

ENTERPRISE PRODUCT · CASE STORY

The engine was never the problem

Modernizing a scheduler big companies trust with their reporting, without breaking what already worked.

4 min

Click or press → to continue

Imagine a world-class kitchen with its front door in five different alleys.

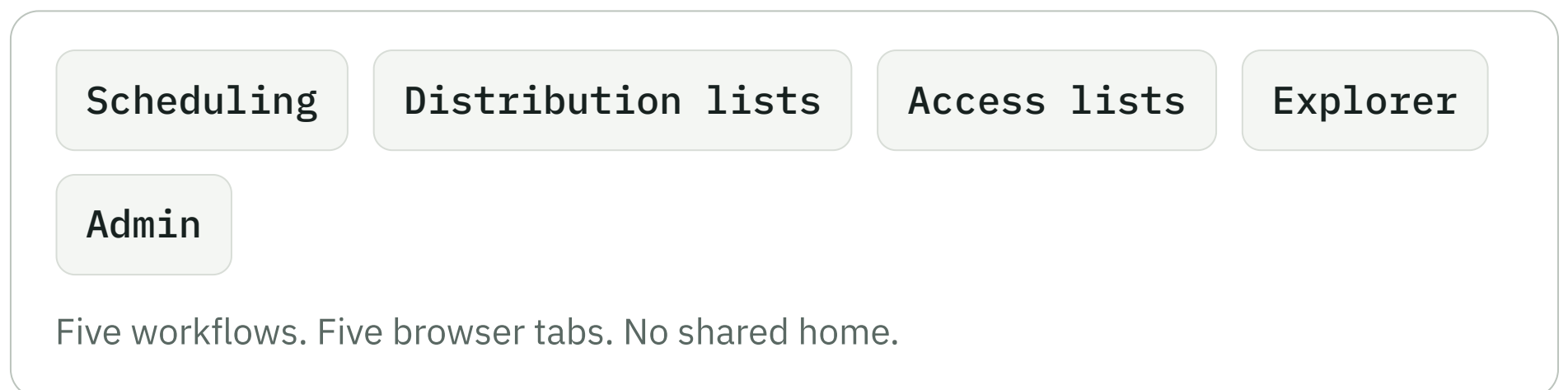
The food is excellent. Getting to it is the problem.

ReportCaster is the scheduling and distribution engine behind WebFOCUS. It was so well built that it ran for decades on fixes alone.

It runs millions of automated jobs a week for large enterprise customers.

If ReportCaster stops, their reporting stops.

Meet the Reporting Guru, who lives in it all day, and the BI Developer, who touches it now and then.



The engine was trusted. The experience around it hadn't moved in decades. A five-year, multi-million-dollar account was ready to leave.

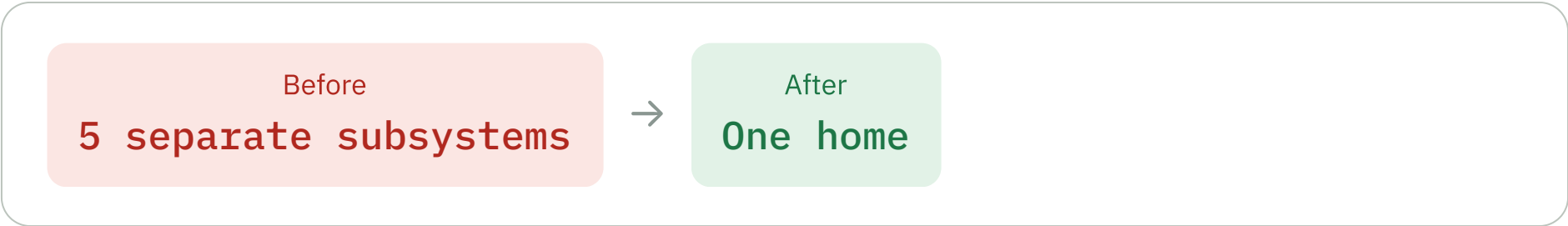
Rule 1

Understand it before you redesign it.

- ✓ Decades of undocumented logic, reverse-engineered by hand from a single sandbox link
- ✓ Hundreds of screenshots
- ✓ Long sessions with Customer Support, the only real source of truth
- ✓ Conversations with the engineer who built it decades earlier

Rule 2

A structural redesign, not a facelift.



New paint on five rooms would still leave five rooms.

Rule 3

Let each "no" narrow the answer.

- ✗ **Version 1: a standalone product.** Leadership wanted every workflow in the platform's hub. A fair constraint, not a design flaw.
- ✗ **Version 2: a hub plugin.** Too much engineering for one release.
- ✓ **Version 3: built into the platform itself.**

Instead of fighting the decisions, the problem got reframed from scratch.

Rule 4

Ask how the platform wants to behave.

Not "where should it live?" Instead: where does every creation flow already start?

The "+" menu

Create Visualization

Create Data Flow

Create Schedule

Create Distribution List

Create Access List

If ReportCaster is fundamentally a creation workflow, it should start there too. The "+" menu wasn't UI. It was architecture.

Rule 5

Follow the platform's own conventions.

SIDE PANEL

Built for read-only file properties. Never for multi-step work.

MODAL

Matches the platform, keeps people in context, and removes the old basic-vs-advanced split.

● PRINCIPLE

Consistency and standards

People shouldn't have to wonder whether different words or layouts mean the same thing. Follow the platform's conventions.¹

Rule 6

One pattern for every asset.

Schedule

Distribution list

Access list

Same structure, same controls. Learn one and you've learned all three.

Rule 7

Design so the next ten releases don't need a redesign.

Delivery methods at launch

5



Later, same modal

8

The product kept growing on the structure, not around it.

WHAT CHANGED

Shipped in April 2024.

- ✓ Five subsystems into one home · 250+ screens
- ✓ Clicks to reach Explorer: 4 → 1. Scheduler, access and distribution lists: 4 → 2. Admin: 3 → 2
- ✓ The account stayed: a five-year contract worth millions, signed after it shipped

End-to-end clicks still vary by configuration, so there's no single number for the whole workflow.

BEHIND THE STORY

I asked to lead it in my first week.

Its redesign had waited years. One week into the job, I asked to lead it, and it was handed to me the same day.

The engineer who built ReportCaster decades ago became my partner in rebuilding it:

"From the start, she impressed everyone with how quickly she grasped all aspects of a highly intricate system and translated that understanding into a clear, modern, and user-centered design."

Yingchun Chen, Principal System Software Engineer

ReportCaster was never a UI project. It was a decision, an architecture, and a negotiation.

CHEAT SHEET

Modernizing software people depend on

- 1 **Understand it first.** The logic is often undocumented.
- 2 **Fix the structure,** not just the surface.
- 3 **Let each "no" narrow the answer.**
- 4 **Ask how the platform wants to behave.**
- 5 **Follow its conventions.**
- 6 **One pattern,** learned once.
- 7 **Build to extend,** not just to ship.

READ MORE

The full case study.

The research, the three versions, and the details of the redesign.

Full case study: anujaharsha.com/work/reportcaster

Enterprise story 01: "The engine was never the problem." Designed by Anuja Nimmagadda for WebFOCUS at Cloud Software Group.