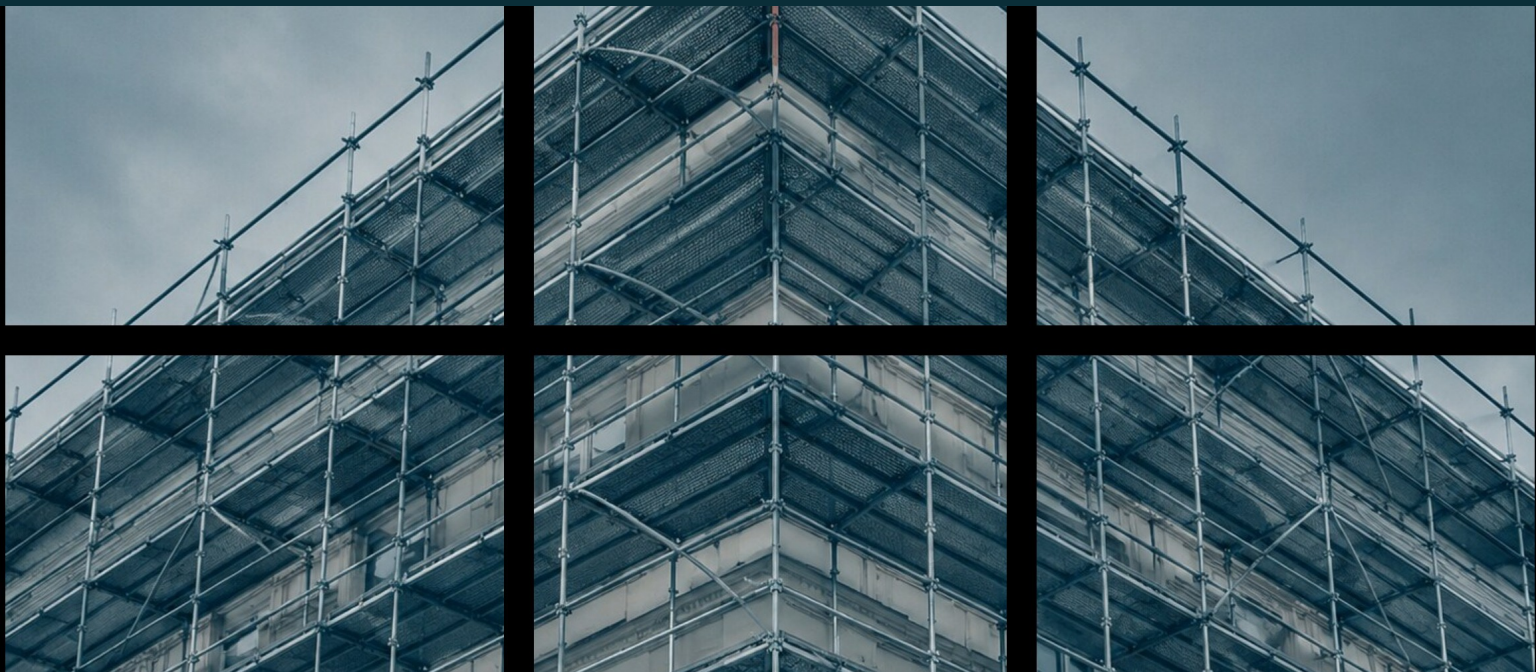


BEYOND THE PROJECT CHARTER

Product Thinking Before Anyone Called It That

A Practitioner's Perspective on the Quiet Shift from Projects to Products in Enterprise
Delivery



Giovanni Leonardi

January 2012

Product Thinking Before Anyone Called It That

Transformation · No. 210

Originally written January 2012 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

*“The best delivery teams had already stopped thinking in projects
– they just had not yet found the vocabulary to explain what they
were doing instead.”*

The Shift No One Named

Something has been changing in how the most effective delivery organisations think about their work, and it does not yet have a settled name. The shift is subtle but consequential: a move away from the project as the primary unit of delivery and toward something more persistent, more iterative, and more closely tied to the outcomes that actually matter to the business.

The traditional model is familiar to anyone who has spent time in programme management. Work is conceived as a project: it has a start date, an end date, a budget, a scope, and a team assembled for the purpose. When the project ends, the team disperses. The deliverable is handed to operations. The business case is closed. Success is measured by whether it was delivered on time, on budget, and to specification.

This model has served well for certain kinds of work – infrastructure builds, regulatory implementations, system replacements with hard deadlines. But for the growing class of technology-enabled capabilities that organisations now depend on, it is increasingly mismatched. These capabilities do not have end dates. They evolve. They require continuous attention, continuous improvement, and a persistent team that understands both the technology and the users deeply enough to make good decisions quickly.

What the Best Teams Already Know

The pattern I have observed is that the best delivery teams have already made this shift intuitively, even where the organisational structures around them have not caught up. They think of their work not as a project to be completed but as a capability to be grown. They maintain a persistent backlog tied to user needs rather than a requirements document tied to a business case. They measure themselves by adoption and impact rather than by milestone delivery.

They have, in effect, become product teams – though most would not use that term, and their organisations certainly do not recognise them as such.

The gap is not in what teams are doing – it is in what organisations are willing to fund, govern, and sustain. The project model persists not because it is effective but because it is legible to the finance function and the PMO.

This is the crux of the problem. Organisations still fund projects, not products. They still govern through stage gates and business cases designed for finite endeavours. They still measure delivery teams by whether they hit a date rather than whether they moved a metric. The teams that have shifted their thinking are operating inside structures that actively resist what they are trying to do.

Why This Matters Now

The reason this observation matters is that the mismatch is becoming expensive. Organisations are running projects to build capabilities that then atrophy because there is no funding model for ongoing evolution. They are assembling teams, building knowledge, and then dispersing that knowledge when the project closes — only to reassemble a new team months later when the next phase is approved. The waste is enormous and largely invisible, because it is built into the project model itself.

The organisations that will navigate the next decade most effectively will be those that find a way to fund persistent teams around persistent capabilities — to move, in other words, from a project economy to something more like a product economy. The teams already know this. The question is whether the governance, finance, and leadership structures around them will catch up before the cost of not doing so becomes impossible to ignore.