

BEYOND VELOCITY

Benefits in Agile Delivery — Continuous Value or Continuous Hope?

A Practitioner's Perspective on Benefits Management in Iterative Programme Delivery



Giovanni Leonardi

August 2006

Benefits in Agile Delivery – Continuous Value or Continuous Hope?

Programme · No. 57

Originally written August 2006 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“Agile promised to deliver value early, but in most programmes it simply delivered outputs early – the question of whether those outputs translated into benefits was left to someone else, somewhere later.”

The Promise and the Gap

The appeal of agile delivery within programme environments is straightforward: deliver in short cycles, get feedback early, and course-correct before the investment is spent. In principle, this should make benefits realisation easier, not harder. If you are delivering working capability every few weeks, you have more frequent opportunities to ask whether value is being created.

In practice, I have observed something quite different. The adoption of agile methods within larger programmes has, in many cases, made the benefits question harder to answer – not because the delivery is worse, but because the mechanisms for tracking benefits were designed for a different rhythm entirely.

Where the Disconnect Lives

Traditional benefits management assumes a linear sequence: define benefits at the outset, deliver the capability, then measure whether the benefits arrived. The entire framework depends on a stable scope baseline against which realisation can be assessed. Agile delivery, by design, rejects that stability. Scope is deliberately fluid. Priorities shift between iterations. Features are added, deferred, or dropped based on emerging understanding.

This is good practice from a delivery perspective. But it creates a genuine problem for benefits tracking. If the scope changes every sprint, against what baseline are benefits measured? If a feature that was central to the original benefits case is deprioritised in favour of something the product owner considers more urgent, who notices that the benefits profile has shifted?

The pattern I have observed across several programmes is that agile delivery proceeds with energy and discipline at the team level, while benefits management – to the extent it exists at all – remains anchored to the original business case, growing steadily more disconnected from what is actually being built.

Agile delivery and benefits management are operating on different clocks, with different vocabularies, and answering different questions. The result is a programme that delivers continuously but cannot say what it has delivered in terms that matter to the investment decision.

The Velocity Illusion

There is a particular trap that agile programmes fall into when reporting upward. Velocity — the rate at which the team completes work — becomes the proxy for value. Boards and sponsors see a team completing stories, burning down backlogs, and shipping releases, and they conclude that value is being delivered. The dashboard is green. Progress is visible.

But velocity measures effort consumed, not value created. A team can maintain excellent velocity while delivering features that contribute nothing to the benefits case. The product owner may be optimising for user requests, technical debt reduction, or stakeholder appeasement — all legitimate concerns — without any reference to the benefits that justified the programme's existence.

This is not a failure of agile. It is a failure to connect agile delivery to the investment logic that funds it. The methodology is working exactly as designed. The problem is that nobody has extended the design to include benefits.

What the Textbooks Miss

The emerging literature on agile programme management tends to treat benefits as something that will naturally emerge from iterative delivery. The argument is intuitive: if you are continuously delivering what users need, benefits will follow. This is sometimes true. But it relies on two assumptions that frequently do not hold.

First, it assumes that what users request is aligned with what the organisation needs. In practice, user priorities and strategic benefits are often different things. A sales team may want a reporting feature that makes their weekly meeting easier; the benefits case may depend on improved conversion rates that require a fundamentally different capability.

Second, it assumes that benefits are incremental — that each iteration delivers a small slice of value that accumulates over time. Some benefits do work this way. But many of the most significant transformation benefits are threshold-dependent: they only materialise when a complete capability is in place. Process efficiency gains, for example, often require end-to-end redesign. Delivering half the redesign does not deliver half the benefit; it may deliver no benefit at all, or even negative benefit if the partial change disrupts existing workflows.

A Practitioner's Honest Assessment

I do not think agile delivery is the wrong approach for programme environments. The feedback loops are valuable. The adaptability is genuine. The alternative — rigid waterfall delivery against a scope baseline that was wrong from the start — is demonstrably worse

in most contexts.

But I think the profession has been insufficiently honest about what agile does and does not solve. It solves the delivery problem: how to build the right thing, iteratively, with feedback. It does not solve the benefits problem: how to ensure that what is built translates into the outcomes that justified the investment. These are different problems, and treating the first as a solution to the second is a category error that too many programmes are making.

What is needed is a benefits approach that can operate at the same cadence as agile delivery — one that tracks benefit indicators iteration by iteration, flags when scope changes affect the benefits profile, and gives sponsors visibility into whether the programme is converging on value or simply converging on completion. This does not yet exist as a mature practice. Until it does, agile delivery in programme environments will continue to promise continuous value while delivering, in too many cases, continuous hope.