

BEYOND THE ENGINEERING LEDGER

Technical Debt Is Strategic Debt — Why the Post-Pandemic Reckoning Was Always Coming

The shortcuts that saved organisations in 2020 quietly mortgaged their futures — and the bill is falling due in the currency of foreclosed strategy, not slow releases.

Giovanni Leonardi**November 2021**

Technical Debt Is Strategic Debt — Why the Post-Pandemic Reckoning Was Always Coming

Transformation · No. 325

Originally written November 2021 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“We took the debt as engineers. We are paying it back as strategists.”

1. The nine-day miracle and its afterlife

In the second week of March 2020, a team I was close to stood up a customer portal in nine days. The same portal had sat inside a business case for eighteen months, blocked on ownership, security sign-off, and which budget line would carry it. Then the offices closed, the phone lines drowned, and the eighteen-month questions evaporated overnight. Nine days. Everyone was, rightly, proud.

Eighteen months on, that portal is still running, and the pride has curdled into something quieter. The integration behind it was never meant to survive contact with real life: customer records dropped into a spreadsheet, a macro reshaped them, and one analyst ran a reconciliation each evening to catch whatever the macro missed. It worked. It kept working. By the summer of 2021 that nightly reconciliation was moving tens of thousands of records a week, three downstream initiatives quietly leaned on it, and exactly one person truly understood how the mapping behaved. When that person handed in their notice — as so many have this year — the organisation finally learned what it had borrowed back in March 2020. Not nine days of engineering shortcut. A strategic position it could no longer change.

This is the part the technical-debt literature leaves out, and it is why the reckoning now arriving in so many organisations was never really avoidable. We took the debt as engineers. We are paying it back as strategists.

2. The metaphor that hides the bill

The idea of technical debt has served us well for thirty years. The original point was subtle and honest: shipping code you know to be imperfect is like taking on debt — acceptable, even wise, provided you understand you are borrowing and intend to repay before the interest compounds. Later writers sharpened it into a tidy quadrant — debt that is deliberate or inadvertent, prudent or reckless — and handed delivery teams a vocabulary for a conversation they had always struggled to have with the business.

But every metaphor smuggles in assumptions, and this one smuggles in three. It assumes the debt is recorded, the way a loan is recorded — that someone knows it exists. It assumes the debt is local, sitting inside a codebase a team owns and can refactor. And it assumes the currency of repayment is engineering effort — sprints spent on paydown instead of features. When all three hold, the metaphor is excellent and the remedy is disciplined refactoring.

The trouble is that the debt organisations took on in 2020 satisfies none of the three. It was not recorded — nobody wrote "surrendered our integration strategy" on a backlog. It was not local — it sprawled across departmental subscriptions, spreadsheets, and access arrangements that no single team owns. And its interest is not paid in slower releases. It is paid in options foreclosed: the initiatives you cannot start because something fragile and unowned now sits in the path. Call it what it is. This is not technical debt. It is strategic debt wearing technical clothing.

The most expensive debt an organisation took on during the pandemic was never written in code. It was the strategic optionality it quietly surrendered when it chose the fastest path under duress — and that debt appears on no engineering ledger.

3. The debt that never touched the codebase

Consider what actually happened in most organisations through 2020. Under genuine emergency, people did the resourceful thing: they bought a tool, wired up a workaround, stood up a service, and moved on to the next fire. Shadow IT — for years the thing security teams tried to stamp out — became, overnight, mainstream IT. It was how the work got done. And because it got the work done, the temporary quietly became permanent. Nobody re-decided.

That phrase is the heart of it. A deliberate technical debt carries an implicit promise: we will revisit this. The pandemic's debt carried no such promise, because there was no moment of decision to revisit — there was a reflex under fire. The reconciliation meant to last a fortnight was still running eighteen months later not because anyone decided to keep it, but because no one ever decided anything at all. The absence of a decision is itself the debt.

And here is why it is strategic rather than technical. When an unowned, half-understood component becomes load-bearing, it does not merely slow your releases. It takes choices off the table. You cannot renegotiate a vendor relationship once three processes depend on a feature only that vendor provides. You cannot consolidate four overlapping collaboration tools once each has grown its own gravitational field of workflows. You cannot pursue the data programme the board approved because the data it needs is trapped inside a spreadsheet nobody dares touch. None of that shows up as a slow sprint. All of it shows up as a strategy that has quietly narrowed.

4. The strongest case for doing nothing much

The honest objection to all of this deserves stating at full strength, because it is not foolish. It runs like this: technical debt is a normal, healthy feature of building software. Good teams take it on deliberately to move fast and pay it down deliberately when it starts to bite. The pandemic merely accelerated an ordinary cycle — more debt, taken faster. The discipline required is the discipline we already know: surface it, prioritise it against the roadmap, refactor steadily. There is no need for grand strategic language, only for the engineering maturity many organisations have always lacked. Reaching for the word "strategic," on this view, is just inflation.

I have real sympathy for that argument, and for the category of debt it describes it is exactly right. A team that owns its whole system, took its shortcuts knowingly, and can see them on its own board should absolutely treat them as engineering work. But that is precisely the category the pandemic's debt is not in. The objection assumes debt that is recorded, local, and team-owned; the debt I am describing is unrecorded, spread across the business, and owned by no one. You cannot prioritise on a backlog a thing that sits on no backlog. You cannot refactor your way out of a vendor dependency your engineers do not control. The disciplined-team answer is correct — about a different problem. Mistaking one problem for the other is exactly how the reckoning keeps getting postponed: the issue is handed to engineering, engineering reaches only the part it owns, and the strategic remainder sits untouched until it fails.

5. Why the reckoning was always coming

Deferral has a maturity date. That is the iron rule the sprint let everyone forget. A decision postponed under emergency is not a decision cancelled; it is a decision moved into the future, usually to a worse moment to have it. Three forces set that date, and all three came due in 2021.

- The temporary became structural. Eighteen months is long enough for a workaround to grow roots — dependencies, habits, and downstream processes that make removal a project in its own right. What could have been unwound in a week in mid-2020 now needs a business case of its own.
- The people who held it in their heads began to leave. The great churn of talent this year did not create the debt, but it called it in. Undocumented, un-owned infrastructure is survivable only while the person who built it stays. Their resignation letter is the statement of account.
- The buffers were already gone. For a decade we optimised for efficiency — just-in-time everything, slack engineered out as waste. The supply shocks of this year, from silicon to shipping, taught the operations side a lesson the technology side has yet to fully absorb: a system with no spare capacity has no capacity to repair itself either. There was no slack left to pay the debt down with.

Put those together and the timing was never mysterious. The sprint borrowed against a future that has now arrived. The reckoning is not bad luck; it is the maturity date on a loan taken in March 2020, coming due precisely on schedule.

"A decision postponed under emergency is not a decision cancelled. It is a decision moved to a worse moment to have it."

6. Rewriting the ledger

If the debt is strategic, the response cannot be purely technical — and the first move is to stop accounting for it as engineering effort and start accounting for it as foreclosed options. The board-level question is not "how much technical debt do we carry?" It is "which decisions did we defer in 2020, and what can we no longer do because of them?" That is a question a strategy function can own, and a backlog cannot answer.

Three disciplines follow from the reframing.

1. Re-decide, deliberately, what was never decided. Every piece of pandemic-era infrastructure deserves an explicit verdict — keep, replace, or retire — taken by whoever owns the business capability it serves, not inherited by whichever team happens to touch the code. The aim is not to purge the shortcuts; many were good choices. The aim is to convert a reflex into a decision, so that what stays, stays on purpose.
1. Locate ownership before you locate faults. Unowned infrastructure is the actual risk; the fragility is only its symptom. For each load-bearing workaround, name the person accountable for the capability, not merely the person who happens to understand the code. Ownership is what turns a single point of failure back into a managed asset.
1. Rebuild the buffer you spent. The pandemic was an uncontrolled experiment in what happens when every reserve is drawn down at once, and it returned a clear result: resilience carries a value the efficiency models had been pricing at zero. Restoring some deliberate slack — in capacity, in documentation, in people who understand more than one system — is not waste. It is the premium on an insurance policy the last two years proved you were carrying whether you paid for it or not.

None of this is exotic, and that is rather the point. The reckoning is painful not because the remedy is unknown but because we have been hunting for it in the wrong ledger. Treat the residue of the pandemic sprint as an engineering backlog and it will lose, every quarter, to the roadmap — as it has been losing all year. Treat it as what it actually is, the sum of strategic decisions an organisation deferred under fire, and it becomes visible at last to the people who can act on it.

We were told, through 2020, that the crisis had finally made our organisations fast. It had. What we are learning through 2021 is the other half of that sentence: speed bought on credit still has to be paid for, and the currency is not code. It is the range of futures still open to us. That bill was always going to arrive. It is arriving now.