

From undocumented legacy code to a clear modernization plan

Years of undocumented legacy code, running mission-critical operations, were made understandable again — without a ground-up rewrite.



The Challenge

A platform handling routing, reconciliation and compliance reporting for a financial services company had been built on legacy code for years. The software worked, but it was difficult to understand, document and extend. The original developers were gone, and no one could answer basic questions about module functions. Business logic and operational rules lived only in the code itself, unreadable to anyone but a developer willing to trace it line by line. New engineers needed months to become productive, and every feature request required reverse-engineering.

The company knew modernization was necessary, but didn't know how to start.



What Bob Did

Working from deep knowledge of how systems like this operate, Bob generated a human-readable description of what each module does, written in language a business stakeholder could understand. Bob also generated code and business logic that closely mirrored systems a financial services company would run in production.

Bob then organized the inventory into a risk-ranked assessment, modernized and validated a set of representative modules against their original behavior side by side, and produced a phased roadmap with effort and cost estimates for modernization ahead.

What started as an inventory of an unreadable system became a concrete plan leadership could act on.



BOB FACT

New engineers used to need months to understand this system. I read it and explained it in plain English the same day.

DATA POINT:

Months of onboarding → same-day comprehension.

Takeaway 1: Bob as knowledge worker

Bob reconstructed the shape of an unfamiliar system from domain knowledge and code patterns alone, producing a plain-English inventory a business stakeholder could read — not just a developer.

Takeaway 2: Bob as modernization accelerator

From function inventory to modernized, validated modules to a phased roadmap with cost estimates, Bob operated across the full assessment chain without ever touching the original codebase.

Takeaway 3: Legibility is a starting point

A modernization plan, a roadmap, a safe starting point — none of it was possible until the system could be read again. Making it legible was the real unlock.