

Spec-Driven Development

Ship at AI speed — with an audit trail your future self will thank you for.

You write the specs. AI agents build the code. Every line of production code traces back to intent you approved — not to a chat session nobody can reconstruct.

Your specs !' Business intent, design, contracts

Agents build !' Frontend, backend, tests

Production !' Gated, traceable, yours

Proven in production: Designed and ran this model for a Fortune-100 automotive OEM — two applications, ~10× leverage on human effort.

THE PROBLEM

Your AI is already writing code. Who governs it?

With AI in the loop, the real choice was never AI vs. hand-coding. It's vibecoding vs. spec-driven.

	VIBECODING	SPEC-DRIVEN
Source of truth	Chat history & developer memory	Versioned, reviewed specification
Consistency	Varies by prompt, session, person	One constitution — same gates, every run
Audit trail	Can't reconstruct why code exists	Business intent !' model !' code !' test
Scale	Bottlenecks on whoever prompted	New contributors inherit the rules
Governance	Little to none	Human checkpoints + automated gates
		Vibecoding produces working code — and no durable asset. Six months on, nobody can say why a behavior exists.

THE MODEL

You own intent. Agents own labor.

YOU AUTHOR · with my help

- Business spec — Plain-language intent + Given/When/Then acceptance criteria
- Visual model — A committed prototype + design intent — the visual law
- Domain model & API contract — One source of truth; the API contract is generated from it
- Constitution — A short set of binding rules every build must obey

AGENTS BUILD

- Frontend + backend — React/TypeScript UI and API services, end to end
- Tests & docs included — Auth, error handling, layering — built in, not bolted on
- Gates run by the agent — The agent verifies its own work against your specs
- "Nothing is made up" — unclear spec? The agent stops and asks.

Every feature runs the same four stages:

specify !' plan !' tasks !' implement

GOVERNANCE

Agent speed. Human-defined correctness.

TIER 1 · BUILD-TIME GUARDRAILS

- Binding specs — Acceptance criteria are requirements; contracts can't be contradicted
- Human checkpoints — The run pauses at defined points — no code until you approve the intent
- Conformance checks — Architecture and model conformance verified mechanically, every run
- Fidelity checks — Visual parity to the design; requirement-to-test traceability

TIER 2 · EVERY AGENT PR CLEARS THE SAME BAR AS HUMAN CODE

- AI code review — A second model reviews every pull request
- Quality & security scan — Static analysis on every change
- License & dependency check — No surprise licenses in your stack
- API security conformance — Contract-level security checks

Wired to your stack — these map to whatever scanners and review tools you already run. Nothing ships that your specs and your gates didn't approve.

P R O O F

This runs in production today.

Designed and ran this operating model for an engineering platform at a Fortune-100 automotive OEM — two production applications, built by agents, governed by specs.

"H10×

leverage on human effort

28,600

lines of production code

3,000

lines of spec govern it all

300

acceptance criteria traced

0 architecture violations

Application 1

· Web UI

9 business specs !'

12 live screens.

React / TypeScript ·

workflow

management,

authoring,

publishing,

dashboards — pixel-

parity to the

committed design.

Application 2

· API services

5 business specs !'

19 endpoints.

FastAPI /

SQLAlchemy ·

domain-model-

driven, generated

API contract, full

audit trail from intent

to test.

Not a pilot. Not a demo. This is how both applications are built and shipped, feature after feature.

Three phases. A real feature ships in the pilot.

01 FOUNDATION · Weeks 1–2

- Constitution written for your product and stack
- Spec templates + worked examples for your team
- Pipeline scaffold in your repo, gates configured
- Pilot feature selected from your real backlog

02 PILOT · Weeks 3–5

- Your first feature: spec !' plan !' tasks !' implement
- Human checkpoints live — you approve intent, not diffs
- PR gates enforced on the agent's pull requests
- Feature lands in production, fully traced

03 SCALE · Week 6+

- Your team trained to author specs independently
- Backlog throughput at machine speed
- Playbook + runbook handed over — documented, versioned
- Optional retainer for reviews and pipeline evolution

Fixed-scope foundation + pilot, then scale on your terms. If the pilot doesn't ship, you keep every artifact anyway.

D E L I V E R A B L E S

What you walk away with

Everything lives in your repos, under your accounts. The system doesn't depend on me.

Constitution & spec templates — Versioned rules + templates your team authors against from day one

Working spec!'code pipeline — The four-stage agent pipeline, running in your CI, on your repos

Quality gates, enforced — Checkpoints and PR gates wired so a red gate blocks a merge

Domain model + API contract — One source of truth; the contract is generated, never hand-drifted

A team that can run it — Your PMs and designers trained to author specs without me

Full traceability — Every shipped line answers: which spec, which criterion, which test

T H E A S K

The deliverable isn't code. It's a code factory you own.

Six weeks from kickoff, a real feature from your backlog is in production — specified by your team, built by agents, cleared through gates you control. Everything after that is throughput.

- Start with a 45-minute working session — We pick the pilot feature from your actual backlog
- Get a fixed-scope proposal in 48 hours — Foundation + pilot, one price, defined exit criteria
- Ship, then decide — Scale phase only if the pilot earns it

Jai Natanasabapathi · AI Application Development · Spec-Driven Delivery
njganeshv@gmail.com