

BEYOND THE FRONT END

The Miracle Was a Deployment, Not a Transformation

The channel went live in weeks; the operating model behind it never moved. A practitioner's view, from the middle of 2020, on what forced digital adoption actually accelerated — and what it merely postponed.

Giovanni Leonardi**August 2020**

The Miracle Was a Deployment, Not a Transformation

Transformation · No. 248

Originally written August 2020 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“What we compressed was the front end; what we postponed was everything behind it.”

1. The exclamation-mark email

Somewhere in April a colleague forwarded me a screenshot of a celebration. A digital servicing journey that had sat on our roadmap for three years — costed, gated, business-cased at something north of six million pounds — had gone live in a little under seven weeks. The email had exclamation marks in it. Underneath the screenshot someone had typed a single line: turns out we could do it all along.

I have thought about that sentence more than any other this year, because it is half right, and the half that is wrong is the half that is going to cost us.

We are all telling a version of the same story this summer. The five-year plan that delivered in five months. The transformation the crisis finally forced. The resistance that melted the moment there was no alternative. It is a good story, and the genuinely remarkable thing is that most of it is true — what we built, we built, and we built it fast. The question I want to sit with is narrower and more uncomfortable: what, exactly, did we compress? Because I do not think it was transformation. I think it was deployment. And the distance between those two words is where the next three years of trouble is quietly accruing.

2. What actually moved fast

Let me be precise about the example, because precision is the whole argument. When the lockdown landed and payment deferrals became a matter of regulatory expectation almost overnight, the demand did not arrive as a project. It arrived as a queue. Contact centres that had been sent home to spare bedrooms and kitchen tables were taking call volumes two and three times their planning assumptions, and the wait times were becoming a headline risk in their own right.

So we built a form. A self-service journey where a customer could request a payment holiday without waiting forty minutes on a line. It went live in about five weeks, against an original plan that had assumed the better part of a year, and on launch day it absorbed several thousand requests that would otherwise have sat in that telephone queue. That

was real. I would not take it away from anyone who worked those weeks.

But a payment deferral is not a form. It is a decision. It has to reckon with joint accounts where only one party has asked; with customers already in arrears; with the vulnerable-customer flags that oblige a human conversation; with what the deferral does to a credit file and how that gets explained. None of that was in the five-week build, because none of that is visible on a screen. Of the requests that came through the shiny new channel in the first fortnight, close to two in five could not be resolved by the channel and fell out to manual handling — to colleagues adjudicating case by case from those same kitchen tables, over a strained connection, working from a spreadsheet stood up in a hurry because the case-management system had never been part of the scope.

We did not deliver a five-year roadmap in five months. We delivered the first slide of it, and left the other forty to be discovered later, under load, by whoever happened to be on shift.

That is the pattern, and once you see it you see it everywhere this year. What we compressed was the front end; what we postponed was everything behind it. The visible, demonstrable, screenshot-able front end moved at a speed we had sworn for a decade was impossible. The operating model behind it — the decision rights, the data, the exception paths, the servicing tail — did not move at all. In most cases we did not touch it. We routed around it with willing people and improvised tooling, and we called the result a transformation because the part we could photograph looked finished.

3. Why the textbooks had the constraint wrong

For twenty years the received wisdom has told us that the binding constraint on change is delivery capacity and technology lead time — that things take as long as they take because integration is hard, testing is slow, and the release calendar is full. The whole apparatus of stage gates and phased business cases is built on that premise. What this year revealed is that the premise was, at best, half the story.

What the roadmap assumed	What the crisis revealed
The binding constraint was delivery capacity and technology lead time	The binding constraint was permission — who was allowed to say yes, and to how much risk
Governance de-risks change	With the gates suspended, change went faster and, for a while, no worse
The business case authorises the work	The work happened with no updated business case at all; the authorisation was survival
Scope must be fixed and defended	Scope collapsed to a single, universally agreed objective and everything else fell away

The reason the form shipped in five weeks was not that we discovered a new method of building software. It shipped because three conditions held at the same time. There was a

single, unambiguous objective that nobody in the building disputed. The permission chain — the boards, the sign-offs, the design authority, the change-advisory ritual — was suspended, because there was no time for it and no appetite to defend it. And the risk appetite was, for those weeks, effectively unlimited, because the counterfactual was worse than any failure the change might cause. Those are not capabilities we acquired. They are circumstances that were imposed on us. We did not get better at transformation. We were briefly relieved of the things that normally slow it down.

4. The strongest version of the other view

The obvious objection is a good one, and I want to put it at full strength rather than knock down a weak version of it. It runs like this: the muscle memory has changed, permanently and for the better. Organisations that spent a decade insisting change had to be slow have now watched themselves move at a speed they swore was impossible, and that knowledge cannot be un-known. The old excuses are dead. Every "we can't possibly do that before next year" has been exposed as a choice dressed up as a constraint. On this reading, speed is the transformation, and my worrying about the plumbing behind the screen is the very caution the crisis just discredited.

There is real truth in this, and I want to grant as much of it as I honestly can. The deployment muscle is real. The willingness to ship a working front end in weeks rather than defend a release date is a genuine gain, and the death of the reflexive "not possible" is worth more than most things on any transformation roadmap. If we lose those in the return to normal, we will have wasted the whole miserable experience.

"Speed was not a method we learned. It was a permission we were granted, once, by a threat none of us would wish back."

But here is the part I cannot grant. The three conditions that produced the speed do not generalise, and cannot be manufactured on demand. You cannot conjure a single, universally agreed objective for the ordinary backlog, where the whole point is that reasonable people want different things. You cannot suspend the permission chain for a change that is not existential — nor, on reflection, should you, because most of what the gates catch is real. And the parts that were genuinely hard were not made easier by the crisis; they were deferred by it. The operating model, the data foundations, the exception handling — these did not get solved in the scramble. They got postponed, and then buried under a layer of things that look complete.

5. The honest version of the story

The answer is not to slow back down and reinstate the ceremony for its own sake. That would be to learn exactly the wrong lesson. The answer is to stop conflating two things

we have spent this summer treating as one.

- Keep the deployment muscle. The willingness to ship the visible thing in weeks is a real and repeatable gain, and it should be defended against the quiet return of the gate that exists only to be attended.
- Name the debt while you still remember incurring it. Every one of these five-week channels has a servicing tail behind it that was improvised by people under pressure. That improvisation is now load-bearing, and the temporary has a way of becoming the architecture simply because no one wrote down that it was meant to be temporary.
- Do the unglamorous half deliberately. The operating model, the data, the exception paths were postponed, not solved, and they will not respond to a burning platform — because, mercifully, there isn't one. They need the patient, funded, unspectacular attention that no crisis will ever supply.

The colleague who wrote turns out we could do it all along was right about the channel and wrong about the "all". We could stand up the channel all along; the thing that stopped us was never the technology, and it is genuinely useful to have that proven in public. But what we still cannot do at speed — what we may have just made harder, by hiding it beneath a layer of things that photograph well — is the slow work behind the screen. The screenshot was never the transformation. And the sooner we admit that the miracle was a deployment, the sooner we can get on with the transformation it postponed.