

AI Build Lab · Product Engineering

AI Build Lab - Tools I Needed, Built and Shipped

Personally experienced problems become requirements, architecture, acceptance criteria, tested products, releases, and operations; AI amplifies implementation but does not own context or acceptance.

Period 2024 - present | Evidence state Ongoing | Portfolio tier AI Build Lab

One combined case connects a local-first knowledge system, a multi-CLI desktop application, and a Daegu bus-information application through the same problem-to-product loop.

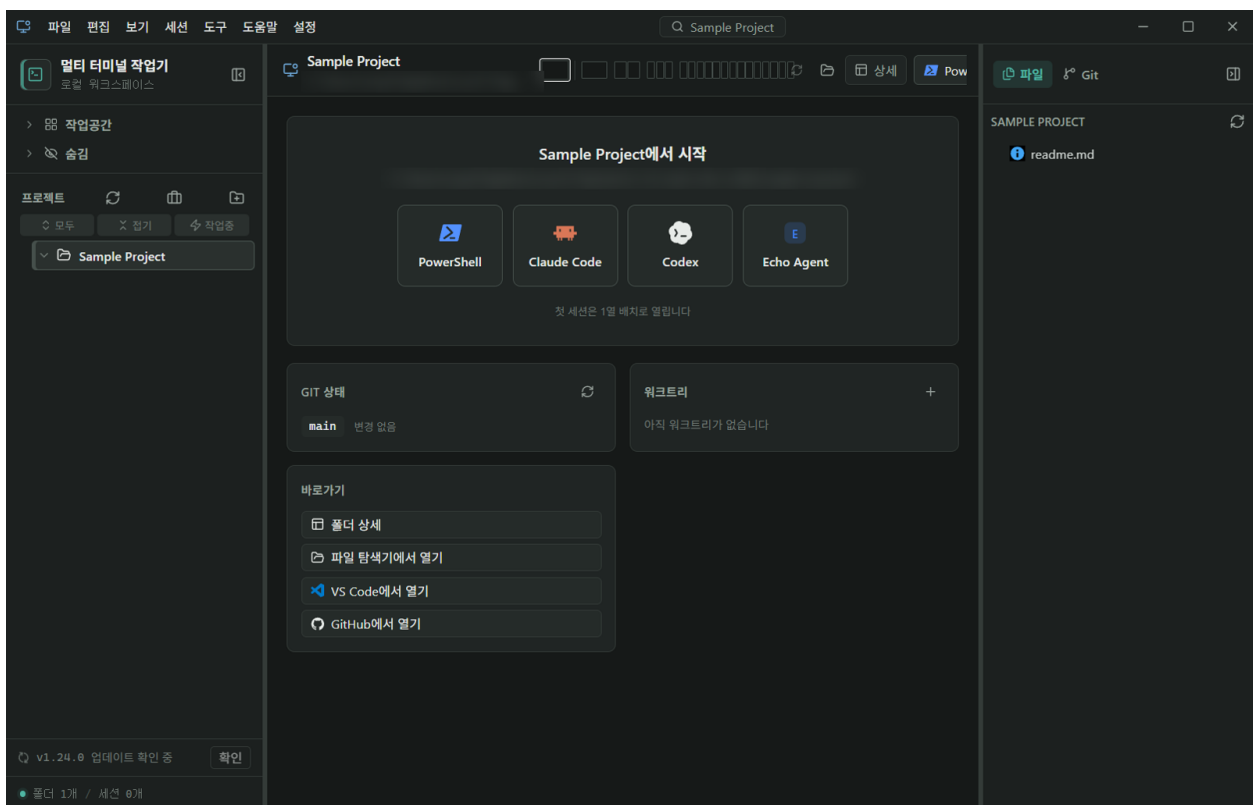


Figure 1. multi-cli-work v1.24: one project opens CLI and agent sessions alongside git status and worktrees. Local paths are blurred.

Problem and system boundary

Recurring friction needed to become reusable product requirements rather than one-off scripts.

Owned decisions and implementation

My role

Owned problem context, requirements, architecture, acceptance criteria, tests, releases, and operating decisions while using AI to amplify implementation.

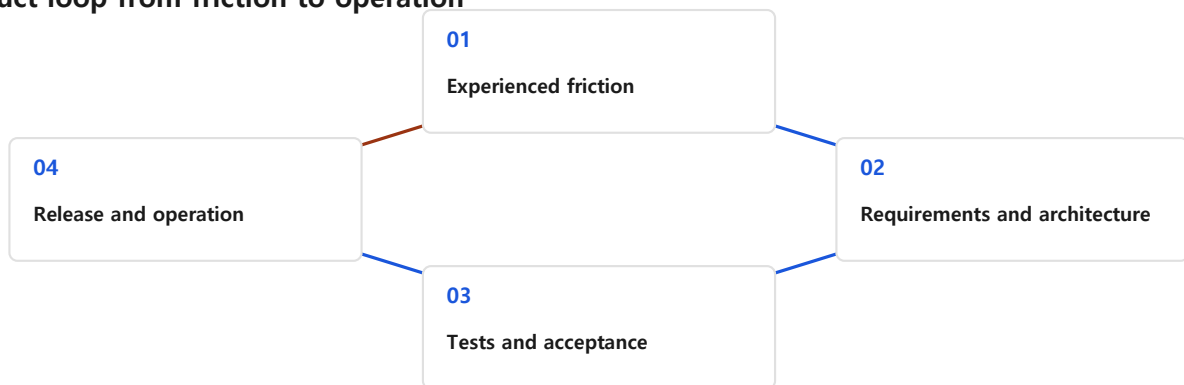
Problem to product

Connect friction to requirements, architecture, acceptance, tests, releases, and operations.

Human-AI boundary

AI amplifies implementation; people own context, acceptance, and release decisions.

Product loop from friction to operation



Explanatory system relationship diagram; it is not photographic or experimental evidence.

Verification and evidence ledger

Screens from a released multi-cli-work build, together with the public repositories, tests, and releases of both applications, provide public-safe evidence; private knowledge data is excluded.

Public product proof

Use public repositories, automated tests, and release artifacts as observable evidence.

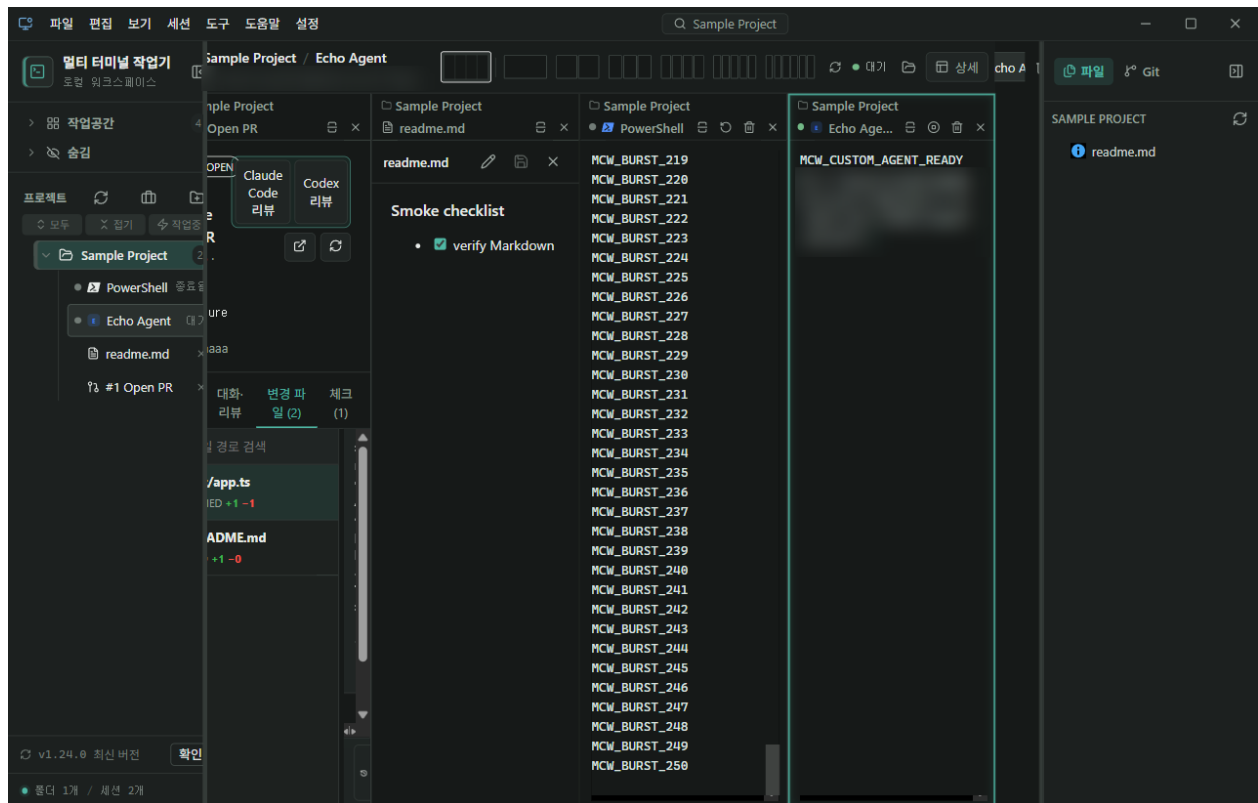


Figure 2. Four panes in one project: pull-request review, a document, and two terminals

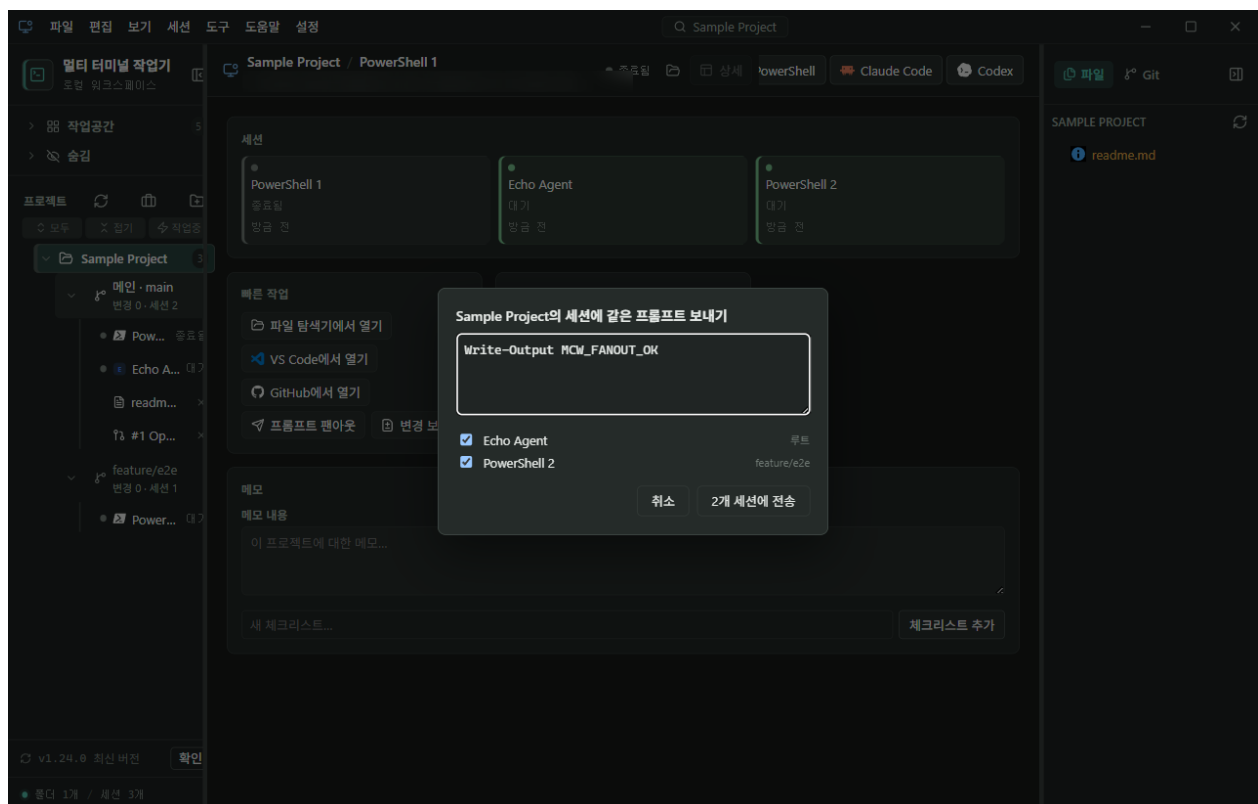


Figure 3. One prompt sent to several selected sessions at once

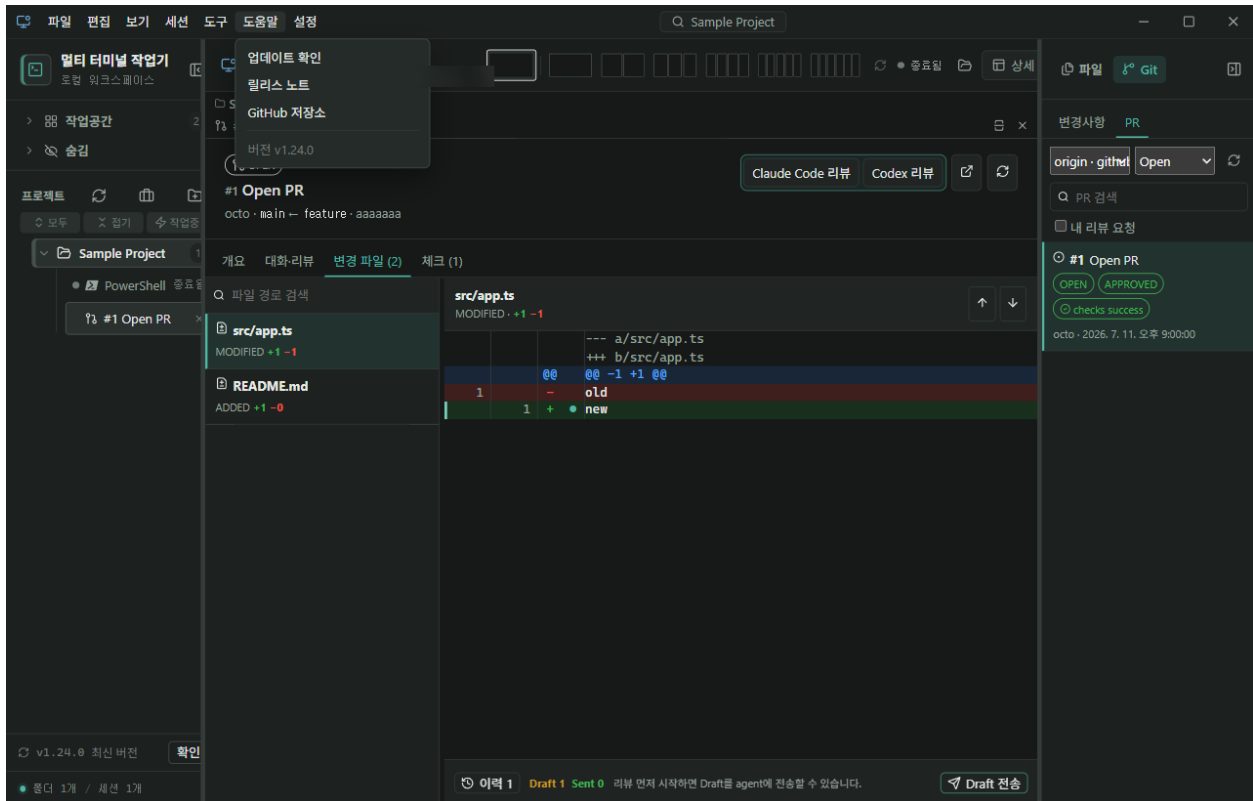


Figure 4. Changed files and diffs are read in the app, then handed to an agent review

Role, team result, and limits

Team result

Public repositories and release artifacts are observable product results; no user-count, productivity, or maintainability metrics are claimed.

Limitations

Never expose private knowledge-system source data or claim unverified user or productivity metrics.

Privacy and metric boundary

Exclude private knowledge data and unverified user or productivity metrics.

Collaboration boundary

AI amplifies implementation; people own context, architecture, acceptance criteria, review, and release decisions.

Implementation technologies

Electron · TypeScript · Node.js · Cloudflare · GitHub Actions · Agent workflows

Primary capability

Product Engineering with AI

Registered public evidence

[multi-cli-work repository](#)

[Daegu bus app repository](#)

What to bring

Share the problem, data or sensors, target validation, and schedule. We can define coordinate and evidence boundaries together.

uiop3847@naver.com

rafaam11.github.io