

THE REAL COST OF "WE'LL UPDATE IT LATER"

Real Time Isn't a Feature. It's the Whole Schedule.

Three small handoffs you've stopped noticing. Multiply them across 40 days and a few hundred jobs, and they're the reason turnarounds run long.

Every turnaround leader will tell you the same thing: the schedule never slips all at once. It slips in **minutes** — a delayed alert here, an unreported job there, a cost report that's a day behind. None of it looks expensive in the moment. Stacked across a 40-day event, it's the difference between closing on time and bleeding billable contractors for an extra week.

The root cause is almost never effort. Your people work hard. The problem is that the information they work from is **old by the time they get it**. Here's what that actually looks like on the ground.

1. Material arrives. Now what?

A DELIVERY SHOULDN'T NEED A RELAY RACE

● THE SPREADSHEET WAY

1. Material hits the dock.
2. Someone waits for the ERP alert.
3. They update a spreadsheet.
4. They message someone in the field.
5. That person notifies the contractor.
6. The contractor sends a crew to the warehouse.

Six handoffs. Hours — sometimes a shift — of crews idle, waiting on a part that's already on site.

● THE REAL-TIME WAY

1. Material hits the dock and is received once.
2. **Every party is notified instantly** — field, contractor, planner.
3. The crew is moving before the old process had finished step two.

One event. Everyone who needs to know, knows — at the same moment.

2. The job that "isn't done yet"

SANDBAGGING, AND WHY IT WRECKS QA

A contractor has five jobs planned for the day. They finish eight. But they won't report eight — they'll report five, and quietly bank the other three for a slow day later, so their numbers always look on track.

It feels harmless. It isn't. **Your quality assurance team is now planning labor and resources around information that's deliberately wrong**. For most of the turnaround it looks like the crews are behind — so QA staffs to that picture. Then, near the end, every banked job surfaces at once and the work "miraculously" catches up.

Now QA can't possibly close out everything in time — and while they scramble, every contractor stays on site, billable, in case rework comes back. That backlog at the finish line is what drags turnarounds past schedule.

Look at how a completed job normally travels: the field reports to supervision → supervision updates a checklist, a logbook, an email, or pings the scheduler → that eventually reaches QA — *if* it gets reported on time at all. Every arrow is a delay and a place for a job to get banked.

● REPORTED UP THE CHAIN

1. Field tells supervision.
2. Supervision logs it somewhere — maybe.
3. It's relayed to the scheduler or QA.
4. QA finds out late, plans on bad data.

QA is always reacting to yesterday.

● SCANNED IN THE FIELD

1. A third party walks the field and scans jobs complete.
2. The moment it's marked done, **all parties are notified**.
3. QA paces its resources against what's actually finished — every day.

No banking. No surprise backlog. No idle billable crews at the end.

3. Cost & headcount — a day behind, every day

COMPILING ISN'T REPORTING

In the usual flow, vendors submit cost and headcount, then your cost team compiles all of it, builds the reports, and only then tells leadership what was actually spent and who was on site. By the time the number lands, it describes **yesterday**. On a fast-moving turnaround, decisions made on day-old spend data are decisions made blind.

In real time, the vendor's own update *is* the report. Cost and headcount post the moment they're entered — no compiling, no lag, no analyst army rebuilding the same picture every morning.

And here's the part most teams miss: Power BI doesn't fix this. If your contractors, inspection, operations, and cost teams still live in separate spreadsheets, someone is still updating files and feeding them into the dashboard. A beautiful dashboard built on yesterday's spreadsheets is still yesterday's data.

Real time only exists when **every player on the turnaround — contractors, company inspection, operations, planning, cost, and the owner — works inside one connected tool**. Anything short of that is a relay race wearing a dashboard.

So what does it actually cost?

AN ILLUSTRATION — NOT A QUOTE

Take a \$100M turnaround over roughly 40 days of maintenance. Say half of that spend lands inside those 40 days. That alone implies a burn rate north of **\$1M a day** while the event is live. Now layer the three delays above across the full run — cost reporting lagging every day, materials waiting on handoffs, and a few hundred jobs whose true status QA learns late.

BACK-OF-ENVELOPE — ILLUSTRATIVE ONLY

| | |
|---|------------------------|
| Turnaround value | \$100M |
| Spend inside the 40-day window (≈ half) | ~\$50M |
| Implied burn while live | > \$1M / day |
| Days every report runs a step behind | -40-50 |
| Jobs whose real status arrives late | 200-300 |

Don't hold us to the exact figures — the point isn't the number, it's the order of magnitude. When a single late day costs seven figures, shaving even a fraction off the delay on every report, every delivery, and every job completion adds up to real money, fast.

That's the honest case for real time. Not a flashy feature list — just the quiet arithmetic of removing the lag from every handoff, on every day, for every player on the event.

SEE YOUR TURNAROUND IN REAL TIME

One connected tool for every player — contractors, inspection, operations, planning, and cost. A 30-minute walkthrough shows exactly where the lag is hiding in your current process.

[Book a 30-min demo →](#)