherwise give to your current tool: a bug fix, a test, a small feature. Keep your current tool running.
3. **Days 4–7:** Review each change at the gate. Note what the review found, whether the tests ran, and how long approval took. Set a workspace budget.
4. **Week 2:** Move one team's routine changes to AI CodeOps. Compare cost per merged change, rework, and review time with your current tool.
5. **Day 14:** Decide with the numbers. Keep both tools, switch, or stop: your pull requests and repositories are unchanged either way.

Moving from our current tool

From GitHub Copilot: Keep Copilot in your editor: AI CodeOps works on the repository, not in the IDE, so the two don't overlap. Connect GitHub in AI CodeOps (Settings → Git Providers) and create a project for the repository. Give AI CodeOps the tasks you'd assign to an agent, and let the release gate decide what reaches a pull request.

From CodeRabbit: Leave CodeRabbit installed during the pilot: AI CodeOps opens normal pull requests, so it reviews them like any other. Connect your Git host in AI CodeOps and create a project. Once AI CodeOps' own review and gate cover what you need, decide whether to keep a second reviewer on human pull requests.

From Devin: Pick the kind of task you give Devin today and run the same task in AI CodeOps. Compare the pull request, the tests that ran, the review findings and the cost per session. Move those tasks over one team at a time; nothing in your repository is tied to either tool.

Questions the team will ask

What does switching cost during the beta? Nothing. AI CodeOps is free during the open beta, with a monthly AI allowance on the platform's models, and you can use your own model provider key.

Do we have to stop using our current tool? No. AI CodeOps opens ordinary pull requests on your Git host, so it runs alongside other tools for as long as you like.

Is our code used to train AI models? AI CodeOps doesn't use your code, prompts or outputs to train AI models. Model providers process them under their own terms.

What if we leave? Your repositories and merged pull requests stay where they are. You can export your account data as JSON and download each change's evidence pack before you go.

Can we run it on our own infrastructure? Self-hosted deployment is available to Enterprise customers on request. Everyone else uses the hosted service in the browser.

aicodeops.dev/switch · Free during the open beta · Runs in the browser, nothing to install