Hey Walter

CASE STUDY · CUSTOM MANUFACTURING

One app for every job, from RFQ to shipped.

Pacific Enclosures builds sound-attenuated outdoor enclosures for diesel and gas generator sets, and every one is custom. We built Pacific Enclosures OS, the one app their estimators, purchasers, receiving and shop floor now share.

CLIENT	LIVE SINCE	OLD PO SYSTEM RETIRED	BUILT ON
Pacific Enclosures Ltd.	October 19, 2026	November 20, 2026	Chipp, with Claude Code

185
releases in the first five weeks live

278
purchase orders moved over at cutover

3,203
automated tests run before every deploy

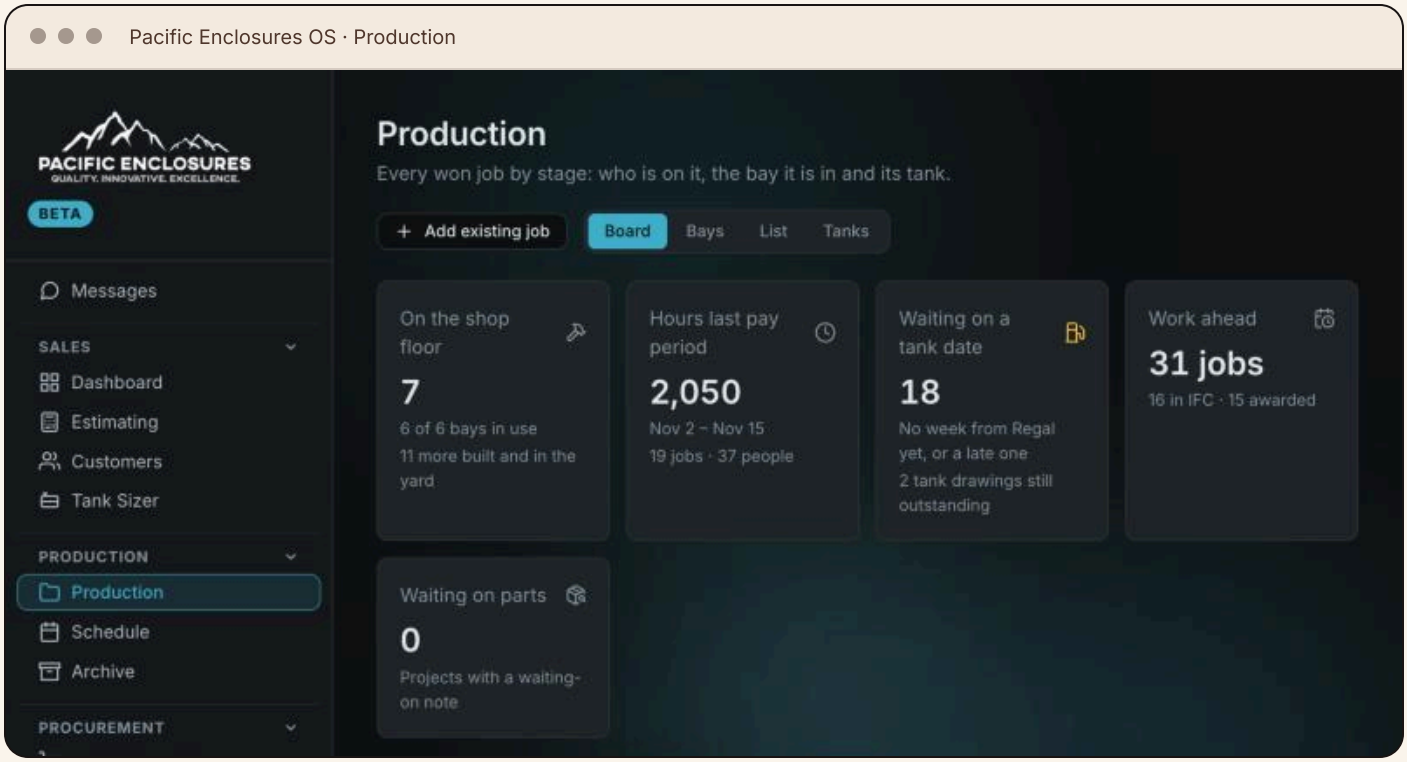
1 hour
from a receiver's request to a live fix

THE CHALLENGE

Every part of a job lived in its own tool.

None of those tools talked to each other, so the people doing the work carried the job between them.

- **Estimating.** The engineer priced each job in a master Excel workbook. A separate Python estimator ran beside it, and quote letters were typed in Word.
- **Receiving.** Deliveries were checked against paper slips, then handwritten, scanned and filed.
- **Job costing.** Actuals lived in Sage reports and timesheet spreadsheets kept in Box job folders.
- **Purchasing.** POs were raised in a separate app that didn't know which job they belonged to. With the lead purchaser off-site, nobody could easily see what had already been ordered.
- **Shop floor.** The shop foreman did a weekly paper bay check to record where every job sat, and couldn't see the POs placed on jobs in the shop.



Today, Production shows every won job by stage, with the bays in use, last pay period's hours and the work ahead.

WHAT WE BUILT

Four stages, one job record.

A job moves from estimate to quote to award to production. Every module reads and writes the same record, so a PO, a delivery or a message always lands on the right job.



Also in the app: a dashboard of pipeline, booked and lost work and win rate, an archive of finished jobs, tank orders, and an AI assistant that answers questions on live data.

UNDER THE HOOD

The AI reads and drafts. Every number comes from the estimate.

The RFQ reader

Claude reads the customer's RFQ and fills in the estimate form. Every value quotes the passage it came from, anything uncertain starts unticked, and it never fills in a price.

The estimating engine

A port of the engineer's Excel workbook. Automated parity tests prove it prices the same, and the engineer's version only saves once it matches the workbook cell for cell.

Receiving from a photo

A photo of the packing slip fills in the delivery, line by line, for the receiver to confirm.

One currency for money

Canadian dollars are the single source of truth. US dollars are shown at the quote's exchange rate, never stored as a second figure.

Safety rails

Quote revisions can't be changed once made, awarding a quote opens exactly one project, and every delete asks for confirmation.

The stack

A Deno and Hono API, a Svelte 5 web app that installs on a phone, PostgreSQL and Redis, deployed on Chipp. Every deploy passes type checks, lint, the full test suite and the web build.

ON THE SHOP FLOOR

Receiving moved from paper to a phone.

The receiver finds a PO by number, vendor, job or part, takes a photo of the packing slip, and records where each line was put away. Back orders stay open until the rest arrives.

When something goes wrong, the fix is quick. Here is one morning, November 23:

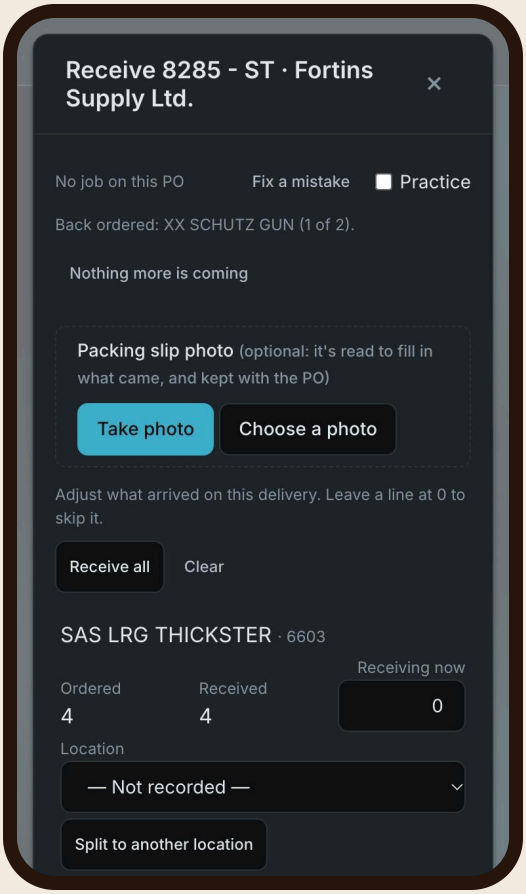
9:50 AM
The request reaches us: a delivery was logged as 2 paint guns when 1 came. How do you undo it?

BEFORE 11 AM
"Fix a mistake" is live. Anyone who receives can correct a count, and each correction keeps their name and reason.

LATER THAT DAY
The receiver fixes the PO. The second gun goes back on back order, and the original receipt stays in the log.



Search finds the PO in a few keystrokes.



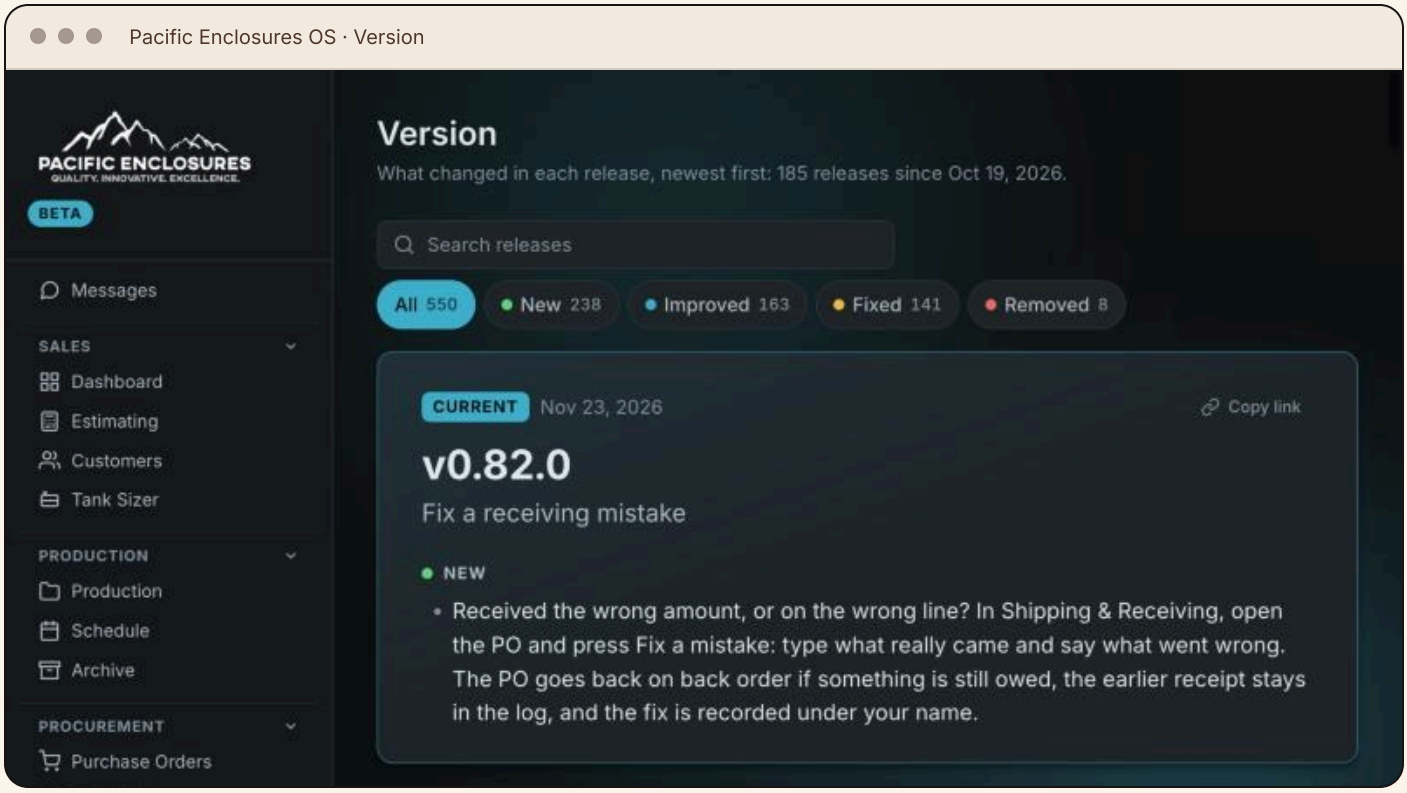
The same PO after the fix: one gun back ordered.

HOW WE WORKED

Small releases, straight from the team's own words.

185 releases went out in the first 36 days, and every one passed the full test suite before it deployed.

- **A written spec, kept in the code.** Every pricing formula and business rule lives in a functional spec beside the code. When the two disagree, the spec wins.
- **Decisions stay with the owner.** When a request could be read two ways, we asked with both readings worked out, and recorded the answer.
- **Built with the people who use it.** On November 4 we sat down with receiving, the shop foreman and purchasing. Thirteen releases went out that day.
- **A careful cutover.** The old PO system's data was copied read-only and nothing in it was deleted. On November 20 its 278 POs moved over, and the old address now points to the new app.



Every release is listed in the app in plain words, newest first.

RESULTS

Every role works from the same job record.

The whole team also shares Messages: a channel per job, direct messages, notifications on their phones and a morning brief built from each person's work.

ROLE	BEFORE	NOW
Estimating	Excel workbook, Word quote letters	Start from the RFQ, which Claude reads. Estimate on the workbook's formulas. Branded quote PDF with a full revision history.
Purchasing	A separate PO app, not tied to jobs	POs filed on their job. Purchasing leads told when anyone else places one. 308 POs live.
Shipping and receiving	Paper slips, handwritten, scanned and filed	Receive on a phone from a photo of the packing slip. A storage location per line, back orders, and a way to fix a mistake.
Shop floor	A weekly paper bay check	A production board by bay, days spent in each location, and PO and receiving updates in each job's channel.
Job costing	Sage reports and timesheets in Box folders	Actuals read from those same files onto each job.

Build it. Tend it. Own it.

Human at the center. AI at the edges.

hello@heywalter.dev · heywalter.dev

© 2026 FUTURE READY HOLDINGS INC. ALL RIGHTS RESERVED.