Chargeback Defender - Project Overview

Daniel Bacon | Built February 2026 | 9 days from conception to live product, including 6 days of active build

What Problem This Solves

Chargebacks hit fast. The response window is tight - typically 7 to 20 days depending on the processor - and the evidence requirements change by reason code. Miss the mark and the cost stacks up: you lose the transaction amount, pay a processor fee, and move one step closer to chargeback-rate thresholds that can trigger account review or termination.

Small merchants usually do one of two things. They take the loss by default, or they respond with incomplete evidence because the process is scattered across payment platform docs, card network rules, and whatever records they can pull together from the transaction.

Chargeback Defender is a 10-file operational kit that turns that mess into a single workflow: decide whether to fight, assemble the right evidence mapped to the exact reason code, submit cleanly, track outcomes, and reduce future exposure. It covers Stripe, Shopify Payments, and PayPal.

Live product: guideworks.co/chargeback-defenderLaunch Dashboard: guideworks.co/chargeback-defender/dashboard

System Components

Start Here Guide - The entry point. Orients the buyer to the workflow and the intended sequence of operations.

Operator Manual - A step-by-step playbook for the full dispute lifecycle: intake, triage, evidence assembly, submission, tracking, and prevention.

Fight or Flight Calculator (Excel) - A decision tool. Inputs: transaction value, dispute fee, time cost, win probability estimate. Output: a fight-or-accept recommendation with an ROI calculation.

Dispute Response Tracker (Excel) - Tracks open disputes, deadlines, evidence checklist status, submission notes, and outcomes. Includes a Rate Monitor tab that flags chargeback rate against processor thresholds (Stripe 1%, Shopify Payments 1%) with a days-to-breach trend indicator.

Evidence Pack Folder Structure + Naming Convention - A ready-made folder tree and standardized file naming so evidence is organized before submission.

Evidence Assembly Guide - Maps each evidence type to where it lives in the merchant's platform and how to export and package it. Includes a Stripe Compelling Evidence 3.0 field map covering all 7 required fields.

Template Library - 8 dispute response templates covering the most common chargeback types.

Post-Purchase Email Sequence - 3 transactional emails (order confirmation, shipping notification, delivery confirmation) designed to do double duty: help the customer and generate documentation that supports dispute responses.

Prevention Mini-Pack - A 35-point audit covering 3DS2 configuration, AVS settings, velocity checks, address verification, and policy clarity.

Reason Code Quick-Reference - Top 10 Visa, Mastercard, and Amex reason codes, with the specific evidence required for each.

Build Process and AI-Assisted Workflow

The kit went from idea to live product in 9 days - 3 days of scoping and planning, 6 days of active build. AI helped throughout, but output speed wasn't the constraint. Quality control was.

Scope was defined and locked before the build started. The 10-file structure, the platforms covered, the pricing, and the delivery method were decided up front. No features were added mid-build. That discipline matters - scope creep is how operational projects miss deadlines and ship with inconsistencies.

I wrote a project conventions document covering file naming, terminology, cross-reference rules, and a rule that new decisions get documented before they get implemented. Those conventions held across all 10 files.

Claude was the primary drafting tool. Each component followed the same workflow: define purpose and scope, draft, then run a human review pass before it entered the kit. At three checkpoints (after the core workflow documents were complete, after the Excel tools were built, and before final packaging), a three-model audit was run: Claude as primary, with Gemini and ChatGPT reviewing for factual accuracy, internal consistency, and completeness against the scoped requirements.

Version control was maintained across 13 document revisions. Structural errors and corrections were tracked explicitly instead of being silently overwritten. That kept the build history reviewable and reduced regressions between revision cycles.

Quality Controls

Three realistic test cases validated the full workflow. Each one covers a different dispute scenario and runs a different lane through the system: a straightforward fraud claim on Stripe, a product-not-received dispute on Shopify Payments, and a subscription cancellation dispute on PayPal.

Those tests confirm the Calculator produces the right recommendation, the Tracker captures the right fields, the Evidence Assembly Guide points to the correct platform locations, and the Templates match the reason code.

Every component was tested against the product boundary: this is an operational toolkit, not legal advice, not a SaaS product, and not a guarantee of dispute outcomes. That boundary is documented in the product copy and enforced in the language of every file in the kit.

What This Demonstrates

This project demonstrates the skill set required for Operations Specialist, AI Operations, and Revenue Operations roles:

Workflow design - A 10-component system where each file has a defined role and the sequence is intentional. The workflow moves from decision (Calculator) through execution (Evidence Assembly, Templates) to tracking (Tracker) to prevention (Prevention Mini-Pack).

AI-assisted production with QA discipline - AI tools were used to move fast. Human review and multi-model audits were used to maintain accuracy. The speed is notable, but the quality controls maintained at that speed are the point.

Documentation as an operational asset - Every component is documented to the level where someone unfamiliar with the domain can pick it up and execute.

Scope discipline - The project shipped what was scoped, on time, with no feature creep. Requirements were defined before the build started and held throughout.

Speed to value under constraints - 9 days from conception to live product, including 6 days of active build. It's a live product, not a prototype or a demo.