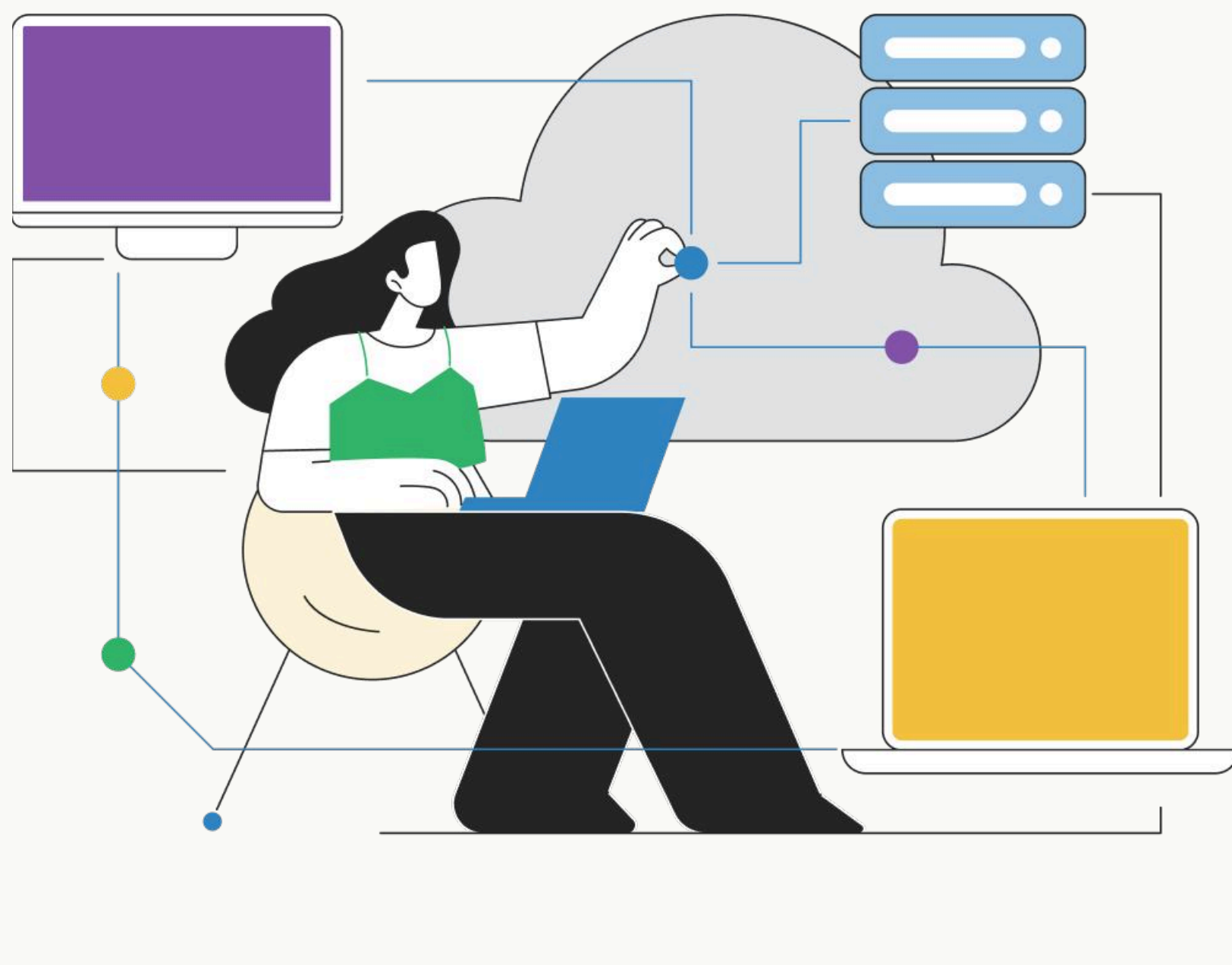


# From Chaos to Cockpit

Redesigning a high-stakes AR Claims Portal to eliminate decision fatigue and recover revenue faster.



**DOMAIN**  
Revenue Cycle Management

**ROLE**  
Sole UI and UX Designer

**TIMELINE**  
3 Weeks

**IMPACT**  
20-30% reduction in per-claim processing time · Improved closure rates on high-value claims

## Problem Context

**AR Executive 01**  
"I spend 4 hours a day just figuring out which 40 claims actually need attention. By the time I start working, I'm already exhausted."

**AR Executive 02**  
"I spend half my morning just logging into various payer portals. By the time I start working a claim, I'm already mentally drained."

**JOBS TO BE DONE**

- "Help me pick up where I left off"
- "Help me not forget to call back"
- "Help me prioritise high risk value ticket"
- "Help me prioritise low risk value ticket also"
- "Help me not to switch back and forth with multiple tabs"

### Business Reality

Healthcare payments in the US and UK lag behind treatment, creating a revenue risk window between discharge and reimbursement. This gap is managed by Accounts Receivable (AR) teams.

AR executives handle claims post-submission—following up, resolving issues, and securing payments. Each executive processes ~30 claims daily, requiring review, outreach, and decisions.

The challenge is straightforward: slower claim closures delay revenue, and even small lags at scale significantly impact cash flow.

**Constraints**

- High volume, low margin for error:** 30 claims/day is a baseline. Falling short has direct revenue consequences.
- Time-bound claims:** Insurance claims have aging windows. Crossing thresholds makes recovery significantly harder.
- Compliance sensitivity:** Every interaction is an audit touchpoint. Outcomes must be logged with clinical precision.

## Research and Insights

**Stakeholders**

- How do current recovery targets align with the long-term margins?
- What are the 'unacceptable' audit risks in the current manual workflow?
- Define the ROI threshold for automating L1 claim responses.

**Managers**

- How do you redistribute workload when a specialist is out?
- At what point does a claim become 'stale' enough to escalate?
- What data is currently missing from your team throughput dashboards?

**AR Executives**

- Walk me through the first 30 minutes of your shift.
- If you could delete one tool from your desktop, which one and why?
- 'What-if' scenario: A \$50k claim is denied; what is your first action?

### What We Observed

User interviews and shadowing revealed a fragmented system held together by workarounds and tribal knowledge.

AR executives worked across 4-6 disconnected tabs with no integration or structured prioritization—decisions were largely intuition-driven. Notes were unstructured, making claim handovers slow, with 3-5 minutes spent just on context building.

Callback tracking was managed in separate spreadsheets, leading to missed follow-ups, rework, and delayed claim closures.

PAIN POINT	ESTIMATED IMPACT
Context reconstruction per claim	~4 min lost per handoff
Missed follow-ups per user)	3-5 callbacks dropped
Time lost to tab-switching	~2 min average
High-value claims delayed	15-20% of \$1,000+ claims

### Quantified Pain & Insight

**Lack of direction, not data:** Initial assumptions suggested a data gap, but research revealed a deeper issue lack of direction, not data.

**What to tackle first:** AR executives had all the information they needed, but no guidance on what to tackle first or how to act.

**information display to decision orchestration:** This reframed the design challenge from information display to decision orchestration, shaping every design choice that followed.

### Secondary Insight – Ethics vs. Revenue:

Prioritizing claims purely by revenue risked neglecting small-dollar claims, which could have real human impact.

The system needed to **balance business goals** with **patient fairness**, navigating a tension between revenue optimization and ethical responsibility.



## Strategy & Approach

### Ticket Value with Aging

The core strategic challenge was building a prioritization engine that balanced two variables: **Ticket Size (\$)** and **Aging (Days Pending)**.

### Patient Fairness

We explicitly encoded **"patient fairness"** as a system constraint. This meant occasionally interrupting the revenue-optimal workflow to surface a \$75 claim that had been sitting for 45 days.

Most enterprise systems don't do this.



## Alternatives

APPROACH	WHY REJECTED
Pure revenue-sort	Creates ethical risk; small claims age out.
User-defined	Adds cognitive burden; inconsistent outcomes.
Separate queues	Siloed workflows; doesn't scale.

### Pure Revenue-Based Sorting

Prioritizing only high-value claims maximized short-term returns but created a neglected long tail of smaller claims.

### Aging-Only Queue

Sorting purely by days pending improved fairness but ignored revenue impact and business priorities.

### Static Weighted Scoring Model

A single score (value + aging) lacked transparency and flexibility, making it hard for users to trust or interpret prioritization.

### Manual Prioritization Controls

Giving users full control increased cognitive load and led to inconsistent, non-scalable decision-making.

## Design Strategy

**EXECUTION: THE HIGH-FIDELITY SINGLE-SCREEN COCKPIT**

### Design Philosophy: Air Traffic Controller, Not Analyst

The strategy was guided by a single analogy: the interface should function like an air traffic controller's workstation, not a data analyst's dashboard. The goal was to reduce cognitive load and guide decisions, not just display data.

### Execution: The Single-Screen Cockpit

The most significant choice was eliminating page navigation. Users could review details, call insurers, update notes, and return to the worklist seamlessly — all from one screen.

**BEFORE: "THE TAB CHAOS"**

PAYER PORTAL, CLAIM NOTES

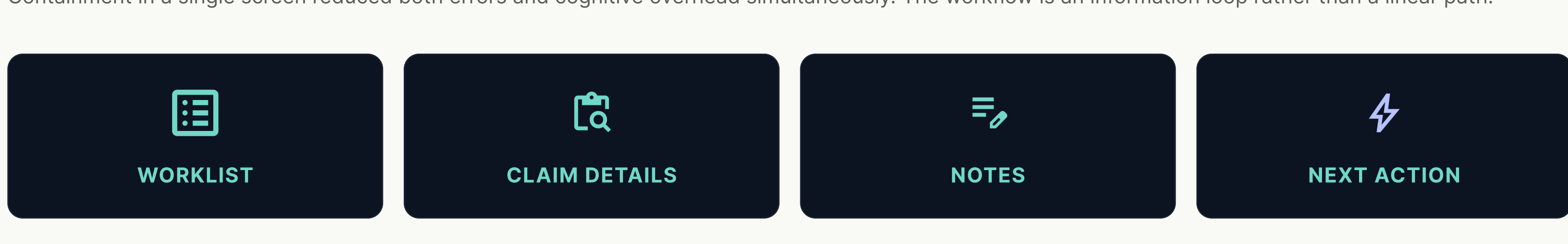
Information fragmentation leads to context switching fatigue.

**AFTER: "THE COCKPIT"**

Unified environment reduces mental reload time by 40%.

## The Session Workflow Loop

Containment in a single screen reduced both errors and cognitive overhead simultaneously. The workflow is an information loop rather than a linear path.



## Results & Impact

The impact was measured over a 90-day period following team rollout. Accelerated closure of high-value claims directly reduced the average days outstanding (ADO).

Reframing the problem unlocked the solution. The shift from **"information design"** to **"decision design"** wasn't cosmetic — it fundamentally changed what success looked like.

**Constraints as features.** The forced "Next Action" field is the best example of productive friction. Well-placed guardrails improve quality without reducing autonomy.

↓ 28% PROCESS TIME

↓ 80% CALLBACKS

↑ 15% CLAIM YIELD

0% CONTEXT LOSS

Thank You