INTEGRATION

Who owns what happens between the workstreams

Finance is green. Supply chain is green. The order-to-cash process that runs across both is broken. Integration failure hides in the space no single workstream owns.

Roltrader Consultancy Group · 6 min read

Look at a typical programme status report and you will see workstreams: finance, supply chain, procurement, data, technical. Each has a lead, a plan, and a colour. What you will almost never see is a line for the thing the business actually experiences — the order that flows from a customer's request through availability, fulfilment, invoicing and cash. That process lives *across* the workstreams, in the seams between them, and the seams are precisely where no one's status report reaches.

This is why a programme can show a board full of green and still deliver a process that doesn't work end to end. Every team can be genuinely, honestly on track against its own scope while the handoffs between them quietly fail — because a handoff belongs to two teams, which is another way of saying it belongs to neither.

Integration is an accountability, not a connection

The word "integration" invites a category error. It sounds like a technical task — build the interface, map the fields, move the message — and so it gets handed to the technical workstream and treated as done when the data arrives intact. But moving data correctly is the small part. The hard part is agreeing what the data *means* on both sides, what happens when it's late or malformed, who is accountable when the end-to-end result is wrong even though every individual system did what it was told. That is not a connection to engineer. It is an accountability to assign.

> A handoff belongs to two teams — which is the same as belonging to neither. That is exactly where end-to-end processes break.

Green workstreams, red process

The mechanism is worth spelling out, because it fools experienced people. Workstream A delivers its scope: the order is captured correctly. Workstream B delivers its scope: the warehouse receives what it's sent. Neither is wrong. But the assumption A made about how the order would be structured is subtly different from what B was built to expect, and no one owned the space between the assumptions. The defect is real, the process is broken, and yet there is no red workstream to point at — because the failure lives in a gap that no plan drew a box around.

Draw the process, then assign the seams

The programmes that avoid this do one deceptively simple thing: they govern by end-to-end process, not only by workstream. They name the handful of business processes that actually matter — order to cash, procure to pay, record to report — and for each one they name an owner whose accountability

explicitly includes the seams. That owner cares about the whole flow being correct, not about any single system's scope being met. When the process breaks, there is someone whose job it was to see it coming, rather than five workstream leads each correctly pointing out that it wasn't their bit.

Test the seams, not just the systems

The same logic governs testing. Component tests confirm each system does its job; only end-to-end testing across the seams confirms the business process works — and that is exactly the testing most often compressed when the timeline tightens, because it is the hardest to coordinate and belongs to no single team. Cutting it doesn't remove the risk. It just moves the discovery of the broken seam from a test environment to a live one.

> **Where we sit.** Roltrader governs and assures by process as well as by workstream — Orchestra tracks the end-to-end flows and their owners, and our assurance layer checks the seams where systems reconcile against each other. But the habit is free to adopt: draw the processes that matter, put a name against each one's seams, and test across the joins before go-live decides it for you.