

CASE STUDY

How Peer AI built a Claude-native SDLC for regulated medical writing

Introduction

Peer AI is an AI-native medical writing platform built for pharmaceutical sponsors and contract research organizations (CROs). It automates the most demanding class of clinical trial documentation, covering patient narratives, clinical study reports, and regulatory submissions. In that class, accuracy and regulatory compliance are not product features but preconditions, and a hallucinated detail is a compliance exposure rather than a bug.

A platform held to that standard cannot be built by a delivery process held to a lower one. Peer AI built the platform on a Claude-native Software Delivery Lifecycle, embedding Anthropic's Claude into specification design, pull request review, vulnerability patching, and QA architecture. Routine first-pass review, security-alert grouping, and documentation drafting now run through Claude-assisted workflows, with Peer AI's engineers retaining approval authority and spending their time on architecture, product decisions, and clinical-domain judgment. Peer AI's engineering team builds and runs the platform together with Zemoso, its product engineering partner. The two teams jointly designed and operated the Claude workflows described here.

Industry Challenge

High-stakes product requirements, compounded by delivery overhead

An enterprise life sciences platform creates intense operational and technical demands across two distinct fronts:

The product risk: Synthesizing complex clinical trial data, regulatory guidelines, and patient-level metrics into draft documentation allows no margin for error. Unhandled edge cases in patient narrative

generation expose organizations to severe regulatory liability.

The security alert trap: Under strict security controls and SLA commitments, manual vulnerability triage forces engineering teams to constantly trade off between patching security risks immediately or pausing strategic feature development.

Partnership challenge

To help Peer AI scale its platform while upholding strict regulatory and architectural discipline, Zemoso and Peer AI jointly engineered solutions for four core technical barriers:

- **PR review queues:** Pull request volume grew faster than synchronous human review could absorb, creating queuing delays and uneven convention enforcement across active repos.
- **Specification bottlenecks:** Design decisions and technical specifications required consistent senior review, creating a natural coordination constraint as the platform scaled and domain knowledge spread across more services.
- **Security and vulnerability friction:** Under Peer AI's security controls and SLA commitments, security alerts must be triaged and resolved within defined timelines. Manual triage forces the same trade-off on every alert: patch now and pause feature work, or defer and carry the risk.
- **Test suite maintenance:** A ~290-test Playwright end-to-end suite across 53 spec files gives broad coverage of critical flows. Maintaining that depth by hand scales poorly against a rapidly evolving UI.

How did we do this

Peer AI embedded Claude across three operational workstreams, each aimed at one of these constraints directly.

1. Spec-driven engineering on grounded context

Specification drift is the failure mode that matters most in a regulated domain: a model that does not know a system's conventions will confidently violate them. Every active repository therefore carries a canonical CLAUDE.md file covering architecture guidelines, coding standards, domain invariants, and explicit anti-patterns. That file is the contract every agent reads before it acts.

Peer AI's engineers work a structured loop:

Plan → LLM-reviewed spec → Implementation → Spec-grounded review. Architecture documents, C4 diagrams, and API specifications that previously took several days are now drafted and reviewed the same day.

2. Automated pull request review

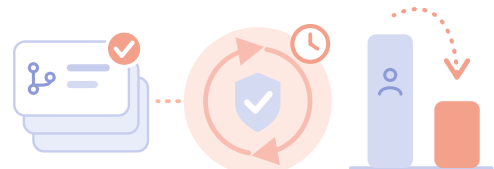
A Claude review agent is wired into CI/CD through GitHub Actions and processes pull requests across active repositories before a senior engineer engages. It checks each change against that repository's CLAUDE.md standards, flagging convention violations, logic gaps, and formatting inconsistencies. Peer AI's engineers arrive at a pull request already cleared of routine defects, and spend their attention on design and clinical correctness.

3. Two-stage autonomous vulnerability pipeline

Compliance obligations do not pause for the roadmap. Peer AI runs an automated security pipeline on a standing 48-hour cadence, split across two agents:

- **Analysis agent:** Pulls open Dependabot alerts, groups related CVEs into logical batches, assesses risk, and files consolidated GitHub issues, so the team makes a handful of triage decisions rather than working a stream of individual alerts.
- **Fix agent:** Applies the dependency updates (langchain, langgraph, pydantic, and tornado among them), verifies the build locally, and submits a pre-verified pull request. The originating issue closes automatically on merge.

Impact



Successfully automated first-pass reviews for 190+ monthly pull requests, cutting senior review effort by 30-40%, while slashing documentation cycle times by 50-70% and turning vulnerability resolution into a standing 48-hour autonomous pipeline



This systemic transformation accelerates delivery velocity, drastically reduces security noise by 10x, and frees senior engineering leadership to dedicate 80%+ of their time to high-value architecture and clinical domain judgment.

Conclusion

Building enterprise-grade AI products requires an engineering pipeline as sophisticated as the application itself. By pairing deep platform architecture with a Claude-native SDLC, Peer AI's engineering team ships a medical writing platform built for regulated use while keeping senior engineering attention on architecture, product judgment, and clinical-domain review. The same pipeline has removed the operational friction that typically slows engineering teams as they scale.

Related resources

Case Study

How MagicBI built an AI-Native Analytics Platform with Anthropic's Claude...



Case Study

Digitally Transforming Resource Allocation for a Data An...



Blog

AI observability: Choosing the right tool for produc...

