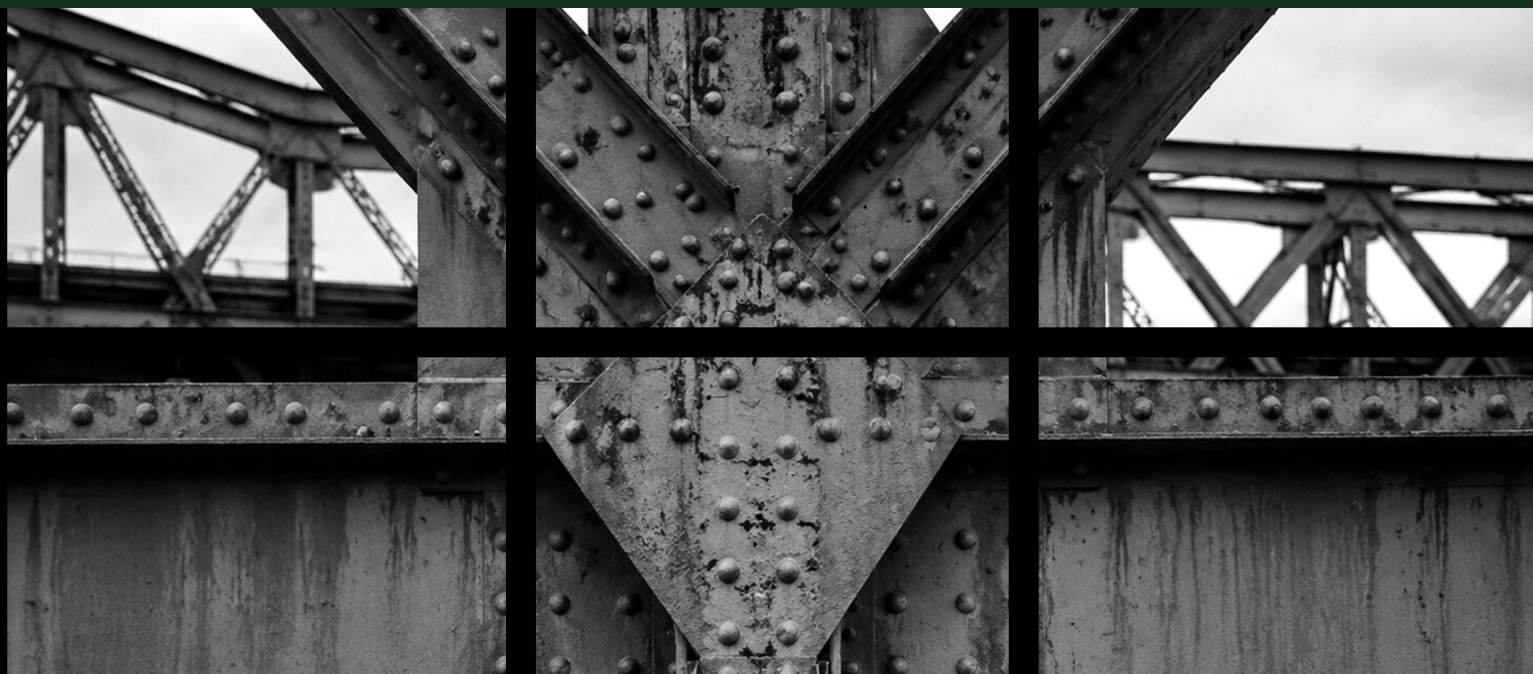


BEYOND CLOSURE

The Lessons-Learned Log Is Where Learning Goes to Die

A Practitioner's Perspective on evaluating programmes after the work is done



Giovanni Leonardi

March 2006

The Lessons-Learned Log Is Where Learning Goes to Die

Programme · No. 53

Originally written March 2006 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“A lesson is only learned when it changes a future decision.”

The document nobody reads

Every programme of any size closes with a lessons-learned exercise. A workshop is held, sometimes catered, and the team that has spent two years together spends two hours listing what went well and what did not. The output is a document. The document is filed, with genuine good intention, in a place from which it is meant to inform the next programme. I have written several of these documents. I have almost never seen one read by the people it was written for. The lessons-learned log is, in most organisations, the place where learning is sent to die with a clear conscience.

Evaluation that cannot teach

The reason is not laziness, and it is not that the lessons are wrong. It is that the entire apparatus of post-programme evaluation, as commonly practised, is built in a way that makes learning nearly impossible. Three faults do most of the damage.

The first is that we evaluate at the wrong moment. Closure happens when the last deliverable is signed off and the team is being demobilised — which is to say, precisely when the outcomes the programme existed to produce have not yet arrived. A new operating model has been implemented but not yet lived in; a system is live but its benefits depend on behaviours that take a year to settle. So the evaluation, forced to report at closure, reports on the only thing available to it: whether the programme was delivered. Delivery is not the outcome. We are grading the building of the machine, months before anyone can see whether the machine does what it was bought to do.

The second is that we evaluate the wrong subject. The lessons-learned workshop tends to catalogue execution — the plan was too optimistic, the requirements churned, the third party underperformed. These are real, but they are the symptoms most visible to the exhausted team at the end. The questions that would actually teach the organisation are harder and are rarely asked: was the programme worth doing at all; did the original business case describe a benefit that materialised; would we, knowing what we now know, have approved it in the form we approved it. Those questions indict the decision to invest, not merely the effort to deliver, and the decision-makers are not in the room.

The third is that we evaluate in a way that punishes the truth. When evaluation carries the faint odour of audit — when its findings can attach to named individuals and follow them — everyone in the room manages their exposure. The account that emerges is the safe one. The most valuable lesson in any programme is usually the one someone would be embarrassed to volunteer, and embarrassment is exactly what the closure evaluation is structured to produce.

An evaluation that can damage the people who tell the truth will be told a version of events optimised for safety, not for learning. You cannot audit and learn with the same instrument at the same time.

Learning from what actually happened

If the closure ritual is the wrong instrument, the practitioner has to ask what genuine learning from a programme would require. It is not more elaborate templates. It is a different design, and the changes are not complicated to describe — only uncomfortable to adopt.

1. Separate delivery closure from benefit evaluation in time. Close the programme when the work is done, by all means; that is an administrative necessity. But schedule the evaluation that matters for the moment when the benefits can actually be observed — six months, a year, whatever the benefit case implies — and fund it, name an owner for it, and put it in the diary before the team disperses. An evaluation nobody is left accountable to perform will not be performed.
2. Evaluate against the promise, not the plan. Go back to the original business case and its stated benefits, and ask, plainly, which ones happened, which did not, and why. This is the comparison the organisation most needs and most avoids, because it is the one that reflects on the people who approved the investment. It is also the only comparison that improves the next investment decision rather than merely the next delivery.
3. Aim the findings at the system, not the individuals. The purpose is to improve how the organisation selects, shapes, and governs its programmes — not to assemble a case against anyone. That means deliberately severing the evaluation from performance management, saying so, and meaning it. The learning worth having lives in the candour that only safety unlocks.
4. Feed it back where decisions are made. A lesson is only learned when it changes a future decision. That means the findings have to reach the body that approves the next business case, in a form it will actually use — a revised assumption, a sharper question added to the approval gate — not a filed document nobody opens. Learning is institutional or it is nothing.

The honest reckoning most programmes skip

Behind all of this sits a discomfort that the closure ritual is designed, however unconsciously, to avoid. To ask honestly whether a programme delivered the value it promised is to risk discovering that it did not — that a great deal of effort and money produced an output that works but a benefit that never came. Organisations do not like to look directly at that possibility, and the beautifully filed lessons-learned log is one of the more sophisticated ways they have of not looking. It performs the appearance of reflection while quietly ensuring that the one question capable of changing behaviour is never put.

I do not say this cynically. The people involved are usually tired, usually proud of what they built, and usually being pulled onto the next thing before the last one has cooled. The system gives them no time, no incentive, and no safety to conduct the reckoning that

would teach the organisation something. That is precisely why it has to be designed in deliberately: named, funded, timed to when the truth is visible, and protected from the instinct to turn it into judgement.

The test of whether a programme has been learned from is not whether a document exists. It is whether the next business case through the gate is interrogated with a question that the last programme taught the organisation to ask. Until the evaluation reaches that far, we have not been learning from what actually happened. We have been filing it.