builder.io

From Idea to Production:
An Engineering Leader's Guide to AI-Native Development with Builder

A step-by-step workflow using Builder, from design system index to merged PR

Your team adopted AI coding tools, and developers are moving faster, but delivery timelines haven't changed.

The teams shipping significantly faster aren't using better AI. They've changed how the work moves. Every role contributes directly. Engineers stop being the middleware for decisions that don't require engineering expertise. Code gets reviewed before it reaches engineering, not after.
This guide walks through that workflow step by step, using Builder as the implementation layer. Each step maps to a concrete action your team can take.

- WHAT THIS GUIDE COVERS: How to run an AI-native development workflow, from indexing your design system to merging a PR, with every role participating directly.
- WHO IT'S FOR: Engineering leaders evaluating or implementing AI-native development workflows.
- WHAT YOU NEED: An existing codebase, a component library or design system, and a product requirements document for the feature you want to build.
- WATCH THIS WORKSHOP to see a real idea go from blank prompt to a production-ready component in 60 minutes, no matter your role on the team.

Ready to see the workflow in action? See the walkthrough session

builder.io

HOW THE WORKFLOW RUNS

1 INDEX YOUR DESIGN SYSTEM

Eng lead / DS lead — *1–2 hours, one-time setup*

Run CLI at repo root. Connect codebase and Figma plugin. Confirm component library is indexed.

2 KICK OFF A BRANCH

PM or eng lead — *15–30 min*

Create branch. Attach PRD. Write directional prompt. Answer agent questions. Review initial build via preview URL.

3 OPEN TO EVERY ROLE

PM, designer, QA — *Ongoing*

Share preview URL. Assign roles and permissions. PM in interact mode, designer in style mode, QA validates live. Encode standing rules.

4 REQUIRE APPROVAL BEFORE PR

Eng lead, design lead, PM — *30–60 min*

Configure required approvals. Each role validates against explicit criteria. Agent addresses feedback on the branch. PR opens after all sign-offs.

5 MERGE AND MEASURE

Eng lead — *Ongoing*

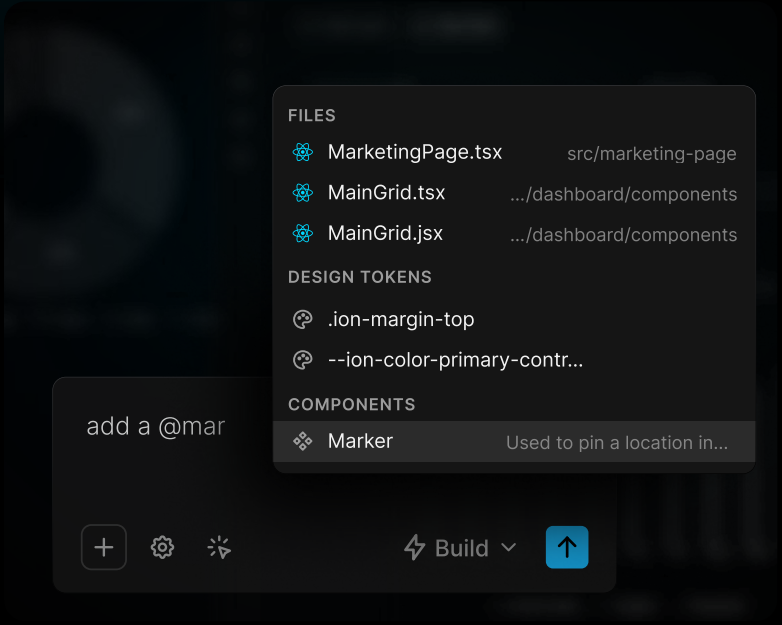
Review code quality. Merge. Track implementation time, rework cycles, adherence, capacity freed.

Ready to see the workflow in action? See the walkthrough session

builder.io

Step 1: Index your design system

Engineering lead or design systems lead
1–2 hours, one-time setup



Generic AI output is the single biggest reason AI coding tools fail in production. The agent generates code that your engineers spend days rewriting because it doesn't know your components or follow your conventions. Better prompting helps, but the real fix is context.

Builder needs to know your codebase, your design system, and your component library before any code is generated. Run a CLI at the root of your repo. The indexer generates documentation about your design system (what components exist, how they're structured, what tokens govern them) and feeds it directly to the agent. Every subsequent prompt generates code that uses your actual components, follows your patterns, and passes your linting rules.

An incomplete or outdated component library caps what AI can produce. If your design system has significant drift between Figma and code, resolve that before indexing.

HOW TO DO IT:

1. Connect your target codebase to Builder via GitHub, GitLab, Bitbucket, or Azure Repos.
2. Run the Builder CLI at the root of your repo to generate the design system index.
3. Add reference codebases (e.g., a Storybook project) to give the agent additional component context.
4. Connect your Figma plugin and map Figma components to their coded equivalents.
5. Confirm the index in project settings. Your component library should be listed and indexed.

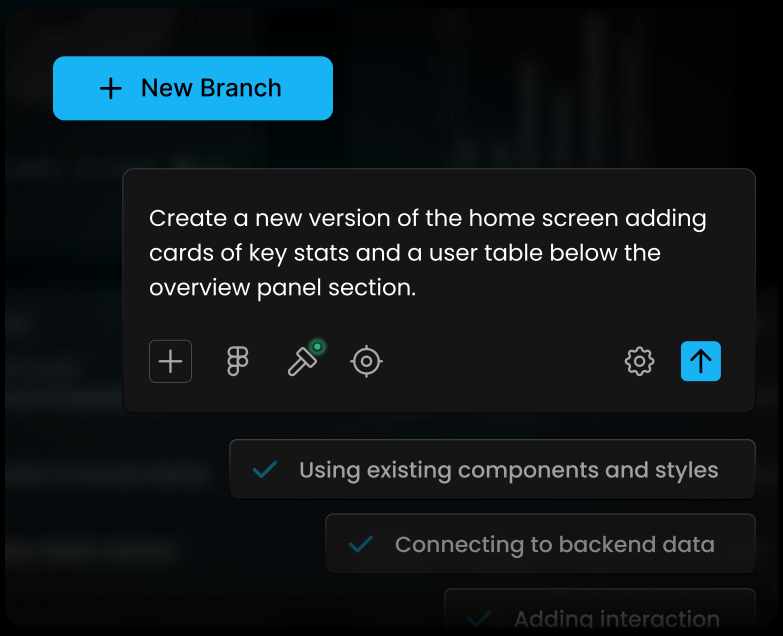
WHAT SUCCESS LOOKS LIKE:
When you prompt the agent to build new UI, it references existing components by name and imports from your component library rather than generating custom code.

Ready to see the workflow in action? See the walkthrough session

builder.io

Step 2: Kick off a branch with real context

PM or engineering lead
15–30 minutes



The starting point for any feature is a branch. Use your spec as direct input to the agent and let it generate a first pass. You're no longer debating what something might look like. You're reacting to something that exists. Feedback becomes concrete immediately.

Keep your initial prompt directional. The agent will ask clarifying questions when direction is unclear, surfacing ambiguity in seconds rather than in alignment meetings. Each agent runs in its own isolated container with a full dev environment, so every branch is immediately previewable.

HOW TO DO IT:

1. Create a new branch from your connected project in Builder.
2. Attach your PRD or requirements document using the file upload in the prompt bar.
3. Write a directional prompt describing what you want to build.
4. Answer any clarifying questions from the agent.
5. Review the initial build via the branch preview URL.

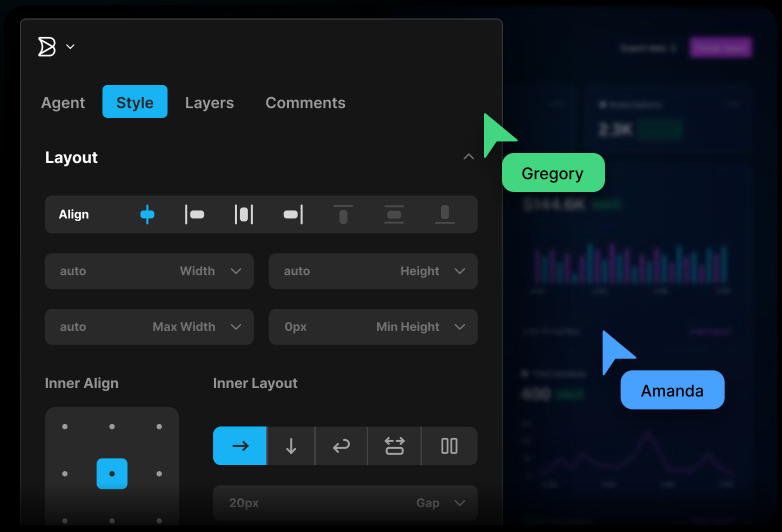
WHAT SUCCESS LOOKS LIKE:
A working, previewable branch that uses your real components and follows your conventions. A starting point the whole team can react to, not a finished product.

Ready to see the workflow in action? See the walkthrough session

builder.io

Step 3: Open the branch to every role

PM, designer, QA
Ongoing during iteration



Once the branch exists, every role contributes directly. Designers select elements in the visual editor and prompt the agent to adjust spacing, color, or component choice. PMs make copy changes themselves. QA flags issues directly and verifies fixes on the spot. No tickets. No handoffs. No waiting.

Explicitly set the expectation: design, product, and QA run concurrently on the branch. Engineering reviews the result, not each individual change. Use the agent's rule system to encode standing instructions (design system adherence, accessibility requirements, testing patterns) so the team doesn't repeat them in every prompt.

INTERACT MODE:	Natural language prompts for new features, functional changes, or behaviour updates. Best for PMs and engineers.
STYLE MODE:	Select any element and edit visual properties directly. Best for designers refining layout, spacing, color, or component usage.
PLAN MODE:	Collaborate with the agent on strategy before generating code. Best for scoping larger additions.

HOW TO DO IT:

1. Share the branch preview URL with your PM, designer, and QA. No login or local environment required.
2. Assign roles and permissions so each person has the access level appropriate for their contribution.
3. Let each role work in their mode. Track progress via the real-time Kanban view.
4. Encode standing instructions via the agent's rule system.

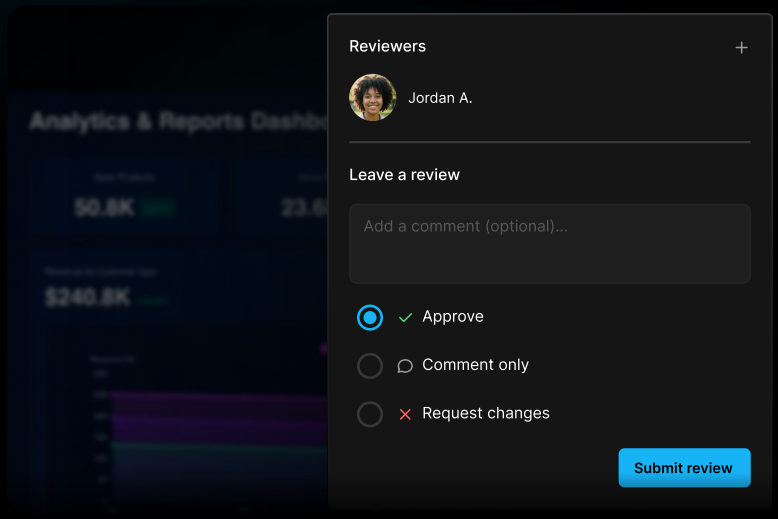
WHAT SUCCESS LOOKS LIKE:
Design has refined the implementation. PM has validated the flow. QA has run through the happy path. All before engineering touches it.

Ready to see the workflow in action? See the walkthrough session

builder.io

Step 4: Require approval before the PR opens

Engineering lead, design lead, PM
30–60 minutes



Most teams do QA and design review after engineering builds something. By that point, feedback creates rework and the cycle restarts. In an AI-native workflow, the review happens before the PR opens.

Configure required approvals from design, product, and QA before generating a pull request. Each reviewer accesses the branch preview URL to test the feature in a live environment. Any feedback requiring a code change is addressed by the agent on the same branch. What reaches engineering has already been validated by everyone with an opinion about it. The engineer's job is to review code quality, not relitigate product decisions.

Set explicit criteria for each role. Design checks component adherence and visual accuracy. Product checks the flow against requirements. QA runs the happy path and at least one edge case.

HOW TO DO IT:

1. In Builder's project settings, configure required approvals before a PR can be opened.
2. Assign design, product, and QA as required reviewers.
3. Reviewers approve or leave feedback directly in the platform.
4. The agent addresses feedback on the same branch.
5. Once all approvals are in, the PR is generated with a full summary of changes, commits, and context.

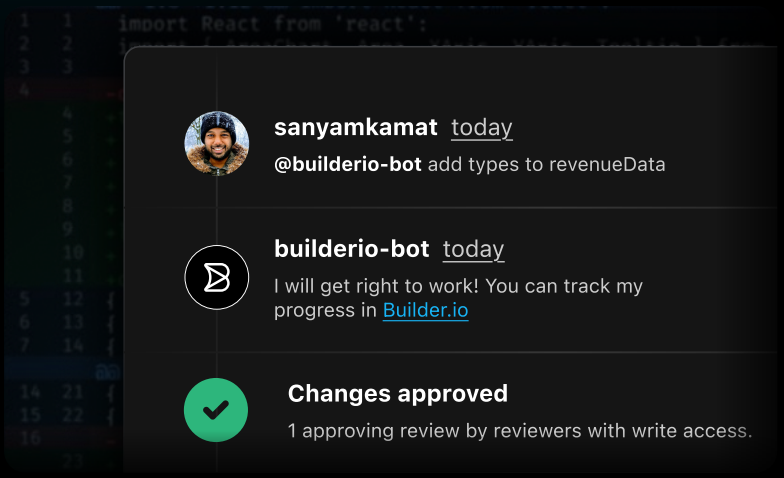
WHAT SUCCESS LOOKS LIKE:
Engineering receives a PR with a full change summary, UI screenshots, and confirmation that design, product, and QA have signed off.

Ready to see the workflow in action? See the walkthrough session

builder.io

Step 5: Merge and measure what changed

Engineering lead
Ongoing



The PR that reaches your engineers at this point is different from what they're used to receiving. It comes with context: what was built, why, who reviewed it, and what changed. The code uses real components and has already been validated against product and design requirements. Engineers review the diff, check code quality and architecture, then merge or send feedback to the agent.

After a few cycles, measure what's changed. Track delivery metrics, not AI adoption metrics.

TIME FOR UI IMPLEMENTATION

Hours per feature from approved design to merged PR. Should drop within the first 30 days.

REWORK CYCLES

Rounds of feedback after a PR opens. Target is zero. All feedback happens before the PR in this workflow.

DESIGN SYSTEM ADHERENCE

Percentage of merged UI code using indexed components. Should increase as the index matures.

ENGINEERING CAPACITY FREED

Sprint percentage previously spent on UI translation and design QA, now redirected to higher-value work.

WHAT SUCCESS LOOKS LIKE:
Within 30 days, measurable reduction in UI implementation time and rework cycles. Within 90 days, engineers report spending significantly less time on work that doesn't require engineering expertise.

Once the workflow is running on one team, add more teams to the same design system index. Extend the agent's standing rules to cover more of your engineering conventions. Open the platform to more non-engineering contributors as trust in the output grows.

Your developers are already faster. This is how you make the whole team faster. Try Builder for free or contact us for an enterprise trial.