

BEYOND PROGRAMME CLOSURE

The Transformation Backlog as Organisational Debt

The Hidden Liability That Compounds Faster Than Technical Debt

Giovanni Leonardi

June 2019



The Transformation Backlog as Organisational Debt

Portfolio · No. 80

Originally written June 2019 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“Every programme that is announced but not completed, every restructuring that is implemented on paper but not in practice, every new process that is trained but not adopted, withdraws from a finite account of organisational attention and trust.”

1. The Accumulating Ledger

Every quarter, the portfolio dashboard presents its familiar portrait of enterprise change: programmes plotted on heat maps, benefits tracked against baselines, resource utilisation charted in confident percentages. What the dashboard never shows — what no standard portfolio report has a column for — is the growing distance between the change the organisation has announced and the change it has actually absorbed.

We have a precise name for this phenomenon in software engineering. Technical debt — Ward Cunningham's metaphor from the early nineties — describes the accumulated cost of expedient decisions: shortcuts taken today that must be serviced tomorrow, with interest. The concept has become so embedded in engineering culture that teams budget for it, measure it, and make deliberate decisions about how much to carry. Yet the enterprise change agenda — the portfolio of transformations, restructurings, regulatory programmes, and digital initiatives that collectively reshape how an organisation operates — has no equivalent accounting. The backlog of unfinished, half-absorbed, or quietly deferred change sits on no balance sheet, accrues interest that nobody calculates, and compounds in ways that make each subsequent initiative more expensive than it needed to be.

This is transformation debt, and by mid-2019, most large organisations are carrying more of it than they realise.

2. The Anatomy of the Liability

Technical debt is legible because code is legible. A developer can point to the function that was written as a workaround, estimate the cost of refactoring it, and calculate the drag it imposes on every sprint that works around it. Transformation debt is harder to see because it lives in the space between what an organisation has decided and what its

people have actually internalised.

The pattern I have watched recur most consistently plays out like this. A new operating model is announced. Reporting lines shift on paper. Role descriptions are rewritten. The programme board declares the restructuring complete. Six months later, decisions are still being routed through the old informal networks because the new model was implemented structurally but never operationally. The programme has moved on; the organisation has not.

That gap — between the structural change and the operational reality — is a unit of transformation debt. And like its technical counterpart, it accrues interest. Every subsequent initiative that depends on the new operating model being functional — the process redesign that assumes the new accountability lines, the technology deployment that assumes the new ways of working — inherits a hidden dependency on change that was declared but not completed. The interest is paid in workarounds, in rework, in the inexplicable slowness of programmes that should have been straightforward.

3. Why the Interest Compounds

Technical debt compounds because each shortcut constrains the options available for the next piece of work. Transformation debt compounds through a different but equally relentless mechanism: the erosion of organisational capacity for change.

Every programme that is announced but not completed, every restructuring that is implemented on paper but not in practice, every new process that is trained but not adopted, withdraws from a finite account of organisational attention and trust. The withdrawal happens at announcement — the energy, the disruption, the demand for engagement. But the deposit — the benefit, the resolution, the new steady state — never arrives in full. The organisation has paid the cost of change without receiving the return.

We can observe the compound effect in the way transformation portfolios behave over time. In the first year of a sustained change agenda, programmes land with reasonable efficiency. By the third year of overlapping initiatives, each new programme takes longer, costs more, and meets greater resistance — not because the initiatives themselves are more complex, but because they are being deployed into an organisation already carrying the weight of unabsorbed change from everything that came before. The change capacity is overdrawn.

The pattern is visible in the language people use. When teams say "we haven't finished the last reorganisation and now there's a new one," they are describing interest payments. When middle managers quietly maintain shadow processes alongside the officially mandated ones, they are servicing transformation debt. When a digital programme finds that the "current state" it was designed against no longer exists — because three other programmes have partially altered it — the rework cost is interest on debt that was never acknowledged.

4. The Accounting Gap

Software engineering developed its debt vocabulary because it needed one. The metaphor gave teams a way to make the invisible visible – to argue that a sprint spent on refactoring was not wasted effort but a deliberate reduction in future cost. It gave leadership a framework for understanding why velocity was declining and what could be done about it.

The transformation portfolio has no equivalent framework, and the absence is not accidental. Three features of how we govern enterprise change conspire to keep transformation debt off the books.

The first is the binary completion model. Programmes are either in flight or closed. There is no recognised state for "structurally complete but operationally unabsorbed" – yet this is precisely where transformation debt accumulates. A programme that has delivered its outputs and been closed has, by the conventions of portfolio reporting, succeeded. That the organisation is still metabolising the change, still routing around the parts that did not take, still paying the cost of the gap between design and adoption – this appears nowhere in the governance record.

The second is the absence of a carrying cost. Technical debt is measured in developer time – the additional hours spent working around the shortcut, sprint after sprint. Transformation debt has no equivalent unit. The cost is real – slower adoption, higher resistance, duplicated effort, degraded trust – but it is dispersed across so many budget lines and so many programmes that it is effectively invisible. No portfolio director can point to a figure and say: this is what our unfinished change is costing us per quarter.

The third is the sunk-cost problem applied at scale. Writing off transformation debt means acknowledging that a programme's benefits were not fully realised – that the restructuring did not land, that the new process was not adopted, that the investment in change delivered less than was promised. In an environment where careers are built on programme delivery and where the incentive structure rewards completion over absorption, the organisational pressure to carry the debt silently is overwhelming. Writing it off means admitting it exists, and admitting it exists means confronting the question of what went wrong.

5. Towards a Debt Register

If we are serious about managing transformation portfolios as portfolios – with the same discipline we bring to financial or technology portfolios – then we need a mechanism for making transformation debt visible, measurable, and manageable.

The question is not whether the organisation is carrying transformation debt. Every organisation running a sustained change agenda is. The question is whether it knows how much, where the highest-interest items sit, and what the compounding effect is doing to its capacity for future change.

This does not require a new methodology. It requires an extension of the one we already have. Three practices would make a material difference.

The first is an absorption audit as a standard gate in programme closure. Before a programme is marked complete, an honest assessment — not of whether the outputs were delivered, but of whether the change was absorbed. Where was it absorbed fully? Where partially? Where not at all? The gap between delivery and absorption is the debt created by that programme, and it should be recorded as explicitly as any other programme output.

The second is a carrying-cost estimate for each item of unabsorbed change. This need not be precise — the soft costs of transformation debt resist exact quantification — but even a rough order of magnitude, reviewed quarterly, would transform the conversation. When a portfolio board can see that the unabsorbed elements of three closed programmes are collectively imposing an estimated fifteen percent drag on current programme velocity, the case for addressing the debt becomes concrete rather than philosophical.

The third — and politically the hardest — is a write-off mechanism. Some transformation debt will never be repaid. The operating model that was designed for a context that has since shifted. The process change that was overtaken by a technology deployment. The cultural initiative that quietly died. Carrying these items as though they might still be completed is dishonest accounting. Writing them off — formally acknowledging that the intended change will not be pursued and that the residual state is the new baseline — is the only way to clear the ledger and free the capacity that silent debt consumes.

The reasonable objection — that change absorption is a gradient, not a binary, and therefore resists the crisp measurement a debt metaphor implies — is precisely the argument that has protected the status quo. The same objection was raised about technical debt in its early years. What settled it was not precision but utility: rough measurement proved more valuable than no measurement at all. A portfolio that tracks transformation debt approximately is in a fundamentally stronger position than one that does not track it at all.

6. The Portfolio That Knows What It Owes

The metaphor of debt is not perfect — no metaphor is. Transformation debt does not behave with the mathematical regularity of financial debt, and its interest payments are harder to isolate. But the metaphor does essential work that no current portfolio framework performs: it names the phenomenon, it insists on measurement, and it creates the vocabulary for a conversation that most organisations are not yet having.

We are fluent in the language of programme delivery. We know how to plan change, govern change, and report on change. We are far less fluent in the language of change absorption — in understanding the difference between what we have delivered and what the organisation has actually become.

Until we develop that fluency, every portfolio review will present a picture that is technically accurate and substantively incomplete: a balance sheet that records the assets but omits the liabilities.

The transformation backlog is organisational debt. It is time we started accounting for it.