

Builder and Your AI Coding Tools

Your engineers keep the AI coding tools they already use, and Builder wraps them in a collaboration layer that lets the whole team build, review, and ship together.

Where delivery time actually goes

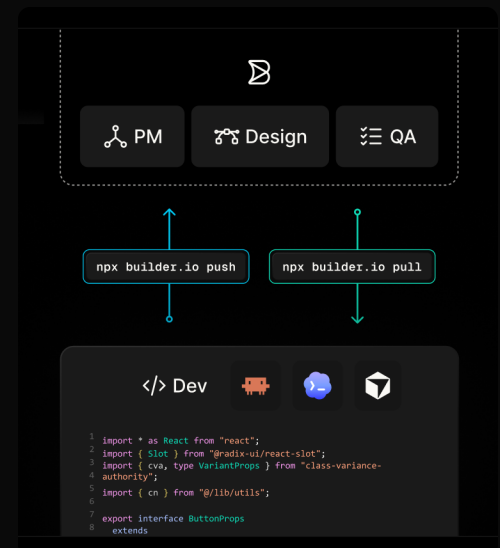
AI coding tools like Claude Code, Cursor, and Copilot speed up how fast an engineer writes code, but coding is only a small share of the work.

Along the full path from idea to production, the product spends most of its time waiting in a queue for the next person to pick it up, moving between PM, design, engineering, and QA. Every handoff is another wait, and those waits are where the weeks go. Speeding up one person's step does little when the delay lives in the space between steps, which is exactly the space a team shares.

The team layer

AI coding tools are built around a single engineer working through code one task at a time. Builder gives the whole team that same kind of agent power in a place built for many people at once: multiplayer, in the cloud, connected to Jira, GitHub, Slack, and everyone who touches the product.

Your engineers keep working with the AI coding tools they know. They can code with Builder's agent or their own tool and move a branch between the two freely, so each person can work where they're fastest. Builder brings the same capability to product, design, and marketing, adds its own review and QA agents, and enforces the governance you need to ship safely. Every role works from the same code in one place, so a change can move all the way to a reviewed PR without being handed off between queues.



Building in Builder alongside your AI coding tools

AI coding tools speed up how one engineer writes code. Builder adds the team layer, where everyone works together in a shared environment, so changes move to a reviewed PR without waiting in queues between roles.

	YOUR AI CODING TOOLS	WITH BUILDER
The center of work	Fast individual coding, one engineer at a time	A shared environment that the whole team works in
Who can use it	Engineers, in their editor	Product, design, engineering, and QA, together
Where it runs	Local or cloud, spun up and tracked by one engineer	Cloud environments per branch, shared across the team
Parallelism	Parallel agents, dispatched and tracked by one engineer	Builder's coding agents run in parallel, visible and assignable across the whole team in a Kanban view
Preview	The engineer runs or opens the preview themselves	Full dev environment per branch, front-end and back-end, ready for anyone to preview
Visual editing	Point-and-edit against whatever renders in the tool's browser	Direct editing against your real design system and tokens
Review	The PR is reviewed after the work is done	Designers, PMs, and QA test and refine before the PR exists
QA	Verification set up and read by one engineer	Builder's review and QA agents run end-to-end checks on every PR
Handoffs	Each role translates work to the next	The whole team works from one shared environment

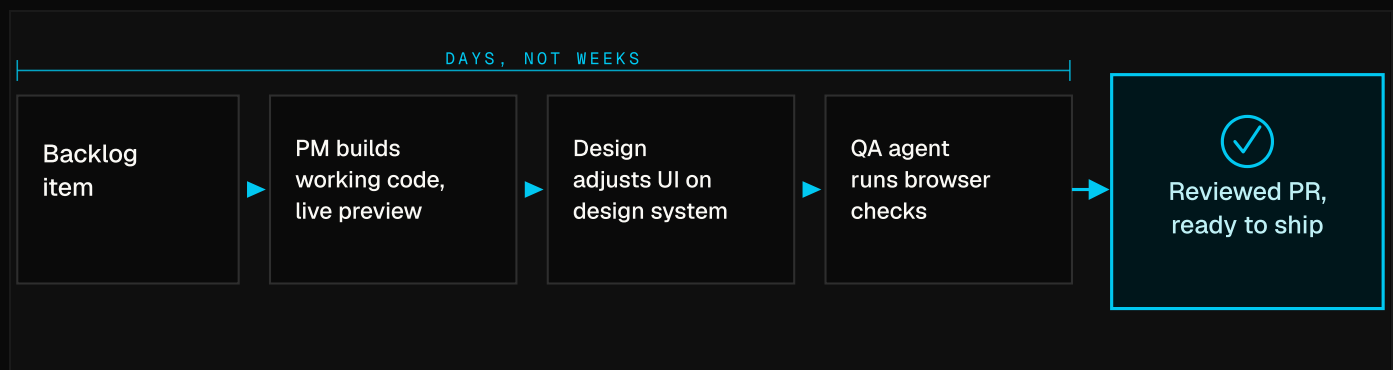
A parallel workflow

Most companies say they run Agile, and in practice, run waterfall, moving work through a long, multi-step process where most of the time is spent waiting in queues.

In Builder, everyone on the team works on the same codebase and can assign a coding agent to it, which changes how the process runs.

Take a single backlog item. A PM turns it into working product code and tries it in a live preview the same afternoon. A designer opens the same branch and adjusts spacing and states against the real design system. Builder's QA agent runs the change in the browser to confirm nothing else broke, and an engineer reviews the results and merges.

By the time the PR lands, the whole team has built, reviewed, and verified it together, so it arrives ready to ship. Work that used to take weeks in queues moves in days, and the team gets to ship early and learn from real feedback while there is still time to act on it.



Opening the codebase to more of the team does not change how code gets approved. Every change, regardless of who starts it, becomes a normal pull request and runs through the same review, tests, and quality gates you already enforce. Builder's review and QA agents check each PR and flag issues early, and an engineer still approves the merge.



Security and deployment

Run the entire system in your own cloud, behind your firewall, with your repos, your permissions, and your governance. Builder is SOC 2 compliant and builds and reviews inside your existing workflows, on your own infrastructure.

For every role on the team

Because the whole team works on the same codebase, each role picks up work that used to sit in someone else's queue. Here's what changes day to day with Builder:

ENGINEERING

Engineers no longer make every routine change themselves. A designer or PM can handle layout, color, and copy fixes in Builder, or an engineer can assign them to an agent. Engineers code with the agent they choose and move branches between it and Builder freely. Builder handles branch setup, the dev environment, and PR mechanics, so every PR arrives reviewed with agents checking safety and compliance.

PRODUCT

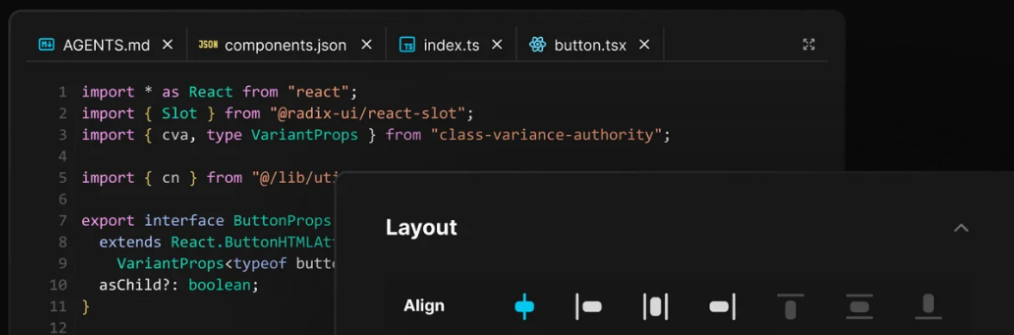
Product builds ideas into working drafts in the real codebase and tries them in a live environment to see how they actually behave. A PM can put a real version in front of stakeholders, gather feedback on something people can click through, and refine it in the same place, without writing a spec that has to survive the handoff to engineering.

DESIGN

Design works on the real product, not a picture of it. Instead of red-lining a mockup, a designer adjusts spacing, states, and motion directly against the production design system, and those changes become the code that ships. The version customers see is the one the designer approved.

MARKETING

Marketing builds and ships landing pages and campaign updates against real product components. A marketer can edit copy, adjust layouts, and stand up new pages directly, then send them through the same review and approval path as any other change, launching on marketing's own timeline without waiting on an engineering cycle.



Builder puts the whole team on one codebase alongside the tools you already run, so work stops waiting between roles.

[SCHEDULE A DEMO →](#)[TRY BUILDER FREE →](#)