

One More Thing: When Scope Creep Becomes Scope Leap

Welcome to InCogSights™, Issue Two.

The original statement of work included one sentence about reports. Steric Hindrance Co. expects to require COA reports to be developed and sent with each shipment.

On the surface that seems like a reasonable, well-scoped line item. Standard report, standard trigger, done. But a few weeks into discovery, we learned Steric Hindrance Co. had hundreds of COA formats due to various product/grade combinations, coupled with a meaningful number of their larger accounts each requiring their own specific COA format. One report became dozens. And it wasn't just the number of reports that grew. Managing who got which format required its own logic, which meant additional coding and configuration of the LIMS to make it all possible.

The ripple effects didn't stop there. The timeline shifted. Professional service hours increased. And it pulled additional resources from both sides, ours and Steric Hindrance Co.'s. The project didn't creep off track. It jumped.

Scope creep versus scope leap

Most people know scope creep. It's the slow accumulation of small asks, one field here, one report tweak there, until you look up months later and realize the project quietly became something bigger than what everyone signed up for.

Scope leap is different. It's not slow and it's not an accumulation. It's a single request that jumps the project into a materially different thing, often disguised as something small. Scope creep erodes a project over time. Scope leap changes what the project is, in one sentence.

The reason scope leap is more dangerous isn't the size of the request. It's that it rarely announces itself. It shows up wearing the same casual language as every other small ask that came before it.

And the cost isn't only time and money. When a project jumps in scope without either side fully flagging what just happened, it puts strain on the vendor-customer relationship itself. Small disagreements start getting escalated rather than resolved in the room, which only slows things down further. The customer can start to feel like the project is spinning out of their control. The vendor can start to feel like the goalposts keep moving. Neither side set out to cause that, but without a shared moment of acknowledgment, trust erodes right alongside the budget and the timeline.

The tells

After thirty years of sitting in these conversations, from both the vendor and customer perspective, I've learned to hear scope leap before it fully lands. A few of the signals I listen for:

It changes who has to be in the room. A request that pulls in a new system owner, a new validation lead, or a new department that wasn't part of the original project team is rarely small, no matter how it's phrased.

It reopens something already signed off. If the ask touches a system, protocol, or process that already went through validation or approval, you're not adding to the project. You're reopening a decision that was already made.

It rides in on the coattails of something already approved. Phrases like "while we're in there" or "it's basically the same thing" are doing a lot of quiet work. They're borrowing the trust of the last approved item to smuggle in a new one.

It has no natural stopping point. A request for one more field is a request. A request that opens a category, one more integration, one more module, one more site, doesn't have an edge to it. If you can't define where it ends and what defines success, it's not a small addition. It's a significant change order that may not get properly scoped.

Why experience matters

None of these tells are obvious in the moment. Each one, taken alone, sounds like a reasonable ask from someone trying to get more value out of a project they're already paying for. That's exactly what makes them easy to miss if you haven't lived through the downstream cost of saying yes without noticing. It's also what makes a scope leap request look identical to a genuinely small one from the outside. The difference only becomes visible in hindsight, after the validation protocol is overdue, or a new stakeholder disagrees with a decision the original team already made.

That's the real value of having someone in the room who has watched this pattern play out before. Not to say no reflexively, but to recognize the moment for what it is while there's still time to make a deliberate decision instead of an accidental one.

The parking lot

The fix isn't necessarily a change order, though you should have a process to handle those too. The fix is simpler and harder at the same time: name the moment out loud when it happens.

"That's a great idea, and I want to flag this as a potential scope leap, not a small add. Let's talk through what it actually takes, and how it affects the project's goals, before we say yes."

That one sentence does most of the work. It doesn't shut the door on the request. It just makes sure everyone in the room is agreeing to what they're actually agreeing to.

If you've had a "one more thing" moment on a LIMS project that turned into something much bigger, I'd like to hear about it. And if you're about to embark on one and want an experienced eye watching for these moments before they happen, reach out.

Thanks for reading, and I'll see you in the next issue.

Matt

Steric Hindrance Co. is a fictional name, but the content is based on actual project experiences.