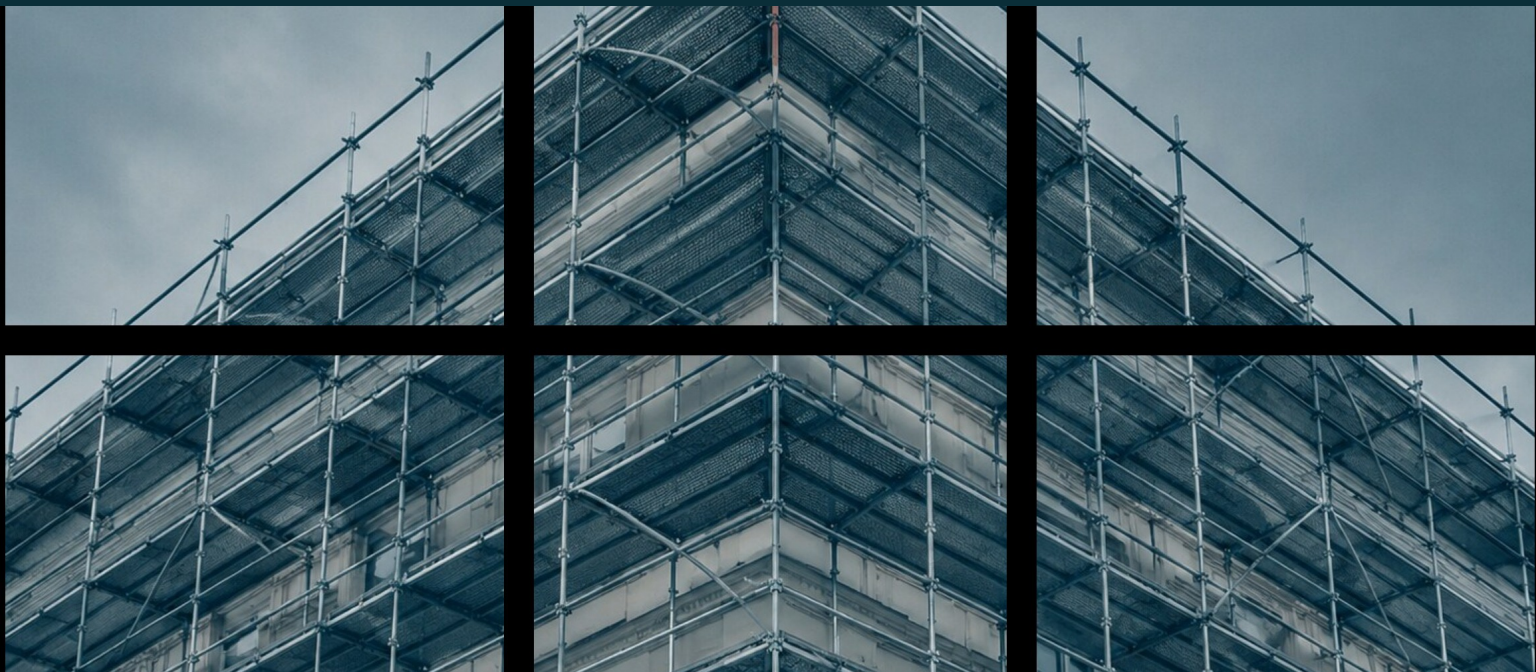


BEYOND THE ADOPTION WORKSTREAM

Bolted On, Not Built In: Why Change Is the Afterthought of Transformation

On the structural reasons the people dimension keeps arriving after the design is frozen — and what building it in would truly require



Giovanni Leonardi

September 2006

Contents

1. Executive Summary	3
2. The Workstream That Arrives Late	3
3. The Anatomy of an Afterthought	4
4. Why the Pattern Persists	4
5. The Iteration That Changed Nothing	5
6. The Case for Bolting It On	6
7. What "Built In" Actually Means	6
8. The Afterthought We Choose	7

Bolted On, Not Built In: Why Change Is the Afterthought of Transformation

Transformation · No. 82

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

“We reward the launch and hope for the landing, and the two are separated by exactly the period in which change management would have had to do its work.”

1. Executive Summary

There is a pattern in the practice of large-scale change so familiar that we have almost stopped noticing it. The work of helping people actually take up a new way of working — what we have come to call change management — arrives late. It appears on the plan after the solution has been designed, it draws on the softest line in the budget, and it is judged by how many people attended the briefing rather than by how many changed what they did on Monday morning. It is bolted on.

The comfortable explanation is that organisations simply need to do change management earlier and louder. That explanation is shallow. We are not short of method. The models of Lewin, of Kotter, of the transition curve have been in circulation for years, and most large programmes can recite them on demand. The bolt-on persists not because we lack technique but because of how we fund transformation, how we govern it, how we measure it, and — beneath all of these — how we imagine it. Each of these forces quietly pushes the human dimension downstream of the design, and each is entirely rational from where the individual decision-maker sits.

This essay traces those forces. It takes seriously the argument that bolting change on may sometimes be the sensible thing to do. And it sets out what the alternative actually requires — because “built in” does not mean starting the training sooner. It means allowing the way people will really work to constrain what gets built in the first place. That is a more uncomfortable proposition, and a more powerful one.

2. The Workstream That Arrives Late

Picture a transformation nine months into an eighteen-month plan. The technical design is frozen. The build is under way. The steering report is a wall of green. And somewhere around this point a new line appears on the plan: Change, Communications and Training. A change lead is appointed — often capable, often experienced — and asked to prepare the organisation to receive what has already, in every meaningful sense, been decided.

The change lead's task is now one of delivery and persuasion. The processes are set. The system's screens are built. The reporting lines implied by the new operating model are fixed. What remains is to explain all of this to the people who must live inside it, to train them, and to manage the entirely predictable resistance that follows. And when the resistance comes, it is read as a people problem — a failure of engagement, a deficit of communication — rather than as feedback on a design that was completed without the

people in the room.

This is the bolt-on in its natural habitat. It is not that change was forgotten. It was scheduled. It had a budget line and an owner. It simply arrived after the moment at which it could have shaped anything.

3. The Anatomy of an Afterthought

Look closely and the bolt-on has a consistent structure, visible in four places long before anyone talks about resistance.

The first is sequence. The plan is built around the artefact — the system, the process map, the new structure — and the human work is placed after it, in the gap between "build complete" and "go-live". Sequence is destiny here: whatever comes last inherits whatever time and money the earlier work did not consume, which is reliably very little.

The second is budget. In most business cases the people line is the soft one. The licences, the integration, the infrastructure are hard numbers with invoices attached; the change effort is a percentage, an allowance, a rounding item. When the programme runs hot — and it always runs hot — the soft line is where the money is found. Nobody decides that adoption does not matter. They simply decide, under pressure, that this quarter's overspend matters more.

The third is ownership. Change is handed to a function — a change team, a communications group, an external adviser — rather than carried by the executive accountable for the benefit. The moment it has its own owner, it becomes a separable thing, a workstream among workstreams, something that can be reported on in isolation and, when convenient, deferred in isolation.

The fourth, and the most corrosive, is measurement. We measure what the programme builds, not what the organisation absorbs. Milestones are events the programme controls: design signed off, build complete, system live. Adoption is a state the programme does not control, and so it is rarely on the report. The status stays green because green measures activity — briefings delivered, users trained, help-desk staffed — and none of those things is the same as a person doing the work differently.

A programme reports on what it can control. It builds a system and controls the build, so the build is on the report. It cannot control whether people change, so adoption is not — and the green status is technically true and substantively empty.

4. Why the Pattern Persists

If the bolt-on were merely a mistake, twenty years of writing and training would have cured it. It persists because several forces, none of them foolish, all point the same way.

- The engineering worldview. We model transformation as a delivery problem — something to be specified, built, tested and released. That worldview is enormously productive for the technical artefact and quietly disastrous for everything else, because it casts the human response as downstream of "the real work" rather than as part of the design itself. The plan is a bill of materials for a system, not for a changed

organisation.

- The economics of the business case. Benefits are assumed to flow from the artefact. Install the system and the savings appear; stand up the shared service and the efficiencies follow. But benefits do not live in systems. They live in people doing something different, at scale, and continuing to do it once the programme has gone home. A business case that treats adoption as automatic has already written change out of the story before the first pound is spent.
- The professionalisation paradox. Change management has, over the past decade, become a discipline — with its own models, its own accreditations, its own language of stakeholders and readiness and reinforcement. This is largely a good thing, and it carries an unintended cost. The more change management looks like a distinct capability you can procure, the easier it becomes to treat it as a thing you bring in rather than a property of how the whole endeavour is conceived. We professionalised the discipline and, in the same motion, made it detachable.
- The horizon of reward. Sponsors are thanked at go-live. The applause, the steering-committee congratulations, the career credit all cluster around the moment the system switches on. But adoption — the place where benefit actually lands — matures in the twelve to twenty-four months after that moment, long after the programme has been stood down and the sponsor has moved to the next thing. We reward the launch and hope for the landing, and the two are separated by exactly the period in which change management would have had to do its work.

Put these together and the bolt-on is not an aberration. It is the equilibrium. Every local incentive — the plan, the budget, the org chart, the reward — pushes the human dimension to the end, and then we are surprised to find it there.

5. The Iteration That Changed Nothing

There is a tempting belief that the bolt-on is an artefact of one particular way of working — the long, specify-everything-up-front programme — and that newer, more iterative approaches have quietly solved it. Deliver in increments, engage continuously, put working software in front of users every few weeks, and surely the human dimension is built in by construction. The team is talking to the organisation the whole way through; there is no late workstream because there is no "late".

The belief is half true, and the false half is the one that matters. Iterative delivery does dissolve one version of the problem: users see the artefact early and often, feedback arrives while the design is still soft, and the crude spectacle of a launch to a surprised organisation becomes rarer. That is a real gain. But watch where the continuous engagement actually points. It points at the artefact — the screens, the features, the backlog. The team iterates fluently on what is being built and stays as silent as ever on how the organisation will work once it is built: the operating model, the accountabilities, the incentives, the knock-on effects on roles two departments away from the delivery team. The conversation is continuous and narrow. We have tightened the feedback loop around the thing and left the wider human system precisely where it was — outside the room, to be managed later.

So the bolt-on survives the change of method, because it never lived in the method. It lives in the boundary we draw around "the work". Draw that boundary around the artefact — whether you build it in one long phase or fifty short ones — and everything outside the boundary becomes someone else's problem, arriving after the fact. The paradigm changed; the boundary did not; and the afterthought reappeared in new clothes.

6. The Case for Bolting It On

It would be too easy to stop there, and dishonest. There is a serious argument on the other side, and it deserves its strongest form rather than a caricature.

The argument runs like this. Adoption is inherently downstream. You cannot train people on a process that does not yet exist, or build readiness for screens no one has seen. Engaging the organisation too early, before there is anything concrete to engage with, burns goodwill on abstractions and produces consultation fatigue long before there is a system to adopt. There is also a real risk of over-managing: not every change needs a change programme, and organisations absorb a great deal of small change perfectly well without ceremony. On this view, doing the heavy human work late is not negligence but timing — you spend the effort when it can bite, not before.

This is not wrong. Its error is narrower and more interesting. It assumes that "change management" means the activities — the training, the communications, the readiness assessments — and it is quite right that those activities belong later, when there is something real to train on. But the activities are not the point. The point is whether the way people will work is treated as an input to the design or as a consequence of it. That decision is made at the very beginning, when the artefact is being shaped, and it cannot be deferred. To bolt on the activities is reasonable. To bolt on the consideration — to design the system, the process and the structure first and consult the working reality afterwards — is the failure. The steelman defends the timing of the training and, without noticing, smuggles in a defence of designing without the people. Those are different things.

7. What "Built In" Actually Means

If built-in does not mean doing the same activities sooner, what does it mean? It means letting the human reality act as a design constraint on the artefact itself — with the same authority as a technical or financial constraint.

Consider a composite that will be familiar to anyone who has lived through an enterprise systems programme. A finance and operations platform is to replace a patchwork of regional spreadsheets. The build is costed at some fourteen million; a change and training allowance of perhaps six hundred thousand appears on the plan at month nine. Go-live looks green: users are trained, logins climb, the report is clean. Three months later, half the regions are quietly reconciling in shadow spreadsheets alongside the new system, because the month-end process the platform enforces does not match the sequence in which those regions actually close their books. The design was correct in the abstract and wrong in the one place that mattered — the desk of the regional operations manager whose real month-end nobody had mapped. The benefit case assumed adoption; adoption assumed a process that fit; and the process had been designed without the person who would have to run it.

Now imagine the same programme built in. The operating model — who does what, in what sequence, with what handoffs — is designed alongside the technical model, not after it, and the regional close is a constraint on the platform rather than a casualty of it. The accountable executive for the benefit is a line leader, not a programme, so the question "will people actually work this way?" has an owner from day one. Governance reports leading indicators of adoption — how many regions have run a real close on the new process, not how many attended training — so the green status means something. None of

this is a bigger change workstream. It is a different location for the same concern: upstream, in the design, where it can still change the artefact.

Bolted on	Built in
Change scheduled after the design is frozen	The way people work constrains the design
Benefits assumed to flow from the artefact	Benefits owned by the line leader who must realise them
Status measures activity — briefings, logins, training	Status measures adoption — real work done the new way
Change owned by a separable function	Change carried by whoever is accountable for the outcome
Resistance read as a people problem	Resistance read as feedback on the design

The distinction is not effort. A bolted-on programme often works harder at change than a built-in one, precisely because it is trying to compensate after the fact for a design that fought the organisation. Built-in is not more change management. It is change management that has been made largely unnecessary, because the thing being delivered was shaped, from the start, to fit the people who would have to live in it.

8. The Afterthought We Choose

The uncomfortable conclusion is that the bolt-on is not a gap in our methods. Our methods are fine, and there are more of them every year. It is a gap in how we conceive of transformation — as the delivery of an artefact, with the human response as an afterthought to be managed — and no amount of better afterthought-management will close a gap that opens at conception.

“We do not lack the discipline of change. We lack the conviction to let it shape the thing we are building, rather than clean up after it.”

That is why the pattern is so durable, and why it will outlast this essay. Building change in costs something real at the start: it slows the early design, it forces awkward conversations with the people who will have to work the new way, it puts a line leader on the hook for a benefit that will not mature for two years. Bolting it on defers all of that — the cost, the conversations, the accountability — to a moment when the programme is nearly over and the sponsor is nearly gone. Faced with that trade, organisations choose deferral again and again, not out of ignorance but out of the ordinary preference for a cost postponed over a cost paid now.

The first step is not another model. It is to stop treating the human dimension as a workstream and start treating it as a design constraint — and to notice, each time we draw up the plan and place Change, Communications and Training somewhere after the build, that we have just made a choice, and that the choice has a price we will pay later, with interest.