



BEYOND THE END DATE

When Projects Stopped Having End Dates — Programme Management and the Emerging Product Model

A Practitioner's Perspective on the Collision Between Project Governance and Continuous Delivery

Giovanni Leonardi

June 2013

Contents

1. Executive Summary	3
2. The Project Model and Its Assumptions	3
3. The Product Turn	4
4. The Governance Collision	4
4.1 Funding	5
4.2 Progress Measurement	5
4.3 Completion and Benefits	5
5. The Uncomfortable Middle Ground	6
6. The Skills Challenge	6
7. What This Means for Programme Management	7

When Projects Stopped Having End Dates — Programme Management and the Emerging Product Model

Programme · No. 175

Originally written June 2013 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“The question is no longer when will this project finish, but whether finishing is even the right concept for the work we are trying to govern.”

1. Executive Summary

Something is shifting in how large organisations structure their technology-enabled work, and programme management has not yet caught up. Across financial services, telecommunications, and retail, a growing number of delivery teams are moving away from the traditional project model — with its defined scope, fixed timeline, and formal closure — towards a product model, where a persistent team owns a capability and evolves it continuously. This shift, driven by agile adoption, the influence of digital-native companies, and the practical reality that many technology capabilities now require ongoing development rather than one-off delivery, is creating a fundamental tension with the programme management structures that govern enterprise change. This essay explores the nature of that tension, why it is proving so difficult to resolve, and what it means for the future of programme governance.

2. The Project Model and Its Assumptions

The project, as a unit of organisational work, carries a set of assumptions so deeply embedded that they are rarely examined. A project has a beginning and an end. It has a defined scope that can be baselined and controlled. It consumes a finite budget against which progress can be measured. It is staffed by people who are assigned for its duration and who will, upon completion, move to other work. And it produces deliverables that are handed over to an operational function for ongoing maintenance.

These assumptions have served programme management well for decades. The disciplines of scope management, schedule control, earned value analysis, and stage-gate governance all depend on them. The entire apparatus of programme reporting — RAG statuses, milestone tracking, benefits realisation plans — is built on the premise that work can be decomposed into bounded projects with measurable progress towards defined completion criteria.

This model works when the work genuinely has a natural end point. Building a bridge, implementing a regulatory change, migrating from one platform to another — these are projects in the truest sense. They have a scope that can be defined, a state of done that can be recognised, and a handover that makes sense.

The problem is that an increasing proportion of the technology-enabled work in large organisations does not fit this description.

3. The Product Turn

The shift towards product thinking in enterprise technology has multiple roots, and it is worth understanding them because they explain why the trend is structural rather than fashionable.

The first root is agile software development. As organisations have adopted iterative delivery methods — Scrum, Kanban, and their variants — they have discovered that the most effective delivery pattern is not a series of discrete projects but a continuous flow of incremental improvements delivered by a stable team. The team builds deep knowledge of the domain, the codebase, and the stakeholders, and this accumulated knowledge is a significant source of productivity. Disbanding the team at the end of a project and reassembling a new one for the next phase destroys this knowledge and imposes a substantial ramp-up cost.

The second root is the nature of digital services. A customer-facing digital channel — a mobile application, an online platform, a digital service — is not something that is built and then maintained. It is something that is continuously evolved in response to user behaviour, competitive pressure, and changing expectations. The idea that a mobile banking application could be delivered as a project and then handed to a maintenance team is, in practice, absurd. The application requires continuous development to remain competitive, and the pace of that development is a strategic variable.

The third root is the influence of the technology companies that have come to dominate their sectors. Amazon, Google, and their peers do not run projects in the traditional sense. They organise around products owned by small, persistent teams with end-to-end accountability. The success of this model has not gone unnoticed by the enterprises that compete with them or aspire to their operational discipline.

- Stable teams accumulate knowledge that is destroyed by project-based staffing models
- Digital services require continuous evolution, not one-off delivery and handover
- The most successful technology organisations have demonstrated the superiority of persistent product teams

4. The Governance Collision

The tension between product-based delivery and programme governance is not theoretical. It is playing out in real organisations, and it is generating friction that is

both predictable and poorly understood.

4.1 Funding

Programme governance allocates funding to projects. A project has a business case, a defined scope, and a budget that is approved through a capital allocation process. When the project is complete, the funding stops. Ongoing costs are operational expenditure, managed through a different process with different approval thresholds.

Product teams do not fit this model. They require persistent funding — a standing investment in a capability that is expected to deliver value continuously rather than through a single bounded initiative. The business case is not for a defined deliverable but for a sustained capability, and the value is realised incrementally rather than at a single point of completion.

Most organisations' financial governance is not designed for this. Capital and operational expenditure distinctions, annual budget cycles, and project-based approval processes all assume bounded work. Programme managers find themselves writing artificial business cases — packaging continuous product development as a series of sequential projects to fit the governance model, each with a manufactured scope and an arbitrary end date.

4.2 Progress Measurement

Programme reporting relies on measuring progress against a plan. Milestones are set, progress is tracked, and deviations are escalated. This works when the scope is defined and the sequence is known.

Product teams measure progress differently. Their metrics are outcomes — user adoption, transaction volumes, error rates, customer satisfaction — rather than deliverables. They do not have a plan in the traditional sense but a backlog that is continuously reprioritised. The question is not whether they are on track against a baseline but whether they are delivering value at an acceptable rate.

Programme governance structures struggle with this. A RAG status requires a baseline to measure against. A milestone tracker requires milestones. Earned value analysis requires a scope baseline and a budget at completion. None of these instruments work naturally with a continuous delivery model.

4.3 Completion and Benefits

Perhaps the deepest tension is around the concept of completion itself. Programme governance assumes that work completes, benefits are realised, and the programme closes. Benefits realisation reviews assess whether the promised outcomes were delivered.

In a product model, there is no completion. The product team exists for as long as the product is strategically relevant. Benefits are not realised at a point in time but accrue continuously. The question of whether the investment is justified is answered not by a post-implementation review but by ongoing performance metrics.

This is not merely an administrative inconvenience. It challenges the fundamental accountability model that programme governance provides. If a project overruns or fails to deliver its benefits, there is a clear accountability chain. If a product team is delivering continuously but the rate of value creation is declining, the accountability framework is less clear.

5. The Uncomfortable Middle Ground

What makes this transition so difficult is that most large organisations cannot simply abandon project-based governance and adopt a pure product model. They operate in a hybrid world where some work is genuinely project-shaped and some is genuinely product-shaped, and the governance framework needs to accommodate both.

The question is not whether the product model will displace the project model — in many contexts, it already has. The question is how programme governance adapts to a world where both models coexist, and where the boundary between them is neither clean nor stable.

The organisations I observe handling this best are those that resist the temptation to force one model onto all work. They are developing governance frameworks that distinguish between bounded change — work with a genuine scope, timeline, and completion point — and continuous capability development. They apply different funding models, different progress metrics, and different accountability structures to each.

This is harder than it sounds. It requires programme managers to develop new skills and new instincts. It requires finance functions to accept persistent investment models alongside project-based capital allocation. It requires boards and executive committees to evaluate product performance through outcome metrics rather than milestone achievement. And it requires a degree of intellectual honesty about which work is genuinely project-shaped and which is being artificially constrained into a project structure for governance convenience.

6. The Skills Challenge

The shift also exposes a skills gap in the programme management profession itself. Traditional programme management training and certification — MSP, PgMP, and their equivalents — are built on the project model. They teach disciplines that are valuable but incomplete in a world where a significant proportion of the work being governed does not behave like a project.

Programme managers increasingly need to understand product management concepts: roadmaps rather than plans, outcomes rather than outputs, continuous prioritisation rather than scope control. They need to be comfortable with a model where the answer to the question when will this be done? is it will never be done, and that is by design.

This is a significant cultural shift for a profession that has been built on the disciplines of planning, control, and closure. It does not mean those disciplines are obsolete — they remain essential for genuinely bounded work. But it does mean they are insufficient for the full range of work that modern programmes encompass.

7. What This Means for Programme Management

The shift from projects to products is not a passing trend. It is being driven by forces — the economics of software delivery, the nature of digital services, the competitive pressure from digital-native organisations — that are structural and accelerating. Programme management as a discipline will need to evolve to accommodate it, or risk becoming relevant only to a shrinking subset of organisational work.

The evolution is not about abandoning rigour. It is about applying the right kind of rigour to the right kind of work. Bounded change still needs the disciplines of scope management, milestone tracking, and benefits realisation. Continuous capability development needs different disciplines — outcome measurement, investment rate governance, and portfolio-level prioritisation.

The programme managers who will thrive in this environment are those who can hold both models in their heads simultaneously, who can design governance frameworks that accommodate genuine hybridity, and who can resist the institutional pressure to force all work into a single model for the sake of reporting consistency.

*“The question is no longer when will this project finish,
but whether finishing is even the right concept for the
work we are trying to govern.”*

The profession is at an inflection point. The project model served us well when the work was genuinely project-shaped. As the nature of the work changes, the governance must change with it — not by abandoning discipline, but by recognising that discipline takes different forms for different kinds of work.