REGIONAL TELECOM OPERATOR · 5G INFRASTRUCTURE

Streamlining 5G integrations.

How I helped a cell-tower operator cut more than 10,000 labor hours a year out of their 5G rollout — by measuring the waste, killing the bottlenecks, and moving the slow work off-site.

CLIENT	ENGAGEMENT	PUBLISHED
Tower-ops leaders (anonymous)	Field ops, process, field analytics	May 2023

THE CHALLENGE

Integrations that lived and died on the tower.

Installing new 5G equipment on-site was a time-intensive process, regularly stalled by unpredictable software updates. A typical site carried three to four modules, and each one needed updates that often had to be chained in phases — legacy first, then latest.

They also failed at random. When that happened, the technician started over.

The ripple effects were expensive.

SYMPTOMS

- Delayed cut-over times, site after site
- Technicians idle at the top of the tower
- Foremen on the ground waiting to sign off
- Labor waste compounding across multi-site operations

PER-MODULE UPDATE

20–35 min

Typical time to push updates to a single module, before any failure or retry.

MODULES PER SITE

3–4

Each one on its own timer, each one waiting on the last.

CREW STANDBY

Idle time

Tower techs and foremen paid to wait on a software update that might fail anyway.

WHY I GOT THE CALL

Data over decks.

The operators I was working with didn't want another consultant to describe their problem back to them in a slide deck. They wanted someone on the ground who would measure the waste, call it out, and fix it.

That's what I do. I work fast, I communicate directly, and I bring the numbers.

01

Responsive communication

Same-day replies. Clear updates from the field — not monthly readouts.

02

A data-driven approach

I measure before I suggest. Every process change had to show up in the numbers.

03

Time-on-site analytics

I offered to instrument the work itself — so waste stopped being a story and started being a number.

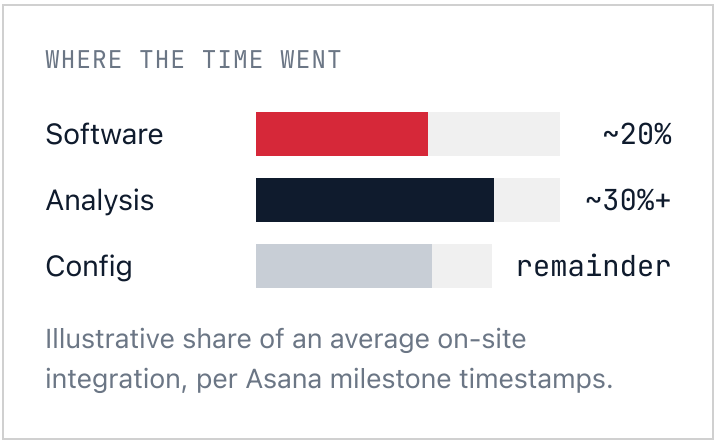
> *A quick note from me — most consultants spend more time making presentations about the work than doing it. I cut the bloat and execute on day one.*

PHASE 01

Measure the waste.

Before changing anything, I needed to see where the time was actually going. I set up milestone-based timestamping in Asana with built-in time tracking, so every integration logged the moments that mattered.

- Integration start and end
- Technician-reported blockers
- Every software update, from kickoff to done



What the data said

Two things dominated. Troubleshooting was bigger, but it varied with site config, hardware, and technician skill — too many confounds to fix cleanly.

Software updates were different. Consistent, repeated, solvable with process alone. That's where I went first.

TARGET

Software updates, first.

Solvable with process changes. No new hardware dependency, no crew retraining.

PHASE 02 — THE OBVIOUS WIN

Stop updating one module at a time.

The easiest bottleneck to kill was also the most embarrassing: the integrators were running updates one-to-one because they each had one laptop. I bought every integrator an extra laptop.

That's it. That was the first change. It cut roughly 45 minutes of waste per integration.

UPFRONT COST

$3,600

$1,200 × 3 integrators — additional laptops so updates run in parallel.

LABOR HOURS SAVED / YEAR

292.5

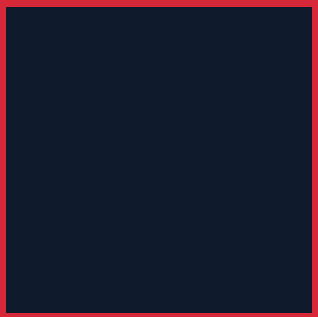
Based on 2.5 integrations per week across 3 integrators, ~45 min saved each.

ANNUAL SAVINGS

$13,006

Midwest telecom integrator salary basis. Does not include tower-crew time saved.

Worth saying out loud: the cheapest fix paid for itself in under three months, and it only came into view because the data from Phase 01 made the waste legible. You can't fix what you haven't measured.



PHASE 03

Catch failing updates in the first five minutes — not the twentieth.

Random failures were the next line in the Asana data. They usually surfaced past the fifteen-minute mark, which meant the technician had already burned most of an update cycle by the time the thing gave up.

I opened Chrome's developer tools and started reading the console during updates. The pattern came out fast.

- Specific console errors consistently predicted a failing update
- Same error signatures on every failed run
- Surfaced in the first 3–5 minutes — long before the update bailed

Short process change: when those errors appear, interrupt and restart. No more twenty-minute dead runs.

RESULT

19%

Reduction in update failure rate. In continued Asana tracking, software-update time dropped from ~20% of the integration down to under 1%.

PHASE 04 — THE BIGGEST LEVER

Move the slow work off the tower.

The largest savings weren't on-site at all. Once I understood the waste, the question was obvious: why are we doing updates on a tower in the first place?

We didn't have to. With the right power and shelving in a warehouse, I could stage modules, update them in bulk, and ship them to the site already configured.

- Updating 10+ modules simultaneously off-site
- Integrators multitasking across parallel bulk runs
- Modules arriving on-site pre-updated and pre-configured
- No more waiting around on the tower for new-module installs

IMPORTANT NOTE

Not every site needs new equipment — reused gear still has to be updated on-tower. The parallel-update and error-monitoring wins from Phases 02 and 03 still apply to those sites.

PHASE 01-03

Running total

Integrator labor only. Tower crew hours not included.

PARALLEL UPDATES	PREVENTED FAILURES	COMBINED / YR
$13,006	$13,873	$26,879

FINAL IMPACT

What eliminating software waste actually did.

1–2 hrs

Labor saved per integration — 5–10 hours per crew, per week.

3–8 hrs

Tower-crew wait time reduced, every site.

10K+

Labor hours saved annually across the client roster.

Crew efficiency up. Field safety up. Delivery timelines tighter. Operational cost down. All from process innovation and real-time field analytics — no new vendors, no re-platforming, no six-month transformation program.

CONCLUSION

Operational insight plus technical hands-on work. At the scale of a 5G rollout, that's millions in labor savings over time.

HAVE A SIMILAR PROBLEM?

Send me a message. We'll know in one call if this fits.

EMAIL DIRECT

contact@corey.consulting

COREY.CONSULTING