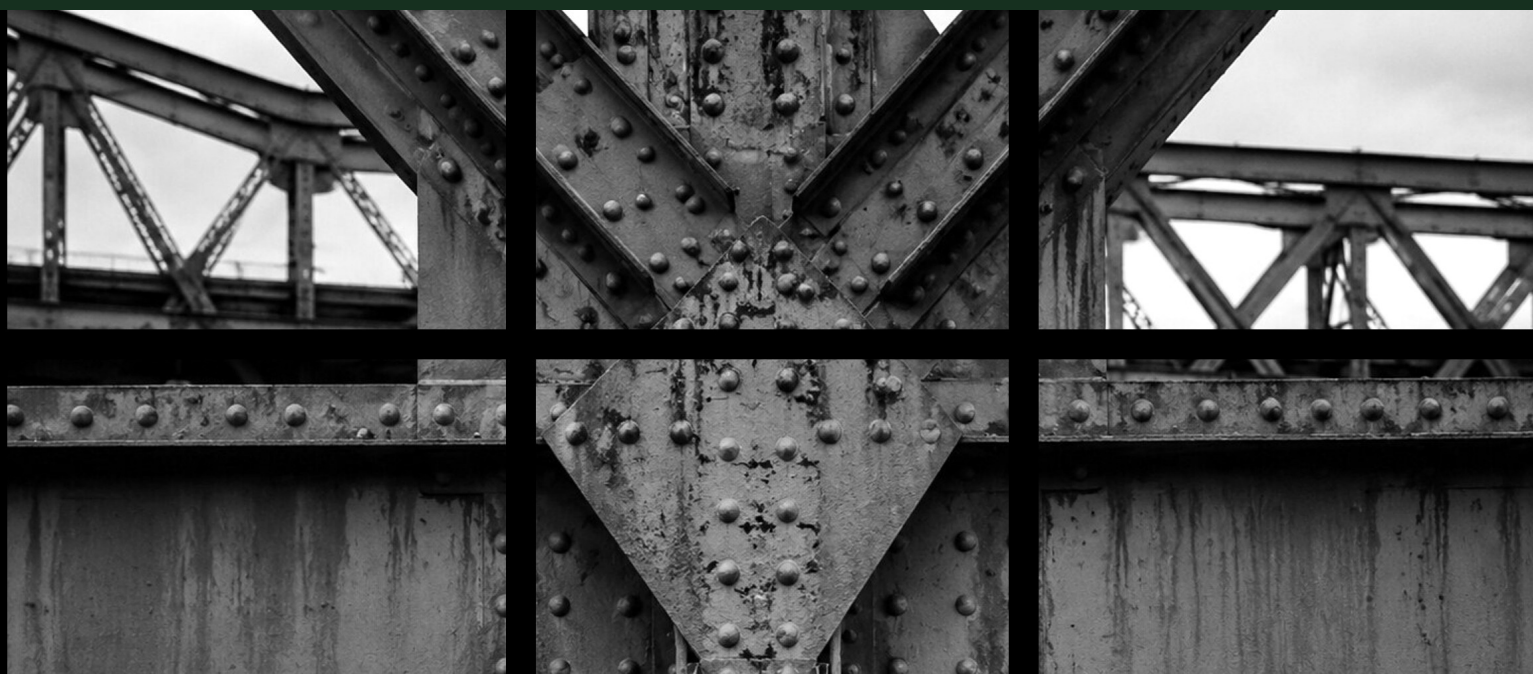


BEYOND BORROWED ASSURANCE

Three Lines of Defence, One Line of Delivery: Why a Banking Risk Model Doesn't Transfer to Programmes

The risk world's favourite governance model is migrating into transformation programmes. From the front line, the seams show.



Giovanni Leonardi

December 2007

Three Lines of Defence, One Line of Delivery: Why a Banking Risk Model Doesn't Transfer to Programmes

Programme · No. 117

Originally written December 2007 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“Programmes rarely die of un-caught risks. They die of un-made decisions, and a defence model has nothing to say about those.”

1. The defence that held while the programme died

A large delivery I have in mind — and I will keep it composite, because the pattern is what matters, not the organisation — was killed in its fourteenth month, and the striking thing was that its governance had worked exactly as designed. The delivery teams had owned their risks and controls. An independent programme assurance function had reviewed the plan quarterly and signed it. Internal audit had conducted its own review and raised no red flags. Three lines, each doing its job, each independent of the one below it. The perimeter held on every side. And the programme still walked off the cliff, because the thing that was wrong with it was not a control weakness that any of the three lines was built to catch. It was a decision that no one would take: two directorates wanted incompatible things from the same platform, and the governance model, for all its layered defence, contained no mechanism whose job was to force that choice. Every line was watching. None of them could decide.

I start here because the three-lines-of-defence model is quietly migrating out of the risk function, where it was born, and into the governance of transformation programmes, where it is increasingly offered as a template for how oversight should be arranged. The migration is understandable. The model is elegant, it is fashionable in the wake of the operational-risk disciplines that Basel II has pushed up every bank's agenda, and it gives a reassuring three-part answer to the anxious question of who is watching whom. My argument is that it does not transfer — that a model built to defend a stable operation against loss becomes something close to a liability when it is imported wholesale into the business of changing that operation. Borrowed assurance is not the same as fit-for-purpose governance, and from the front line the seams are easy to see.

2. What the model is, and where it comes from

It is worth being precise about the thing being borrowed. In its native habitat — the risk and control function of a bank or insurer — the three lines of defence arranges responsibility in depth. The first line is the business that owns and runs the operation and its risks day to day. The second is the risk and compliance functions that set the framework and challenge the first. The third is internal audit, independent of both, providing assurance to the board that the first two are working. It is a genuinely good idea for what it is for. It clarifies that owning a risk and assuring against it are different jobs that must not sit in the same hands, and after the operational-risk failures of recent years that separation has earned its place.

But notice what the model assumes, because the assumptions are the whole point. It assumes a standing operation — a business-as-usual that persists, that has a settled first line who own an ongoing activity, and against whose steady state the second and third lines can define controls. The model is defensive by design and by name: its purpose is to protect an existing thing from loss. Every one of those assumptions is either weak or simply false in a transformation programme, and that is where the transfer breaks.

3. Why it does not transfer

Three problems show up almost immediately when you try to run a change programme on a defensive risk template. I will take them in order of how much damage they do.

- There is no settled first line. The three-lines model rests on a stable business that owns its risks. A transformation programme is, by definition, the thing dismantling and rebuilding that business. Who is the "first line" for a capability that does not yet exist? The delivery teams building it? The operation that will one day run it but today has no visibility of it? The ambiguity is not a detail to be tidied up; it is structural. The model's foundational layer is the one a programme most conspicuously lacks, and everything built on top of a missing foundation inherits the gap.
- The defensive metaphor optimises for the wrong thing. Language shapes behaviour, and "lines of defence" tells everyone in the system that their job is to hold a line — to not be the one who let something through. That is exactly the right instinct for protecting a bank against fraud and exactly the wrong one for delivering change, where the dominant failure mode is not that something bad got through but that nothing got decided. A defensive posture rewards the person who raised the concern and escalated it; it is silent about the person who needed to make a cross-cutting call and did not. Programmes rarely die of un-caught risks. They die of un-made decisions, and a defence model has nothing to say about those.
- Independence hardens into distance. The virtue of the three-lines model is the independence of each layer from the one below. In a running operation that distance is healthy. In a programme moving at pace it becomes latency and detachment. The second and third lines, precisely because they must stay independent, sit at arm's length from the delivery — reviewing quarterly, assuring against a plan, careful not to get so close that they compromise their own objectivity. And so the people with the clearest view that the programme is heading for the rocks are structurally required to keep their distance from the wheel. The independence that protects a bank starves a programme of exactly the engaged, in-the-room judgement it most needs.

A bank's risk model asks "is anything getting through our defences?" A programme needs its governance to ask "is the thing we are building actually going to work, and who will decide when it won't?" These are not the same question, and a model built for the first will not answer the second.

4. The strongest case for importing it anyway

Let me put the opposing view at its strongest, because there is a real one and I have heard capable people make it.

The argument is that programmes have historically suffered from precisely the absence of the discipline the three-lines model brings. Too many transformation efforts have been marked their own homework — the same people who built the plan assured it, the same optimism that drove the delivery also reported on it, and the board received comfort manufactured by the very team whose future depended on the programme surviving. The three lines, on this view, is the cure for marking your own homework. It forces the separation of ownership from assurance that programmes have lacked; it gives the board a genuinely independent third-line voice; and in regulated sectors, where a programme's failure can become a regulatory event, that independent assurance is not optional. Import the model, the argument concludes, precisely because programme governance has been too cosy for too long.

I accept the diagnosis entirely and reject the prescription. Programmes have been too cosy; marking your own homework is a real and chronic failure; independent assurance genuinely matters, and in regulated delivery it is indispensable. But the separation of ownership from assurance does not require importing a defensive perimeter designed for a standing operation. It requires exactly one thing — that the people who assure the programme are not the people who own its delivery — and that principle can be honoured inside a governance model built for change without dragging in the three-lines model's defensive posture, its assumption of a settled first line, and its enforced distance from the wheel. The objection is right that programmes need independent challenge. It is wrong that the risk world's defensive architecture is the way to get it. You can keep the separation and discard the metaphor.

“Keep the independence. Discard the defensive perimeter. A programme does not need to be defended against itself; it needs to be helped to decide.”

5. Adapt, do not adopt

So what should a practitioner do with the three-lines model when it lands on the table, as it increasingly will? Not reject it — there is real wisdom in its separation of ownership from assurance — but refuse to adopt it whole, and adapt it deliberately to the fact that a programme is a machine for change, not a standing operation to be defended.

That adaptation has, in my experience, three moves. Keep the independence of assurance but bring it close — an assurance function that reviews continuously and sits near the delivery, trading a little theoretical objectivity for a great deal of timely, engaged judgement, rather than one that preserves its distance and its quarterly cadence and its uselessness in the moment that matters. Replace the missing first line with an explicit owner of the future operation — a named accountable executive for the capability being built, who stands in for the business-as-usual that does not yet exist, so that the model has a foundation rather than a hole. And add the layer the defence model entirely omits: a decision-forcing mechanism, a body whose job is not to watch the lines but to make the cross-cutting calls the lines will never make, at the speed the programme generates them.

Do that, and you have taken the genuine insight of the three-lines model — that owning and assuring are different jobs — and set it inside a governance model that understands it is governing change. Import the diagram unaltered, and you get what my composite

programme got: three lines of immaculate, independent, well-documented defence, holding firm on every side, while the programme dies quietly in the undefended middle where the decision should have been made. The model was never wrong. It was answering a question the programme was not asking.