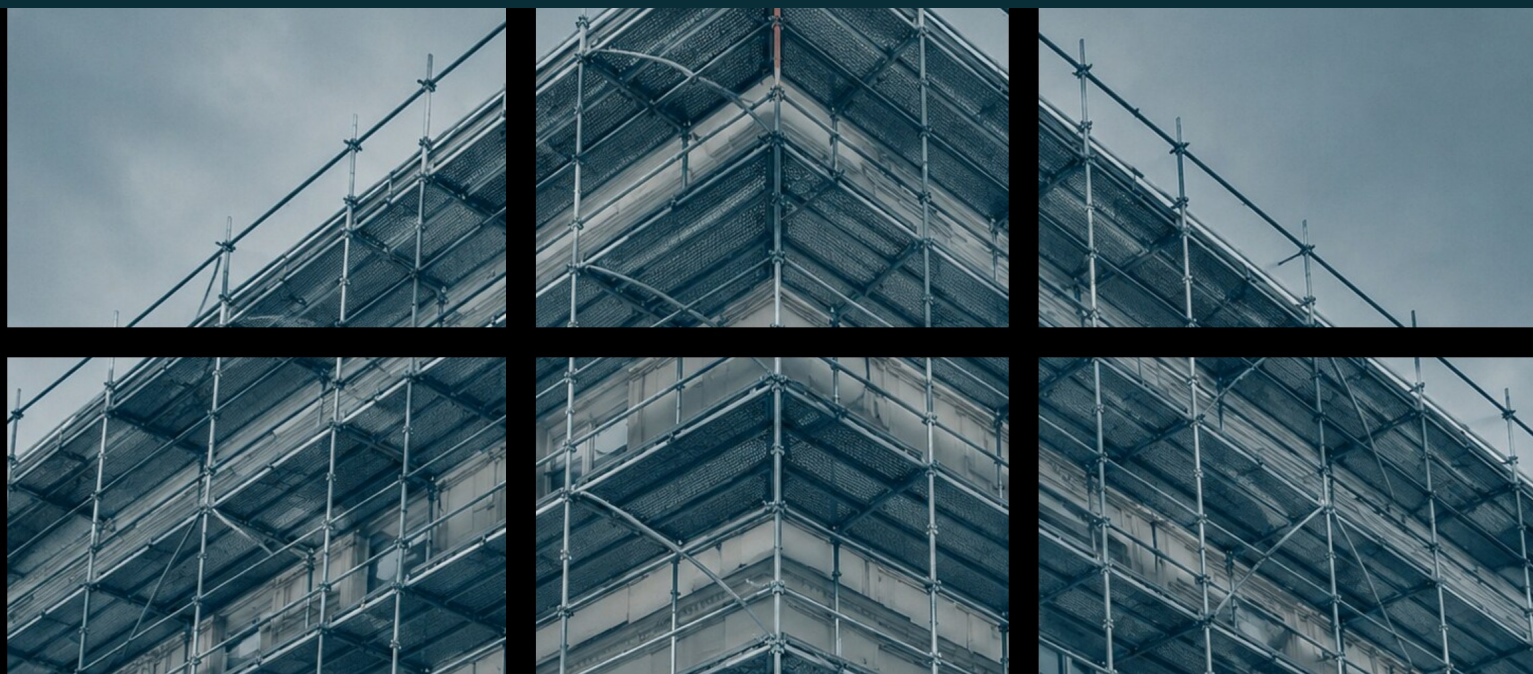


BEYOND THE GO-LIVE

Bolted On, Not Built In: What the Afterthought of Change Management Reveals About How Organisations Actually Change

On the adoption gap, the severable budget line, and the quiet reversion that follows the flawless go-live.



Giovanni Leonardi

September 2006

Contents

1. Executive Summary	3
2. The Flawless Go-Live and the Quiet Reversion	3
3. The Shape of the Afterthought	4
4. Why the Bolt-On Persists	4
4.1 The business case is an engineering document	5
4.2 The budget line that is always severable	5
4.3 Arriving after the decisions are made	5
4.4 The vocabulary of "soft" and "hard"	5
4.5 Benefits credited to the machine, not the behaviour	5
5. The Case for Leaving It Late	6
6. What Actually Changes When an Organisation Changes	7
7. A Consolidation That Went Live and Did Not Land	7
8. Toward the Built-In — and Its Own Temptation	8
9. The Mirror	8

Bolted On, Not Built In: What the Afterthought of Change Management Reveals About How Organisations Actually Change

Transformation · No. 105

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

“Asking someone to secure adoption of a design they could not influence is asking them to sell a house whose foundations they were not allowed to inspect.”

1. Executive Summary

Every experienced hand has stood in the same room. The programme has delivered. The system is live, the data has migrated cleanly, the defect log is green, the steering committee has been thanked. And within a quarter, quietly and without anyone deciding it, a meaningful share of the organisation has drifted back to the spreadsheets, the side-agreements, and the old way of getting the month closed. Nothing failed. And nothing changed.

This essay is about the most durable pattern in transformation work: that we build the machine with rigour and treat the human re-arrangement it requires as something to be bolted on afterwards — a workstream running alongside the real build, funded last, cut first, and staffed by people who arrive after the decisions that would have made their job possible have already been made. The pattern is not a failure of knowledge. Every methodology on the shelf insists that adoption matters. The pattern persists because the deep structure of how we fund, govern, sequence, and account for transformation makes bolting-on the path of least resistance and building-in the path of most friction.

The argument runs in three movements. First, that the bolt-on is manufactured by five specific structural forces, not by ignorance or neglect. Second, that there is a serious and honest case for leaving change late — and that it is wrong for a reason worth understanding. Third, that the whole pattern is a mirror: it reveals that we still, underneath our language of stakeholders and readiness, believe an organisation is a machine to be re-engineered rather than a community to be re-persuaded. Organisations do not change when the system goes live. They change when enough people renegotiate their private idea of what their job is — and that renegotiation is the thing our programmes are structured to treat as somebody else's problem.

2. The Flawless Go-Live and the Quiet Reversion

Picture the go-live that goes perfectly. A shared-services consolidation, eighteen months in the making, pulling three regional finance operations onto one platform. The cutover weekend is a triumph of programme management — reconciliations tie out, the interfaces hold, the help desk is quiet by Tuesday. The programme director sends the note everyone has been waiting for. The system, by every measure the programme was asked to hit, works.

Then the months pass. At month three, someone finally looks at how the new exception-handling process is actually being used, and finds that something close to half of the regional controllers are still resolving exceptions the way they always did — offline, in a private workbook, reconciled back into the system after the fact. The platform records a clean process. Underneath it, the old organisation is running exactly as before, now with an extra step to keep the system looking fed.

Nobody rebelled. There was no dramatic rejection to point to in a lessons-learned deck. There was only the quiet reversion — the slow, sensible, self-protective return to the way of working that people trusted, understood, and could be held accountable for. This is the characteristic failure of transformation, and it is almost never a failure of the build. It is the gap between the system going live and the organisation going along with it. We have a great deal of language for the first event and remarkably little discipline aimed at the second.

3. The Shape of the Afterthought

Look at how a typical programme is assembled and the afterthought is visible in its very anatomy. The plan has a spine — requirements, design, build, test, deploy — and running alongside that spine, in a thinner colour, is a workstream variously called change, adoption, readiness, or simply "the people side." It has fewer people. It starts later. Its deliverables are communications plans, training schedules, and stakeholder maps rather than anything that alters the design of the thing being delivered. And when the programme runs hot — as programmes do — it is the first place the plan reaches for time and money.

The tell is in the prepositions. The change effort is run alongside the build, to support adoption of a solution designed elsewhere. It is never inside the design, shaping what gets built and how the work will actually be done. We say change management and mean, in practice, change communication: the task of explaining, selling, and cushioning a set of decisions that were taken without the people the change will land on ever being genuinely in the room.

Bolted-on change management is not a discipline that failed to be integrated. It is a discipline that was designed, funded, and scheduled to be separable — and separability is precisely the property that guarantees it will be severed.

This is worth stating plainly because the usual explanation — that leaders "don't get the soft stuff" — is both flattering to us and false. Most leaders have sat through enough go-lives to know exactly how the quiet reversion works. They commission the change workstream sincerely. The bolt-on survives not because people fail to value it but because the machinery of transformation keeps producing it regardless of what anyone believes.

4. Why the Bolt-On Persists

If the pattern were caused by ignorance, a decade of Kotter, of Bridges' work on transitions, of the readiness models that every consultancy now packages, would have dissolved it. It has not. The pattern persists because at least five structural forces each independently push change to the margin, and they reinforce one another.

4.1 The business case is an engineering document

A transformation is funded on the strength of its business case, and the business case is almost always written in the language of the deliverable. It describes a system, a consolidated process, a new operating structure, a headcount reduction. Its benefits are engineered quantities — licences retired, processes standardised, error rates reduced, full-time equivalents removed. In this document the human transition is not a line of reasoning; it is an assumption. Adoption is taken as the automatic consequence of deployment, the way a switch being flipped is assumed to turn on the light. When the benefit is modelled as flowing from the existence of the system rather than from a change in what people do, the effort required to change what people do has no natural home in the case that justifies the spend.

4.2 The budget line that is always severable

Because change appears in the plan as a distinct workstream rather than as a property of every other workstream, it appears in the budget as a distinct and therefore cuttable line. And it has three fatal characteristics as a budget line: its benefits are diffuse and delayed, its deliverables are hard to inspect for quality until it is too late, and cutting it produces no immediate, visible failure. Compare it to the build. If you defund testing, something breaks at cutover and everyone can see it. If you defund adoption, nothing breaks at cutover — the reversion is quiet and arrives a quarter later, long after the contingency raid that caused it has been forgotten. A rational programme under budget pressure will therefore reach for the change line first, every time, and will almost never be punished for it within the window in which anyone is still keeping score.

4.3 Arriving after the decisions are made

Consider when the change lead actually joins. Typically after the scope is fixed, the design is largely set, the timeline is committed, and the target operating model has been drawn. They are handed a set of decisions and asked to "drive adoption" of them. But the decisions that determine whether adoption is even possible — whether the new process fits how the work really flows, whether the people who hold the exceptions were consulted, whether the design accounted for the informal arrangements that keep the current organisation running — those decisions have already been taken, by people optimising for cost, timeline, and technical elegance, in rooms the change lead was not in. Asking someone to secure adoption of a design they could not influence is asking them to sell a house whose foundations they were not allowed to inspect.

4.4 The vocabulary of "soft" and "hard"

Our own language sabotages us. We speak of the "hard" side — systems, processes, structures — and the "soft" side — behaviour, culture, mindset. The metaphor is not neutral. Hard connotes real, load-bearing, engineerable, measurable; soft connotes pliable, secondary, nice-to-have, the thing you attend to once the real work is done. No programme director, watching the burn rate climb, cuts the "hard" work to protect the "soft" work; the words themselves have already ranked them. Until the human dimension of change is understood as structural — as load-bearing as the interface design — the vocabulary will keep sorting it into the discretionary pile.

4.5 Benefits credited to the machine, not the behaviour

There is a final, closing force, and it is the one that makes the whole system self-perpetuating. When benefits do materialise, they are attributed to the system. The month-end closes faster: the platform gets the credit. When benefits fail to materialise, the diagnosis reaches for scope, data quality, or "user resistance" — never the absence of the built-in work that would have changed the behaviour on which the benefit actually depended. Because the accounting never traces a realised benefit back to the behaviour change that produced it, the case for building change in is never made, run after run. The organisation learns, wrongly but consistently, that value comes from the machine. And so the next business case is written, once again, as an engineering document, and the cycle begins where it began.

"We defund adoption and nothing breaks at cutover; the reversion is quiet, and it arrives a quarter later — long after the decision that caused it has been forgotten."

5. The Case for Leaving It Late

It would be too easy to stop there, with the structures indicted and our own good intentions exonerated. The bolt-on has a serious defence, and it deserves to be met at its strongest rather than waved away.

The defence goes like this. Change is a genuinely distinct competency — the skills of stakeholder engagement, communication, and transition support are not the skills of process design or systems architecture, and asking one team to hold all of them produces mediocrity in each. Separating concerns is not sloppiness; it is sound organisation of work, the same principle that puts testing in the hands of testers. And there is a sequencing argument that is even harder to dismiss: you cannot meaningfully design adoption for a solution that does not yet exist. Until the design is stable, any change effort is either speculative or actively obstructive — a team writing communications about a process that is still being argued over, training people on screens that will be redrawn next month. On this view, bringing change in late is not neglect. It is the discipline of not doing work before the work can be done well.

This is a strong argument, and the honest response is not to deny it but to see precisely where it slips. It conflates two different things: the modularity of the skill and the sequencing of the involvement. It is entirely true that change is a distinct competency held by distinct people — and entirely false that those people should therefore be introduced only once the design is locked. A structural engineer is a distinct competency from an architect, and no one concludes that the structural engineer should first see the building after the floor plan is signed. They are in the room during design precisely because their competency changes what gets drawn. The separation-of-concerns argument justifies a distinct change capability; it does not justify a late change involvement. And the sequencing argument proves less than it claims. It is true that you cannot design a training course for a screen that does not exist. But the decisions that determine adoption are made before the screen exists — in the choices about how the work will flow, who owns the exceptions, which informal arrangements the design will honour or ignore. Those are exactly the decisions the change competency should be shaping. Leaving it late does not spare us premature work; it removes the one voice that should have been loudest while

the load-bearing decisions were still soft.

6. What Actually Changes When an Organisation Changes

Strip the pattern down and it rests on a buried assumption about what a transformation is. The bolt-on model assumes the change is the system, and the people are the medium through which the system must pass — friction to be reduced, resistance to be managed, a population to be moved from a current state to a future state like water pushed through a pipe.

But watch what actually happens when an organisation genuinely changes, and it is nothing like water through a pipe. It is a great many people, more or less independently, arriving at a new answer to a private question: what is my job now, and how do I stay competent and safe while doing it differently? The controller in our consolidation was not resisting the platform. She was protecting her ability to be accountable for a number she would still be asked about at month-end, using a method she trusted, because nobody had made her equally safe in the new one. Her reversion was not obstruction; it was competent self-preservation in the face of a change that had altered her tools without renegotiating her accountability.

An organisation does not change when the system goes live. It changes when enough of its people have renegotiated, one by one, their private idea of what their job is — and no cutover weekend can perform that renegotiation on their behalf.

This is the reveal, and it reframes everything. If change is the renegotiation of thousands of individual working realities, then "building it in" is not a matter of better communications or earlier training. It means treating the human re-arrangement as part of the design surface itself — designing the new accountabilities, the new definitions of competent work, the honouring of the informal arrangements that keep the place running, at the same table and in the same decisions as the process and the platform. The reason to build change in is not that people deserve to be looked after, true though that is. It is that the thing we are actually trying to alter — how people understand and perform their work — is only reachable through decisions we have systematically located outside the change workstream's reach.

7. A Consolidation That Went Live and Did Not Land

Return to the shared-services consolidation and trace the mechanism concretely, because the abstraction earns nothing until it survives contact with a real sequence of events.

The design optimised, sensibly, for standardisation: one exception-handling process across all three regions, replacing three local variants. The business case counted the benefit as eleven full-time equivalents removed from duplicated reconciliation work — a clean, defensible, engineerable number. The design was elegant. What it did not account for was that in one region, the "local variant" was not sloppiness but the accumulated response to a genuinely different set of counterparties, and the controller there resolved a category of exception through a relationship and a judgement that the standard process had no place for. That single design decision — to standardise the exception process without resolving who would own the exceptions the standard could not handle — was

made in month four, in a design authority meeting, on cost and simplicity grounds, with no one in the room whose job was to ask what it would do to the work.

The consequence surfaced at month three after go-live as an adoption figure: roughly forty per cent of exceptions in that region still handled offline, reconciled back after the fact. The eleven-FTE benefit did not materialise, because the duplicated work had not been eliminated — it had been driven underground and now ran in addition to the standard process, which had to be kept fed. The programme's own accounting recorded this as "user resistance in Region C" and a data-quality issue. The truer description is that a load-bearing decision about accountability had been taken as though it were a decision about process efficiency, at a table the change competency was not invited to, four months before anyone would be asked to "drive adoption" of its consequences. The cost of building change in would have been one different conversation in month four. The cost of bolting it on was a benefit that never arrived and a quarter of shadow work no one had budgeted for.

8. Toward the Built-In — and Its Own Temptation

If the diagnosis is that our funding, sequencing, vocabulary, and accounting each push change to the margin, then building it in is not a technique to be adopted but a set of structural moves that run against each of those forces. It means writing the business case so that benefits are traced explicitly to changes in behaviour, not to the existence of the system, so that the case cannot be closed without the behaviour being someone's responsibility. It means funding the human re-arrangement as a non-severable property of every workstream rather than a separable line, so that cutting it requires cutting the thing it is part of. It means seating the change competency at the design table while the load-bearing decisions are still soft, not after they have set. And it means retiring the "hard and soft" vocabulary for language that treats the work of the organisation as exactly as real and as engineerable as the work of the system.

But intellectual honesty requires naming the temptation that lies on the other side of this argument, because it is real. Once "change is everything and everything is change" becomes the slogan, a different failure appears: the transformation in which every decision is a change decision, every workstream carries a change tax, and the discipline expands until it dilutes into a general fog of engagement, workshops, and readiness assessments that deliver activity without delivering the specific renegotiations that matter. The bolt-on at least has the virtue of being accountable — you can see what the change workstream produced. A change effort dissolved into everything can become accountable for nothing, a way of being present at every meeting and responsible for no outcome. Building change in is not the same as smearing it everywhere. It is the harder discipline of identifying the specific load-bearing decisions on which some future adoption will turn — who owns the exceptions, whose accountability the new tool alters, which informal arrangement the design must honour — and being ruthlessly present for those, while staying out of the decisions where the change competency genuinely has nothing to add. Built-in is a scalpel, not a flood.

9. The Mirror

Why, in the end, does the bolt-on persist against all our stated knowledge? Because it faithfully reflects what we still, beneath our language, believe. We describe organisations as living systems and cultures and communities in our conference talks, and then we fund,

plan, and account for their transformation exactly as we would the re-engineering of a machine — as though value lived in the components and people were the medium of transmission. The afterthought is not hypocrisy. It is a sincere expression of a model of the organisation that our vocabulary has outgrown but our structures have not.

The quiet reversion, then, is the organisation answering back. It is thousands of competent people declining to become the medium through which someone else's design is transmitted, and insisting — correctly — that their work is theirs to renegotiate and not ours to switch over on a cutover weekend. We can keep reading that answer as resistance, and keep bolting on a discipline whose job is to overcome it. Or we can read it as information: that the thing we are trying to change was never the system in the first place, and that until the human re-arrangement sits inside the design rather than alongside it, we will keep delivering flawless machines to organisations that quietly, sensibly, and entirely predictably decline to change.