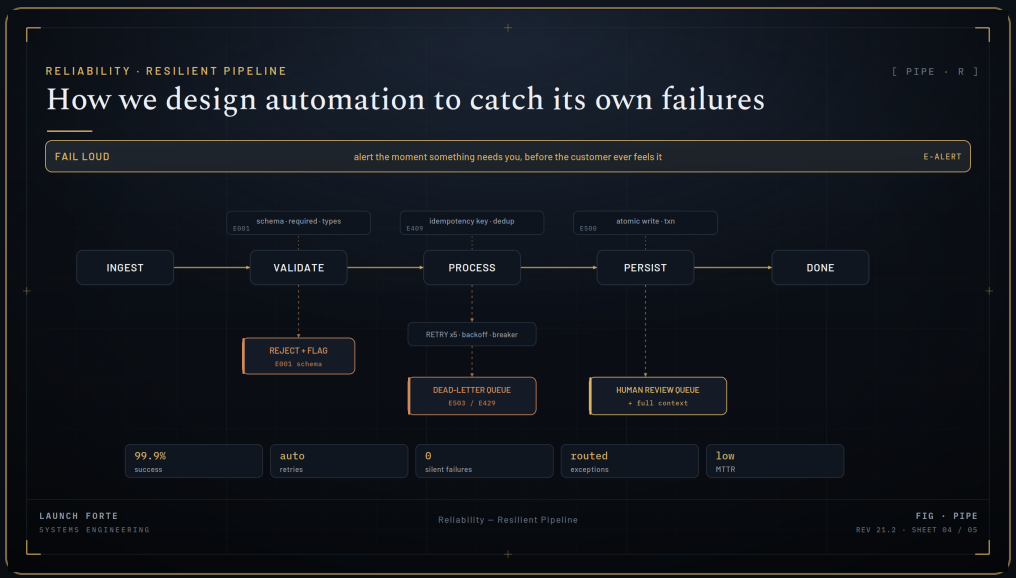
FREE GUIDE · LAUNCH FORTE

Automation That Catches Its Own Failures

Why most automation dies the first hard week, and how to build the kind that holds.



THE FRAMEWORK

The same framework we apply inside a paid build.
The 80 percent that demos never show.

01 · THE PREMISE

Most automation dies the first hard week

Most automation works great in the demo and dies the first hard week. Someone wires two tools together, it runs clean on the happy path, and then a piece of data shows up missing, a payment fails, or a duplicate sneaks in, and the whole thing breaks without a sound.

The happy path is maybe twenty percent of a real system. The other eighty percent is everything that happens when reality does not cooperate, and that eighty percent is the entire reason the system survives. It does not show up in a demo, which is exactly why most builds skip it.

> A silent failure is worse than no automation.
> You trusted it, and it cost you anyway.

HOW TO READ THIS

The next page walks the five design choices that separate automation that holds from automation that falls over. Then a short method for pressure testing your own.

02 · THE DESIGN

Five choices that make it hold

01

Validate the input

Check the data is present and well formed before anything runs.

Where it breaks: If it is not valid, the flow stops and flags exactly what is wrong, instead of passing the problem downstream.

02

Retry with backoff

When a service is down, slow, or rate limited, retry a few times, spaced out.

Where it breaks: If it still cannot get through, the record is held safely and flagged, never dropped.

03

Deduplicate

Match on a stable key before anything downstream runs.

Where it breaks: Duplicates are one of the most common ways a clean system poisons itself. No double charges, no double sends.

04

Route to a human

When the system hits something it cannot resolve, it does not guess.

Where it breaks: It routes the case to a person with full context attached, so they spend thirty seconds on the real exception instead of hunting for what broke.

05

Fail loud

The moment something needs attention, an alert goes out before the customer feels it.

Where it breaks: The opposite of the silent failure that looks fine for two weeks while it quietly does the wrong thing.

03 · THE METHOD

How to know if your automation will hold

Before you trust an automation to run unattended, design for the failure first. Ask what happens at each step when the input is missing, the service is down, or the same record comes through twice. Build the answer to those before you build the part that works.

Then make it fail loud. A system that tells you the moment it needs you is one you can actually leave alone. A system that fails in silence is one you will be cleaning up after for weeks.

THREE QUESTIONS TO ASK FIRST

01 What happens when the data is missing or malformed at each step?

02 When an external service fails, is the record retried, held, or lost?

03 When something needs a human, does the system alert, and with what context?

The happy path is the easy twenty percent. Designing for everything that goes wrong is the job, and it is the difference between a system that demos well and one still running a year later.

NEXT STEP

Want the full diagnosis?

This guide shows the pattern. A Growth Systems Audit applies it to your business. A written diagnosis of what is worth fixing and what it is costing you, prioritized, with fixed pricing on every fix and credits toward your first build.

Book a Growth Systems Audit

calendly.com/seth-launchforte
seth@launchforte.com

Launch Forte
Engineers that speak revenue. Based in Dallas-Fort Worth, serving clients nationwide.