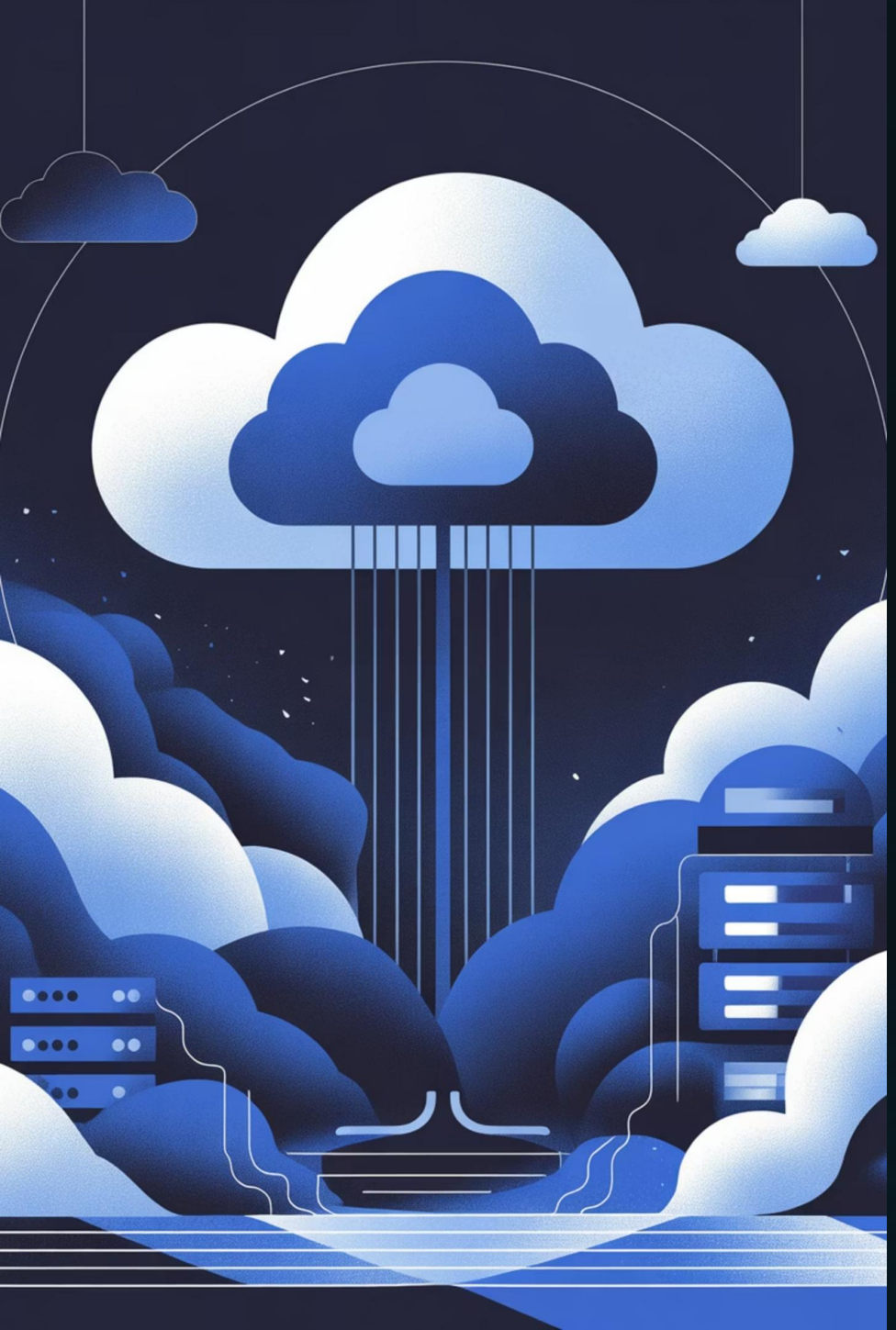


AI-Powered Application Modernization

From Legacy Code to Cloud-Native with Claude-Flow

Derek Ashmore

Meet Derek Ashmore

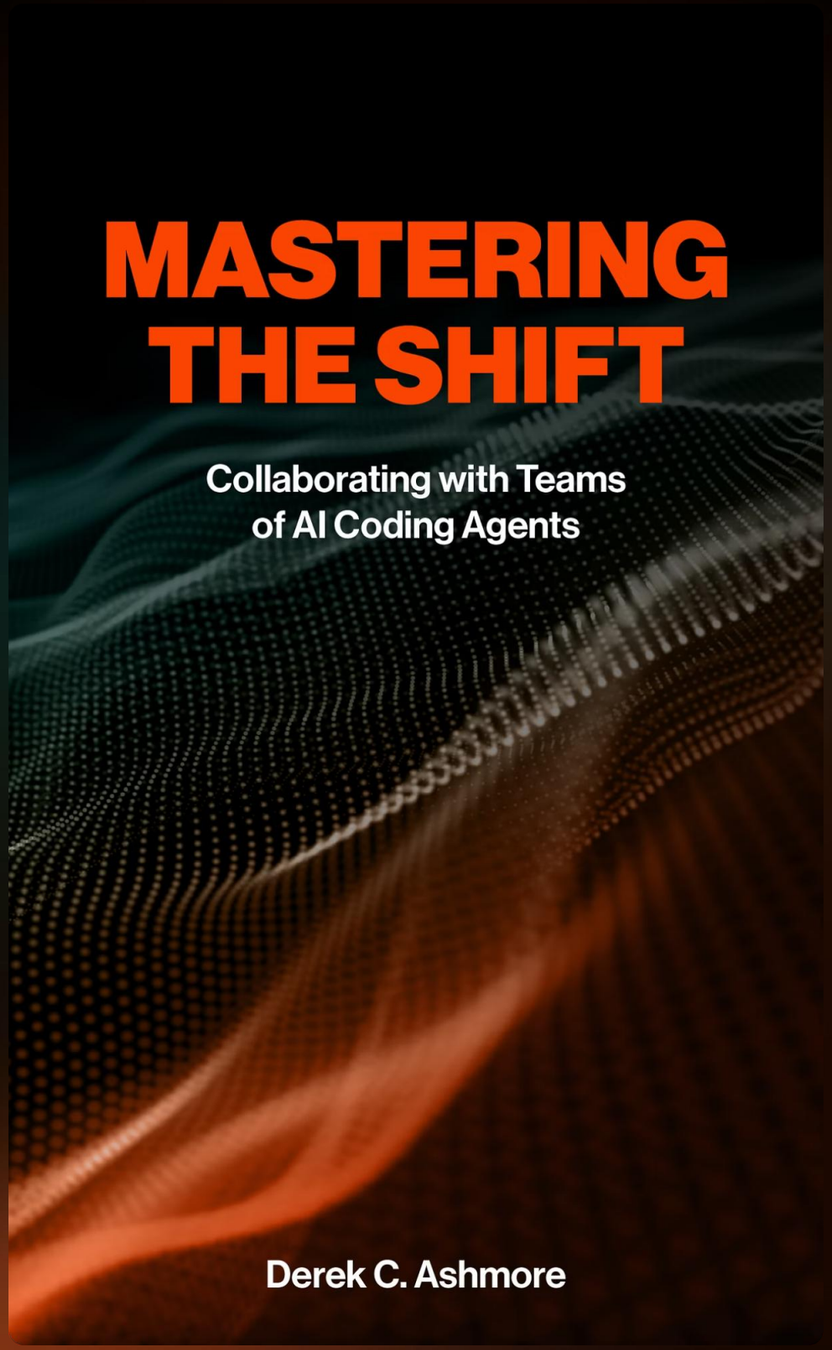
A Professional Geek since 1987, I bring decades of experience to the evolving landscape of IT.

My Expertise Includes:

- Applying AI tooling for IT innovation
- Driving Application Transformation initiatives
- Pioneering Infrastructure Automation solutions

Yes — I still code!

Asperitas CONSULTING



Agenda

A roadmap of our discussion on AI-powered application modernization.

1 Defining Modernization & Tooling

An overview of what application modernization entails and the AI-driven tools we'll leverage, including Claude-Flow.

2 Applying AI to the Orienteer CRM

We'll dive into practical applications, focusing on:

- Achieving a truly cloud-native architecture
- Exploring optimal migration strategies and possibilities

Application Modernization Defined

What It Is

Updating older software for newer computing approaches, commonly associated with cloud migration.

The Goal

Application portability—making hosting decisions a choice, not a technical requirement.

Cloud Benefits

- Increased availability and higher uptime
- Dynamic scaling to handle load spikes
- Reduced costs during off-hours

12-Factor Principles

Framework describing if an app can leverage high availability and dynamic scaling features.

Why Modernize?

Too Hard to Change

The application has become difficult to modify, often requiring improved change management processes.

Cloud Migration

Management wants cloud hosting but realizes lift-and-shift rarely works well without modernization.

Why Claude-Flow?

Quick Setup

Install and run within minutes, not hours or days. Default configuration works for the vast majority—it just works.

Powerful Resources

Dozens of MCP tools and 16 specialized agents (coder, researcher, analyst, tester, coordinator, etc.).

Smart Assembly

Claude-Flow assembles the agent team based on your request, determining the right mix of specialists.

12-Factor Orienteer: Specification Statements

The following detailed instructions were provided to Claude-Flow to initiate the 12-Factor assessment of the Orienteer application:

1 Perform Comprehensive Analysis

Conduct a thorough 12-Factor analysis on the designated application, ensuring all findings are meticulously documented and organized within the specified `12-factor` folder.

2 Reference Official Principles

Utilize the authoritative guidelines and definitions for 12-Factor principles as outlined on the official website: https://www.12factor.net/.

3 Identify Compliance Gaps

Specifically highlight areas where the application deviates from the 12-Factor principles. This critical assessment will inform its readiness to leverage public cloud dynamic scaling and high availability features upon deployment.

Orienteer 12-Factor Analysis Results

The Process

I had my team perform a twelve-factor analysis to assess cloud readiness. I didn't micro-manage the approach.

Time to complete: ~10 minutes

Human equivalent: 1-2 weeks

Key Findings Summary

- The app uses stateful sessions, preventing dynamic scaling.
- Hardcoded credentials pose security risks.
- Health checks are absent.
- Its monolithic architecture hinders independent scaling of components.

📊 12-Factor Compliance Scorecard

Factor	Score	Status	Critical Issues
I. Codebase	7.5/10	✅ Good	Single repo, multiple deployable artifacts
II. Dependencies	8.5/10	✅ Good	Explicit Maven dependencies, no vendoring
III. Config	3/10	🔴 **Critical**	Hardcoded credentials, secrets in codebase
IV. Backing Services	6/10	🟡 Moderate	Embedded database default, limited abstraction
V. Build, Release, Run	7/10	✅ Good	Docker support, but config in build
VI. Processes	2/10	🔴 **Critical**	Stateful sessions, requires sticky routing
VII. Port Binding	10/10	✅ Excellent	Self-contained with embedded Jetty
VIII. Concurrency	4/10	🔴 Poor	Monolithic, no process separation
IX. Disposability	3/10	🔴 Poor	No graceful shutdown, slow startup
X. Dev/Prod Parity	4/10	🔴 Poor	Environment-specific code paths
XI. Logs	6/10	🟡 Moderate	Stream-based but basic implementation
XII. Admin Processes	7/10	✅ Good	Robust migration system, console support

It's not perfect, but it's good enough. The 80/20 rule applies, and my team can quickly answer follow-up questions.

Implementation Planning: Specification Statements

To guide Claude-Flow in generating a comprehensive implementation plan for Orienteer's 12-Factor compliance, the following instructions were provided:

1

Leverage Analysis

Utilize the documented 12-Factor analysis from the `12-factor` folder as the foundational input for creating the plan.

2

Agentic Execution

The implementation itself is to be performed by agentic engineers, leveraging Claude-Flow and Claude Code for the modernization process.

3

Directed Output

Deposit the generated implementation plan into the specified `12-factor-plans` folder for organized access and review.

4

No Immediate Changes

Crucially, the Orienteer application is not to be modified during this planning phase; focus solely on plan generation.

Implementation Planning

Key Implementation Steps:

- **Automated Test Suite:** Establish acceptance tests before changing anything—enables safer modifications and increased change velocity.
- **Externalize Sessions:** Move stateful sessions to Redis—no sticky-sessions required, supports graceful shutdown.
- **Async Operations:** Leverage message queues for asynchronous processing and better scalability.
- **Multi-Process Support:** Enable multiple concurrent processes for improved availability and performance.

Executive Summary

Total Investment Required

Category	Estimate	Details
Human Labor	$800K-1.2M	5-8 FTE for 8-10 months + part-time support
Infrastructure	$40K-60K	Cloud resources, testing tools, CI/CD
Training & Tools	$15K-25K	Licenses, courses, certifications
Contingency (15%)	$130K-190K	Risk buffer
Total Investment	**$985K-1.475M**	Complete transformation

ROI Analysis

Benefits (Annual):

- Infrastructure cost reduction: $150K-250K/year (30-50% savings)
- Reduced maintenance: $100K-150K/year (fewer production incidents)
- Faster feature delivery: $75K-100K/year (2x deployment frequency)
- **Total Annual Benefit**: $325K-500K/year

Payback Period: 24-36 months

Can I Trust AI Analysis?

Reframe the question

Do I trust Claude Code more or less than a human team?

All forecasts have error margins

Whether from AI or humans, no plan is perfect.

Claude ignores corporate politics

Objective analysis without organizational bias.

Unbeatable speed

15 minutes from start to finish, with rapid follow-up responses.

SaaS Migration Analysis Process

- Delivered results in ~15 minutes vs weeks for human teams
- Key insights:
  - Would benefit from specific client meta-data
  - Start of a conversation, not the end

Key Takeaways

For Architects

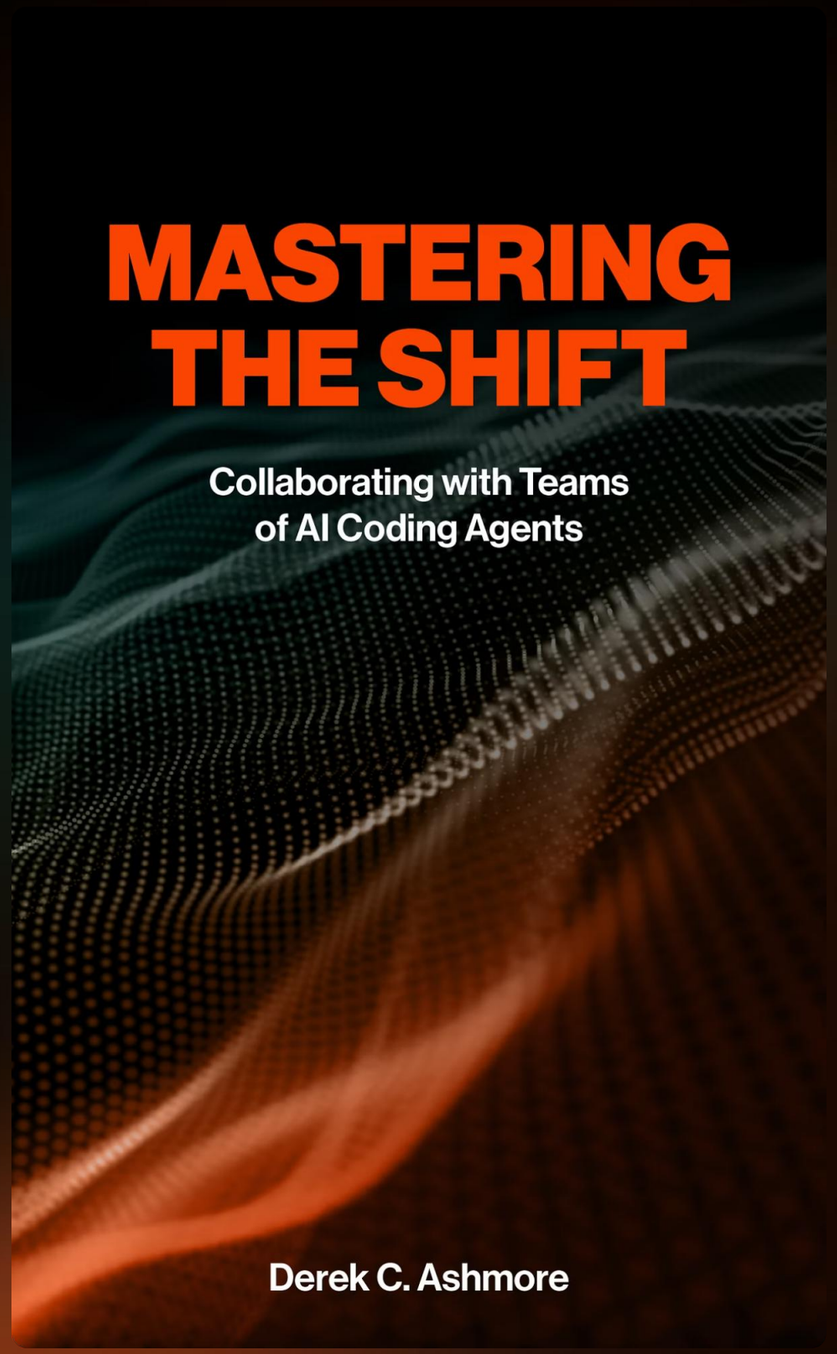
Agentic engineering prepares you for modernization conversations with more complete, consistent analysis than human teams provide. Run alternative scenarios quickly for management.

For Team Leads

Prepare for migrations more completely and quickly than manual approaches. AI sometimes surfaces issues humans forget.

For Agile Teams

Formulate larger-scope stories for agentic engineers. Whether migration or upgrade, AI helps break down work effectively.



Thank You!

We appreciate your time and interest in AI-powered application modernization.

For further inquiries or to discuss how Claude-Flow can transform your applications, please feel free to reach out:

- Derek Ashmore - Agentic AI Enablement Principal
- derek.ashmore@asperitas.consulting

Asperitas CONSULTING