Rethinking Legacy Systems:

AI Modernization

huenei 30 años
IT services

Table of Contents

01 Why Modernize Legacy Systems?

02 Prompt Engineering as a Change Accelerator

03 A Step-by-Step Approach to Modernization

04 Critical Migration, Enabled by AI

05 The Value of a Hybrid Approach

06 What’s Next with Huenei

Why Modernize Legacy Systems?

Many organizations still rely on software built over a decade ago. These systems were once vital, but today, they often slow down innovation and scalability.

Why? Because they tend to come with problems like:

- ✓ Little or no documentation
- ✓ Difficulty scaling or integrating
- ✓ Monolithic architectures
- ✓ Outdated languages and frameworks (VB6, .NET Framework, PHP...)
- ✓ High dependency on specific technical roles or vendors

Modernizing these systems helps regain agility, reduce risk, and unlock long-term business growth.

However, doing it the traditional way can be slow, expensive, and error-prone. That's why at Huenei, we've developed a new AI-powered approach.

Prompt Engineering as a Change Accelerator

Migrating legacy systems often requires months of code analysis, reverse engineering, and manual effort. At Huenei, we've found an unexpected but powerful tool to speed this up: *Prompt Engineering.*

We use prompts to:

- Understand undocumented legacy code
- Design modern target architectures
- Generate tests, documentation, and refactors
- Capture technical knowledge scattered across the system

This reduces lead time, improves accuracy, and helps teams preserve essential knowledge during migrations.

We don't replace developers — we empower them with AI so they can focus on what matters most.

A Step-by-Step Approach to Modernization

Our process is structured into five phases:

01 AI-Assisted System Discovery

- Prompts to interpret legacy code
- Identify dependencies, critical logic, and code duplication
- Auto-generated summaries per class, module, or feature

Helps understand complex systems without reading line by line

02 Target Architecture Proposal

- Design modern architectures (microservices, serverless, REST APIs, etc.)
- Use prompts to simulate migration paths
- Evaluate risk and suggest suitable patterns (strangler, event sourcing...)

03 Assisted Refactoring & Generation

- Prompts that translate old code to modern stacks (e.g. .NET Framework to .NET Core)
- Automated generation of unit tests
- Align code with current performance, logging, and security standards

04 Live, Reusable Documentation

- Prompts like "Explain this class as a senior software architect would"
- Generate OpenAPI specs, business flow diagrams, README files
- Versioned documentation from Sprint 1

05 Continuous Validation & Evolution

- Integration with CI/CD pipelines
- Automated reviews, changelogs, and static analysis via prompts
- Prompt-driven validation of refactors and test coverage

Critical Migration, Enabled by AI

One of our most complex projects involved a critical system built over 15 years ago in a now-discontinued language.

The challenges:

- ✓ Zero documentation
- ✓ Highly coupled codebase
- ✓ Business logic fragmented across multiple layers

Our approach:

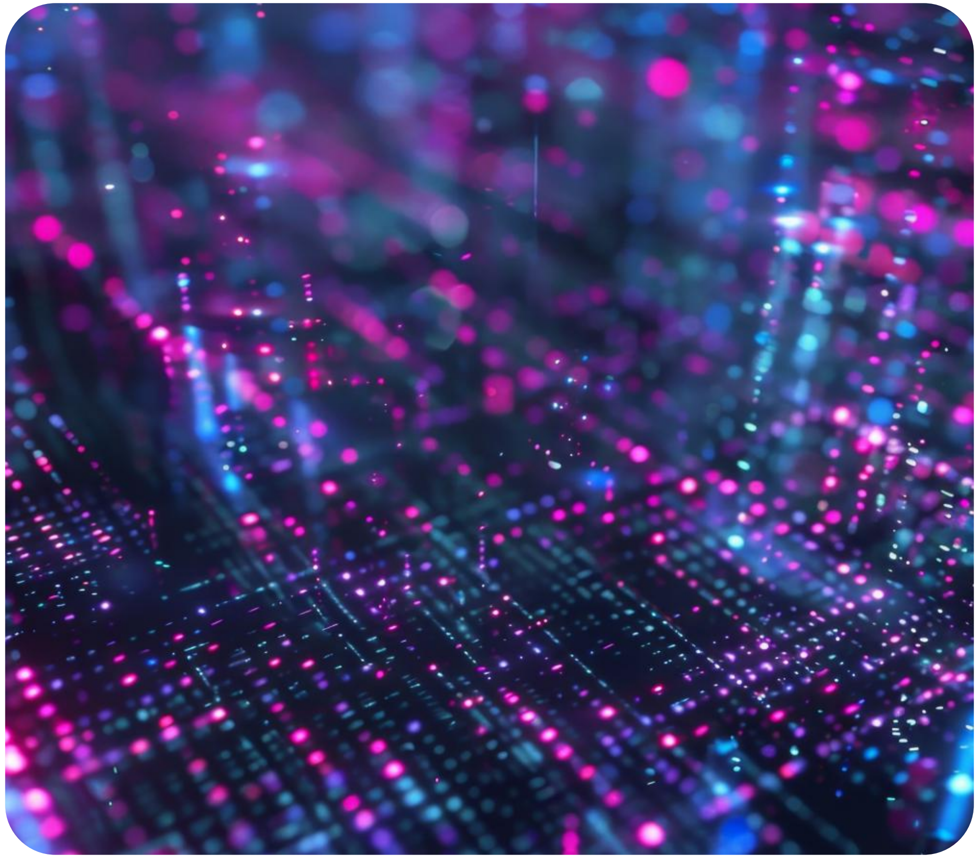
- Prompt-assisted code discovery
- Architecture redesign completed in just two weeks
- Progressive migration with AI-generated tests and documentation
- Delivery of a modern, scalable system with full documentation and trained teams

The result: A validated migration in record time, with lower technical risk and a solid foundation for new features**.**

The Value of a Hybrid Approach

Beyond speed, this approach changes the way teams work. Some concrete benefits we're seeing:

- Accelerated execution of previously manual, repetitive tasks
- High-quality documentation from day one
- Reduced dependency on tribal knowledge or hard-to-find specialists
- Better alignment with functional and technical standards
- Improved collaboration across teams through structured natural language



We also train our teams to write better prompts, implement versioning practices, and validate outputs with engineering rigor.

What’s Next with Huenei

Prompt Engineering isn’t just something we’re testing — it’s already delivering results across multiple projects.

Most importantly, it doesn’t replace people — it augments them.

At Huenei, we believe this hybrid approach, blending engineering, AI, and conversation design, is the most effective way to modernize legacy systems without losing valuable business knowledge.

Is your team facing challenges with legacy systems?

We'd love to help you explore the path forward.

Let's talk

huenei 30 años IT services

www.huenei.com