Duane Grey

METHODOLOGY FULL LIFECYCLE

Proposal to Release

Software Delivery

A structured methodology for taking an idea from first conversation to deployed, production-ready product. Proven across two client MVPs with 15+ development cycles.

2	Status	Idea
MVPS DELIVERED	PORTAL	TO LAUNCH

TIMELINE	KEY RESULT	TECH STACK
Methodology refined over 6 months of active delivery	**Two client MVPs delivered from idea to production with full stack ownership**	TypeScript, React, Next.js, Python, FastAPI, PostgreSQL, Docker

The Problem

The space between having an idea and having a live product is opaque. Traditional agencies give you a timeline and disappear. Solo founders do not know what they do not know: scope creep, missed requirements, deployment surprises. Most projects fail not because of bad code but because nobody managed the journey from concept to production.

- ! No visibility into progress between kickoff and delivery
- ! Scope creep that turns a 4–week build into a 4–month one
- ! Deployment treated as an afterthought instead of a first-class concern
- ! Post-launch support that amounts to 'here is the code, good luck'

The Approach

A gated lifecycle with clear milestones: Discovery validates the real problem, Design proves the solution before writing code, Build delivers in short visible cycles, Deploy is containerized from day one, and Support means actually being there after launch. Each gate prevents the next phase from inheriting problems.

- ✓ Discovery produces a scope document with an explicit 'Intentionally Left Out' section so both sides agree on boundaries
- ✓ Design delivers validated screens and user flows before any code is written, preventing expensive rework
- ✓ Build uses 1–2 week cycles with a tracking portal so the client sees progress in real time
- ✓ Docker containerization from the first commit so production deployment is never a surprise

INTENTIONALLY LEFT OUT

This methodology intentionally excludes ongoing maintenance contracts and team scaling. It is optimized for the journey from zero to one: taking an idea to a working, deployed product. What happens after launch is a separate conversation.

The Solution

A five-phase delivery framework where each phase produces concrete, reviewable artifacts. The client never wonders what is happening because every phase ends with something they can see, touch, and approve. Beyond the build phases, the methodology handles what most freelancers improvise: communication protocols, change management, handoff packages, and post-engagement follow-up.

- ✓ Discovery phase with recorded briefing calls, scope documents, and technical architecture
- ✓ Design phase with validated screens, user flows, and database schema
- ✓ Build phase using 1–2 week cycles with a live tracking portal for real-time visibility
- ✓ Video walkthroughs at every milestone so the client sees exactly what was built and why
- ✓ Proactive weekly status updates so clients never have to ask what is happening
- ✓ Communication protocol with clear channel rules: chat for quick questions, email for formal deliverables, video calls for milestone reviews
- ✓ Change management system: requests classified by type (clarification, minor tweak, new feature, scope expansion, direction change) with a change log tracking impact on timeline and cost
- ✓ One revision round per milestone included in scope. Additional rounds are change requests with documented impact
- ✓ Containerized deployment with NGINX proxy, SSL, and monitoring from day one
- ✓ Tier-specific handoff packages: not just source code but documentation, deployment guide, and a 7 to 14 day support window
- ✓ Close-out process: testimonial ask, pricing analysis, attribution report, and opportunity mining from engagement transcripts
- ✓ Post-engagement follow-up cadence at 7, 30, and 90 days to maintain the relationship and surface new opportunities

Technical Highlights

Tracking portal gives clients real-time visibility into cycle progress, deliverable status, and upcoming milestones

Docker-first architecture means the production environment is identical to development from the first commit

Milestone-based payment structure aligns incentives: you pay for delivered value, not elapsed time

The Results

The methodology has been validated across two real client engagements, each starting from a conversation and ending with a deployed, production-ready product. Both engagements used the full lifecycle: discovery through post-launch support, with change management, milestone reviews, and structured handoffs.

2	Status	Idea
MVPS DELIVERED	PORTAL	TO LAUNCH

“I came in with an idea and Duane delivered everything, branding, logo, and a fully built MVP. Now I have a real product that anchors conversations about scaling into a larger platform.”

Brent — Entrepreneur

Lessons & Takeaways

Scope is a negotiation, not a document

The 'Intentionally Left Out' section is as important as the feature list. Agreeing on what you will not build prevents scope creep better than any contract clause.

Show progress weekly or lose trust

The tracking portal exists because clients need to see movement. A quick video walkthrough at every milestone takes minutes to record and saves hours of status update meetings.

Deploy on day one, not day last

Docker from the first commit means production is never a surprise. Every demo runs in the same environment the client will use. Zero deployment drama on launch day.

Change management is margin protection

Without a classification system, every small tweak quietly expands the scope. The change log makes the cost visible before it gets absorbed. Classifying requests by type (clarification vs. new feature vs. direction change) means both sides know what they are agreeing to before work starts.

Post-engagement is pipeline

The 7, 30, and 90 day follow-up cadence turns every completed engagement into future business. Testimonial collection, opportunity mining from transcripts, and pricing analysis from actual delivery data all feed back into the next proposal. The engagement ends but the relationship compounds.

Have a Similar Challenge?

Let's talk about how I can help solve it.

Start a Conversation