
GUIDE

Vibe Coding in Production

How to Ship AI-Generated Code Without Losing Sleep

Jamie Bykov-Brett | Bykov-Brett Enterprises

Turn Intent Into Impact.

What Vibe Coding Actually Is

Not just using AI to generate code. It's a fundamentally different relationship with implementation.

NOT VIBE CODING

Using AI tools in a tight feedback loop. Cursor, Copilot, reviewing every line. You're still the developer.

You understand every line of code that ships.

VIBE CODING

"Fully give into the vibes, embrace exponentials, and forget that the code even exists."

The question isn't whether to delegate. It's how to do it safely.

Andre Karpathy

Why This Matters Now

AI task length is doubling every 7 months. The window to learn this is closing.

NOW

~1 Hour Tasks

You can still review every line. Cursor and Claude Code handle features. You stay in the loop.

NEXT YEAR

Full Day Tasks

AI generates a day's work at once. You can't read it all line-by-line. You need a new approach.

2028+

Week-Long Tasks

Entire systems built in one pass. If you demand to read every line, you become the bottleneck.

Forget the Code, Not the Product

Managing implementations you don't fully understand is a problem as old as civilisation.

CTO

Writes acceptance tests for an expert in a domain they don't understand. Tests pass = work is good.

PM

Uses the product the engineering team built. It works as expected = feature shipped.

CEO

Spot-checks key facts in the accountant's work. Key slices verified = confidence in the whole.

YOU

Design verifiable checkpoints for AI-generated code. Inputs and outputs verified = safe to ship.

Be AI's Product Manager

What guidance would a new hire need on their first day to succeed at this task?

01

Collect Context

Spend 15–20 minutes gathering requirements, constraints, and relevant files into a single prompt. Use a separate conversation to explore the codebase first.

TOUR THE CODEBASE FIRST

02

Set Requirements

Define what you want at the end, not how to get there. The AI does best when you don't over-constrain it. Think outcomes, not implementation.

SPECIFY OUTCOMES, NOT STEPS

03

Ask the Right Questions

Non-technical people can't be effective PMs for production code. You need to know enough about the domain to guide Claude in the right direction.

KNOW WHAT'S DANGEROUS

Leaf Nodes vs Core Architecture

Not all code is equal. Vibe code the edges. Human-review the trunks.

LEAF NODES

Vibe Code These

- End features and bells/whistles
- Nothing else depends on them
- Unlikely to change or be extended
- Tech debt is contained and harmless
- Verify via inputs and outputs

CORE ARCHITECTURE

Human-Review These

- Trunks and branches of your system
- Other code is built on top of them
- Will change and need to be extended
- Tech debt compounds and spreads
- Must stay extensible and flexible

Verifiable Checkpoints

Design systems so you can verify correctness without reading the implementation.

01

Design human-verifiable inputs and outputs

Structure the system so that a human can check whether the result is correct by inspecting what goes in and what comes out.

02

Build stress tests for stability

Create tests that run at scale and duration. If the system survives the stress test, you can trust it without reading every line.

03

Write end-to-end tests, not unit tests

Encourage minimal, general tests: happy path + key error cases. Avoid implementation-specific tests that break when code changes.

04

Read the tests first, not the code

If you agree with the tests and the tests pass, you can feel confident about the code. The tests are your abstraction layer.

The 22,000–Line PR

Anthropic merged a 22K-line change to production RL code, written heavily by Claude. Here is how they did it responsibly.

01

Days of Human PM Work

Requirements, guidance, system design. Not a single blind prompt.

02

Leaf Node Concentration

Changes targeted areas where tech debt was acceptable.

03

Review the Core

Core architecture got full human review. Extensibility matters here.

04

Test the Outputs

Human-checkable inputs and outputs. Stability measured at scale.

"The marginal cost of software is lower. Things that take two weeks now take a day."



The skill that matters now isn't writing the code. It's knowing what to verify.

The engineers and leaders who thrive will be the ones who learn to manage AI the way great PMs manage teams.

Jamie Bykov-Brett

Ready to Ship With Confidence?

Whether you are exploring AI-assisted development or already shipping with agents, the principles in this guide will help you do it responsibly.

[Visit bykovbrett.net](https://bykovbrett.net)

[Get in Touch](#)

jamie@bykovbrett.net

Turn Intent Into Impact.