

BEYOND THE AGILE TEAM

Agile Teams Cannot Outrun an Un-Agile Organisation

Why short delivery cycles fail inside long decision cycles



Giovanni Leonardi

July 2011

Agile Teams Cannot Outrun an Un-Agile Organisation

Programme · No. 160

Originally written July 2011 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“Local agility without organisational change eventually becomes a source of frustration rather than performance.”

1. The Local Success That Becomes an Organisational Failure

In one composite programme, a product team of twelve people had become the organisation's favourite Agile example. Work was visible on a wall. The team met every morning for fifteen minutes. Software that once arrived in quarterly bundles was now demonstrated every fortnight. Defects were found earlier, business representatives could change priorities, and the atmosphere around delivery had improved.

Yet the programme as a whole was not moving faster.

The team still waited six weeks for a server, eight weeks for a procurement decision, and a month for a security review. Its business representative could reorder stories but could not decide whether a policy should change. Funding had been approved for a fixed list of outputs, so every change in priority required an explanation to a steering committee that met monthly. The team had learned to work in short cycles inside an organisation that still made decisions in long ones.

This is the central limitation of the current enthusiasm for Agile methods. A team can become adaptive without making the organisation adaptive. When that happens, local improvement does not remove delay; it merely reveals where delay has moved.

2. What the Team Actually Changed

The strongest Agile teams alter the economics of learning. They make unfinished work visible, reduce the size of each commitment, and create regular opportunities to test whether an assumption is true. That matters because programme failure rarely begins with an inability to execute a known answer. It begins with confidence in an answer that has not been tested.

A fortnightly demonstration can expose a misunderstood requirement before months of effort accumulate behind it. A prioritised backlog can concentrate scarce capacity on the most valuable work. A retrospective can make recurring friction discussable. These are substantial gains, not fashionable rituals.

But their reach is limited. The team controls how it organises work within its boundary. It usually does not control the boundary itself.

- It cannot release value if operations accepts changes only in quarterly windows.
- It cannot act on customer evidence if the approved business case treats scope as a promise.

- It cannot resolve a policy question if authority remains three committees away.
- It cannot reduce a supplier dependency if the contract rewards document completion rather than working outcomes.

The organisation may therefore congratulate the team for becoming Agile while preserving every mechanism that makes agility difficult.

3. The Queue Outside the Team

Most executives look for delay in the place where people are visibly working. They count coding days, testing days, or the number of stories completed. The more important delay often sits between activities: waiting for approval, waiting for an environment, waiting for a decision, waiting for another supplier, waiting for a senior person to attend a meeting.

In the composite programme, one customer-facing change required roughly nine working days of team effort. It then spent thirty-one working days in queues. No individual queue appeared unreasonable. Architecture wanted consistency. Security wanted evidence. Operations wanted stability. Finance wanted control over the approved scope. Each function could defend its own requirement. Taken together, they made a short-cycle delivery model largely ceremonial.

The problem was not that these controls existed. It was that nobody governed their combined effect on the flow of value.

This distinction matters. Agile advocates sometimes describe every control as bureaucracy, while control functions sometimes describe every request for speed as recklessness. Both positions are convenient. The real management task is to decide which risks require prior approval, which can be managed through standards, and which can be detected quickly after action.

Without that decision, work moves rapidly until it reaches the edge of the team, then stops.

4. The Strongest Case for Keeping the Organisation as It Is

There is a serious argument against extending Agile practices beyond delivery teams. Large organisations cannot be run as collections of autonomous experiments. They carry legal obligations, operational dependencies, reputational exposure, and commitments to customers that no single team can see in full. Shared architecture prevents every project from solving the same problem differently. Financial governance prevents enthusiasm from consuming unlimited capital. Independent assurance exists because teams are not always the best judges of their own risk.

This argument is correct as far as it goes.

A bank cannot allow a team to reinterpret a control because it slows a sprint. A public body cannot treat procurement law as an impediment to be worked around. An enterprise with hundreds of systems cannot permit every product group to select its own infrastructure without regard to supportability. Some decisions must remain outside the team, and some changes must be slower than the people making them would prefer.

The mistake is to conclude that necessary control requires the present form of control. A monthly committee is not inherently safer than a delegated decision with clear limits. A large requirements document is not inherently more accountable than a sequence of tested outcomes. Annual funding does not become prudent merely because it is familiar. Control is valuable when it changes a decision or reduces an exposure; otherwise it is only delay wearing formal clothes.

The answer is not unrestricted team autonomy. It is a more deliberate distribution of authority.

5. Agility Is a Property of the Decision System

An organisation becomes more agile when it can make consequential decisions at the rate that evidence changes. That requires more than a delivery method. It requires alignment between funding, governance, architecture, operations, suppliers, and executive attention.

Consider the difference between two steering conversations.

In the first, the team reports that it completed twenty-seven points, explains why three items slipped, and seeks approval for a change request. The committee receives a great deal of delivery information but makes no strategic decision.

In the second, the team shows that customer uptake is lower than expected, identifies the assumption that failed, and offers three choices: continue, change the proposition, or stop. The committee decides within agreed financial and risk boundaries. The second conversation may contain fewer metrics, but it creates more agility because evidence reaches authority without losing weeks in translation.

The practical test is simple: when a team learns something important on Tuesday, how long does the organisation take to act on it?

If the answer is a month, the organisation is working in monthly cycles no matter how many daily stand-ups it holds.

6. Four Structural Changes That Matter

The first change is to fund a problem or outcome in stages rather than promise a complete solution at the outset. This does not remove the business case. It makes the case subject to evidence. Initial funding should purchase enough delivery to test the critical assumptions. Further commitment should depend on what has been learned.

The second is to define decision rights before delivery begins. Teams need to know which choices they can make, which require consultation, and which require approval. The limits should be based on exposure, not organisational rank. A low-risk decision should not climb a hierarchy merely because several functions have an interest in it.

The third is to bring control expertise into the flow of work. Security, architecture, operations, finance, and legal specialists should help shape options early instead of inspecting completed work at the end. Their independence need not disappear. But independence is not the same as distance.

The fourth is to measure elapsed time from idea to usable outcome, including every wait. Team velocity can help a team plan, but it says little about organisational responsiveness.

A programme that completes more stories while value remains trapped before release is improving the wrong number.

These changes are less visible than a wall of cards and more difficult than training a delivery team. They disturb budgets, committee mandates, professional boundaries, and senior habits. That is precisely why they matter.

7. A Warning About Scaling the Rituals

The predictable response to successful pilots is to spread their ceremonies. More teams are trained. Common templates are introduced. Reporting begins to include burndown charts. Senior managers ask whether every project has a backlog.

This can create the appearance of organisational adoption while leaving the operating model untouched. Worse, the new method may be absorbed into the old one. A team is asked to produce a fixed plan, fixed scope, and fixed date, then report its two-week progress against commitments made before learning began. The vocabulary changes, but the contract with uncertainty does not.

Agile methods become another reporting layer.

A useful scaling question is therefore not, "How many teams are using the method?" It is, "Which organisational decisions now happen differently because teams are learning faster?" If funding decisions, priority decisions, release decisions, and stop decisions remain unchanged, the transformation has not travelled far.

8. The Leadership Obligation

Senior leaders often sponsor Agile adoption as a delivery improvement and delegate it to technology or programme management. That framing protects the rest of the organisation from examination. It suggests that teams must learn a new way of working while executives may continue to fund, govern, and measure them in the old way.

Local agility without organisational change eventually becomes a source of frustration rather than performance.

The capable team sees the obstruction clearly. It can name the queue, quantify the wait, and often suggest a remedy. But if it lacks authority to change the surrounding system, visibility turns into helplessness. The best people either stop raising the issue or leave. Management then concludes that the method failed to scale.

Leadership must accept a reciprocal obligation. If teams are expected to expose uncertainty early, senior forums must be prepared to decide earlier. If teams are expected to change direction on evidence, funding and supplier arrangements must permit direction to change. If teams are accountable for outcomes, they must have enough authority to influence them.

9. Beyond the Agile Team

The rise of Agile delivery has given organisations a valuable instrument for seeing work and learning sooner. It has also exposed a harder truth: many delivery problems are not team problems at all.

They are consequences of how authority is distributed, how money is committed, how risk is interpreted, and how functions protect their own objectives. A better team can illuminate those structures, but it cannot redesign them by itself.

The next stage of Agile adoption should therefore be judged beyond the team boundary. Look at the age of decisions waiting in queues. Look at the time between evidence and executive action. Look at how often a programme can stop work that no longer justifies its cost. Look at whether control specialists help create a safe path or merely approve one after it has been built.

An organisation is not agile because its teams move quickly. It is agile when the whole decision system can learn, choose, and act before the opportunity—or the warning—has passed.