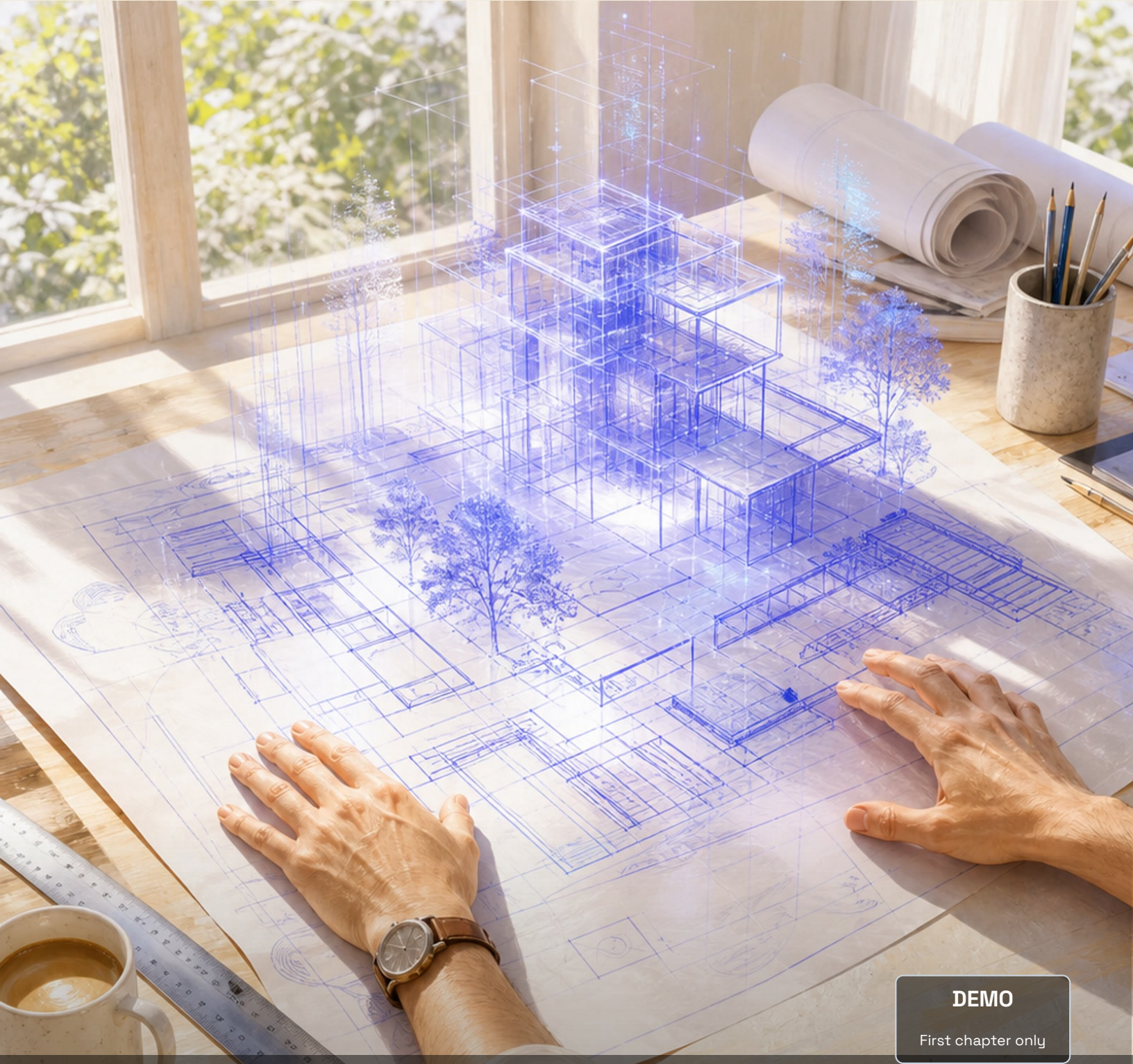

DEMO

First chapter only

Spec-First AI Development

The Workflow That Replaced Our Engineering Team

pragma.vision

SOFT.HOUSE

Spec-First AI Development

Trademark Notice

Google, Google Pay, Google Cloud, and Android are trademarks of Google LLC. Stripe is a trademark of Stripe, Inc. Cloudflare and Cloudflare Workers are trademarks of Cloudflare, Inc. Supabase is a trademark of Supabase, Inc. OpenAI and ChatGPT are trademarks of OpenAI, Inc. Claude is a trademark of Anthropic, PBC. W3C is a trademark of the World Wide Web Consortium. Visa is a trademark of Visa International Service Association. OWASP is a trademark of the OWASP Foundation. Midjourney is a trademark of Midjourney, Inc. Canva is a trademark of Canva Pty Ltd. Etsy is a trademark of Etsy, Inc. Amazon is a trademark of Amazon.com, Inc. All other trademarks are the property of their respective owners.

No Affiliation

This book is an independent publication. It is not authorized, sponsored, or endorsed by any of the companies or organizations whose products or services are mentioned herein.

No Professional Advice

The information in this book is provided for educational purposes only. It does not constitute legal, financial, investment, tax, or other professional advice. Readers should consult qualified professionals for guidance specific to their situation.

Code Examples

Code examples in this book are provided for illustration only. They may not be suitable for production use without additional validation, error handling, and security review.

Published by Pragma Vision LLC

First edition, 2026.

Contents

1

The Specification IS the Product

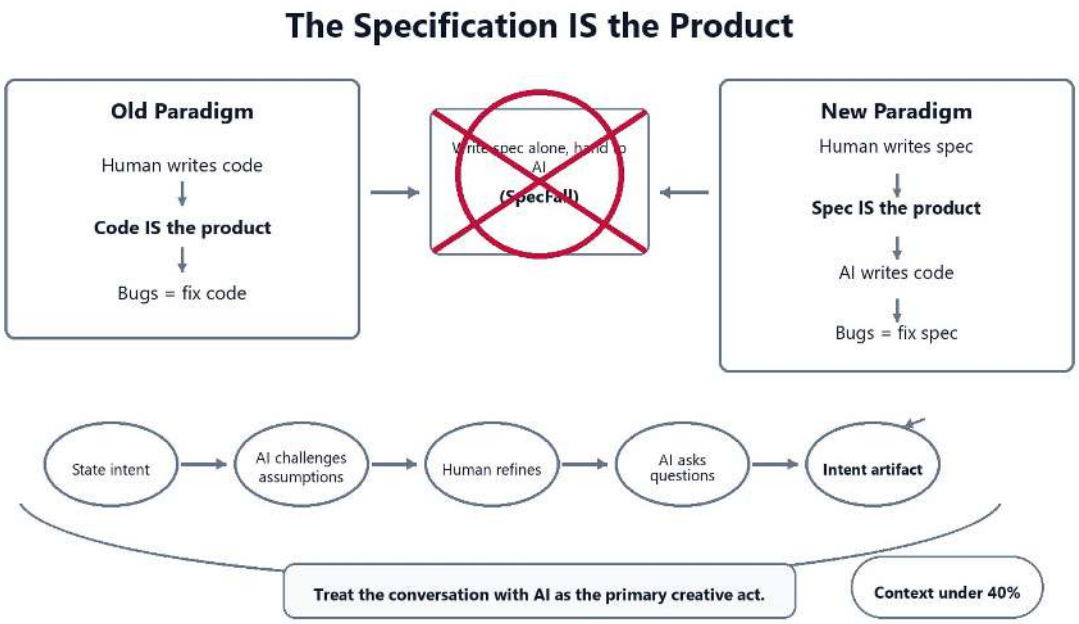

Figure 1. In the old paradigm code is the product and bugs mean fixing code; in the new one the spec is the product—a human writes it, AI writes the code, and bugs mean fixing the spec, refined through dialogue rather than handed off alone

In 2024, we were a team of one. No junior developers. No senior architects. No QA department. Just a founder with a vision for an AI-native commerce ecosystem spanning a growing family of interconnected platforms—and an AI coding agent.

By 2026, we had built over 300 passing end-to-end tests, implemented systems for cryptographic transaction flows, achieved 69.5% code sharing across two competing payment protocols, and kept the early build within a $1,800 budget. Zero dollars spent on engineering salaries. Zero dollars spent on contractors.

The secret was not the AI. The secret was *what we fed the AI*.

5x

effective team multiplier achieved by one developer using spec-first AI development

1.1 The Great Inversion

For fifty years, software development has followed a simple hierarchy: requirements at the top, code at the bottom, and a long chain of translation in between. Business analysts wrote requirements. Architects designed systems. Engineers wrote code. Testers verified output.

AI coding agents have inverted this hierarchy. Code is no longer the bottleneck. Understanding is.

When an AI agent can generate hundreds of lines of syntactically correct code in seconds, the question stops being "can we write this?" and becomes "should we write this, and does the agent understand what 'this' actually means?"

Key Insight

The specification is not documentation about the code. The specification IS the product. Code is merely its compiled output—a lossy projection of your intent into executable instructions.

This is not a theoretical position. Studies show that 48% of AI-generated code contains security vulnerabilities. Not because the AI is incapable, but because the humans directing it provided ambiguous, incomplete, or contradictory specifications. The AI did exactly what it was told. The problem was what it was told.

1.2 About Pragma.Vision

Pragma.Vision is an AI-native commerce ecosystem—a growing family of interconnected platforms sharing identity, payments, and infrastructure. From **wish.now** (conversational commerce) to **phantoid.com** (the App Store for AI Agents) to **trustauthority.ai** (the Certificate Authority for AI), the platform specifications and implementation plans use the workflow described in this book.

We did not arrive at spec-first development by reading a methodology paper. We arrived at it by watching AI agents produce spectacular failures when given vague instructions, and spectacular successes when given precise ones.

This book extracts the workflow we used—the five-phase Discover-Research-Plan-Implement-Retrospective cycle—and makes it repeatable for any developer working with AI coding agents.

1.3 What This Book Covers

Each chapter maps to a phase of the spec-first workflow:

Chapter 2: Discover

Shaping intent through AI dialogue—the phase most teams skip and most regret skipping.

Chapter 3: Research

Understanding the system with file paths, line numbers, and data flow before writing a single line of code.

Chapter 4: Plan

Designing changes with explicit steps, acceptance criteria, and verifiable outcomes.

Chapter 5: Implement

Executing with fresh context, frequent commits, and aggressive context management.

Chapter 6: Retrospective

Closing the feedback loop by classifying issues as spec gaps, agent deviations, or harness gaps.

Chapter 7: Anti-Patterns

The failure modes we discovered—SpecFall, context overload, plan drift, and the solo-spec waterfall.

Chapter 8: Building Your Practice

Templates, `CLAUDE.md` rules, and team adoption strategies.

1.4 Who This Book Is For

This book is for developers who have used AI coding tools and gotten inconsistent results. You have experienced the magic of a perfectly generated function *and* the frustration of a hallucinated API that does not exist. You know AI can write code. You

want to learn how to make it write the *right* code, consistently, across complex multi-file systems.

Pro Tip

If you are entirely new to AI coding agents, start with any basic tutorial for Claude Code, Cursor, or GitHub Copilot. This book assumes you can prompt an AI to generate code. It teaches you the meta-skill of structuring that interaction for production-quality output.

1.5 The Economics of Specification

Here is the uncomfortable math. A senior developer costs $150,000–$250,000 per year. An AI coding agent subscription costs $200 per month. The gap between those numbers is not filled by the AI being "smart enough." It is filled by the human being precise enough.

84%

of developers now use AI coding tools weekly—but only a fraction use structured specification workflows[1]

The developers who achieve 5x productivity multipliers are not better prompters. They are better specifiers. They have internalized a workflow that transforms ambiguous intent into precise, verifiable instructions that an AI agent can execute reliably.

That workflow is what you hold in your hands.

---

[1] GitHub, "Octoverse: The State of Open Source and AI," 2024.

DEMO

This is a free preview of the full edition.

Get the complete book at:

https://shop.pragma.vision