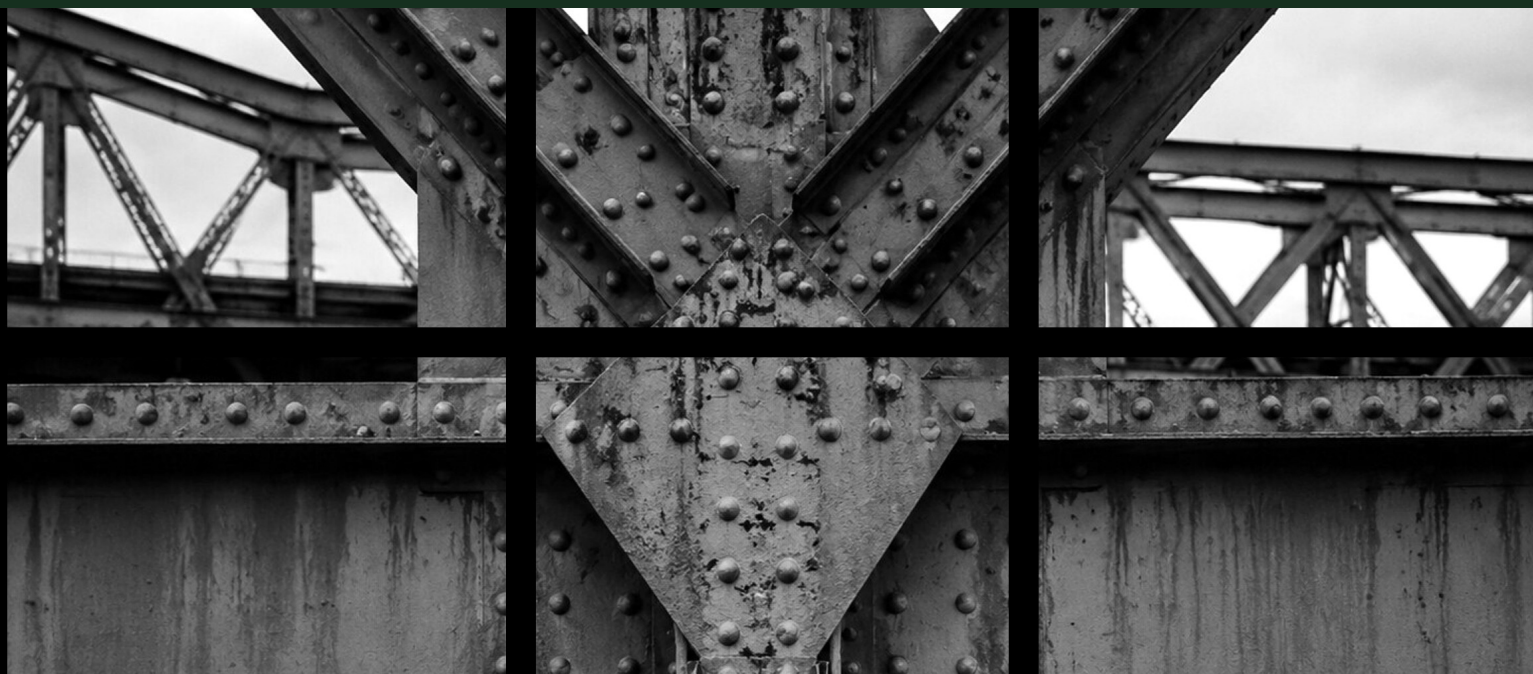


BEYOND VELOCITY

Sprint Velocity Is Becoming the New Status Report — and Recreating the Control Agile Was Meant to Escape

Why a local planning measure is acquiring the authority of programme assurance



Giovanni Leonardi

April 2012

Sprint Velocity Is Becoming the New Status Report – and Recreating the Control Agile Was Meant to Escape

Programme · No. 166

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

“The dashboard improved. The release did not.”

The number reached the board before the software did

A programme review this spring presented four delivery teams on a single page. One showed a velocity of 31, another 24, a third 18. The first was coloured green, the last amber. No working software was demonstrated. No one asked whether the teams estimated work in the same way.

What caught our attention was not the arithmetic. It was how quickly sprint velocity had acquired the authority once given to the percentage-complete line in a status report.

Velocity was intended as a local planning aid: a team's recent pattern of completing estimated work could help it judge how much to take into the next sprint. It was useful precisely because it belonged to the team, reflected its own estimation habits and improved through experience. It was never designed to compare teams, measure productivity or assure a steering committee that delivery was on track.

Yet that is exactly what is beginning to happen.

Familiar control has found a new vocabulary

Many organisations have adopted iterative delivery without changing the management reflex beneath it. Senior leaders still want one number that rises, can be placed beside a target and appears to answer the question: Are we getting enough for the money?

Sprint velocity looks ideal. It is numerical, frequent and apparently connected to output. Unlike a narrative status report, it seems objective. Unlike a demonstration, it fits neatly into a pack.

The mechanism is predictable:

- teams estimate backlog items in relative points;
- completed points are totalled at the end of the sprint;
- a target velocity is inferred from the first few iterations;
- actual velocity is compared with that target;
- management begins to treat the gap as a performance variance.

At that moment, a planning measure becomes a status measure. Soon afterwards, it becomes a behaviour-shaping target.

A composite software programme illustrates the cost. A six-person team averaged 22 points across its first four sprints. A recovery plan then set a target of 28. By the seventh sprint, reported velocity had reached 30. The apparent improvement came from splitting work more finely, revising estimates upward and accepting several items whose testing continued into the following sprint.

The dashboard improved. The release did not. Three unresolved integration defects delayed the next demonstration by nine working days.

Nothing in the figures was necessarily dishonest. The team had simply learned what the organisation valued and adjusted the local language of estimation to satisfy it.

When velocity becomes a target, it stops describing delivery and starts describing the pressure placed on estimation.

The serious case for reporting velocity

There is a reasonable objection. Boards cannot attend every sprint review, and large programmes need some way to see whether delivery capacity is stable. A falling velocity can expose interruption, weak backlog preparation, technical difficulty or loss of experienced people. Ignoring it entirely would discard useful evidence.

That case is sound. The mistake is not reporting velocity; it is allowing velocity to stand alone or travel without its conditions.

A velocity of 30 is meaningful only against the same team's estimation practice, definition of done, mix of work and recent history. It says little about another team's 20. It says even less about whether the programme is solving the right problem, controlling quality or approaching a viable release.

Leaders should therefore ask for evidence that preserves the measure's context:

- the trend for the same team, not a league table across teams;
- working software demonstrated against a real user need;
- unfinished work and defects that sit outside the headline total;
- material dependencies and decisions that may constrain the next sprint;
- progress towards a release outcome, not merely consumption of backlog points.

This is not an argument for less governance. It is an argument for governance that looks at the thing being governed.

Demonstration is harder to manipulate than abstraction

The attraction of velocity is also its weakness: it compresses varied work into a number before leadership has to understand what the work produced.

A sprint review reverses that comfort. It makes the team show what is complete, exposes awkward interfaces and invites the business to say whether the result is useful. A release forecast forces discussion of remaining scope, risk and dependency. Defect trends reveal

whether apparent speed is creating a liability.

These forms of evidence are less tidy than a velocity graph. They are also closer to the economic question.

In 2012, as iterative methods move from individual teams into larger programmes, this distinction matters. The ceremonies may be new to many organisations, but the impulse to convert delivery into reassuring traffic lights is not. If we are careless, the old status culture will survive intact beneath agile terminology.

Sprint velocity should remain a conversation inside the team and one input to programme judgement. It must not become the judgement itself.

The important question is not whether the number rose. It is whether the organisation learned sooner, completed something valuable and made the next decision with better evidence.