

The State of Sprints 2026

A Research Report on Retrospectives, Traceability, and the Widening Gap Between Speed and Context

Published by letRetro | 2026

Meta description: *letRetro's State of Sprints 2026 report surveys 500+ professionals across 50+ organizations worldwide to examine why sprint retrospectives are losing traceability as delivery speed increases — and how engineering, IT, product, scrum leadership, and executive stakeholders each experience the retrospective differently.*

Primary keywords: *state of sprints 2026, sprint retrospective report, agile retrospective survey, retrospective statistics 2026, scrum retrospective trends, engineering team retrospectives, agile team alignment report*

Table of Contents

1. Executive Summary
 2. Methodology and Respondent Profile
 3. The Central Finding: Velocity Is Outpacing Memory
 4. Section One — Engineering Teams
 5. Section Two — IT and Platform Teams
 6. Section Three — Solution and Product Teams
 7. Section Four — Scrum Masters and Agile Coaches
 8. Section Five — Stakeholders and Leadership
 9. Cross-Cutting Theme: Company Size and Retrospective Maturity
 10. Cross-Cutting Theme: Regional Observations
 11. The Unification Problem
 12. Recommendations
 13. Looking Ahead to 2027
 14. Closing Statement
 15. About This Report
-

1. Executive Summary

The sprint retrospective has been a fixture of agile practice for close to two decades, and by most accounts, it remains one of the most widely adopted ceremonies in modern software organizations. Yet the data gathered for this report tells a more complicated story than simple adoption rates would suggest. Retrospectives are everywhere. Trust in what they produce is not.

This report is built on responses from more than 500 individual employees across more than 50 organizations worldwide, spanning industry leaders with mature engineering functions down to early-stage startups still defining their first sprint cadence. It incorporates input from more than 200 IT and cross-functional teams and more than 100 leadership and board-level respondents, giving this study a rare vantage point: it captures not just how the people running the sprint experience the retrospective, but how the people funding and governing the organization experience its output.

The headline finding is this: as delivery speed increases across nearly every organization surveyed, the traceability of decisions, learnings, and context is decreasing at a comparable rate. Teams are not skipping retrospectives outright. What they are doing, almost universally, is compressing them, deprioritizing them relative to delivery work, and — most significantly — failing to connect what is discussed in a retrospective to what actually changes in the next sprint. The ceremony survives. Its function is quietly eroding.

A second, equally important finding is that the retrospective is not experienced as a single, shared ritual across an organization. Engineering teams, IT and platform teams, solution and product teams, scrum masters and agile coaches, and executive stakeholders each described a meaningfully different version of what a "retrospective" is for. These are not minor differences in emphasis. They reflect fundamentally different mental models of what the ceremony exists to produce — a fact that helps explain why so many organizations struggle to standardize the practice even when there is broad agreement that standardization would help.

The report closes with what we are calling the unification problem: nearly every respondent group, independent of role or seniority, expressed a desire for a single, shared source of truth for sprint context. And nearly every respondent group expressed real skepticism that their organization could achieve it. That gap — between the near-universal desire for unification and the near-universal doubt that it is achievable — is, in many ways, the true subject of this report.

2. Methodology and Respondent Profile

The State of Sprints 2026 report draws on primary survey data collected over several months from a deliberately broad cross-section of the software industry. The intent behind the sampling strategy

was to avoid the common bias in agile research toward large enterprises or, conversely, toward early-stage startups. Instead, this report sought a genuine spread across company stage, team function, and geography, so that findings would reflect the industry broadly rather than any single segment of it.

Respondent composition:

- **500+ individual employees**, drawn from engineering, product management, design, quality assurance, DevOps, and operations functions
- **50+ organizations**, ranging from publicly traded, multi-thousand-employee enterprises to pre-seed and seed-stage startups with fewer than fifteen employees
- **200+ distinct IT and cross-functional teams**, including dedicated platform, DevOps, infrastructure, and Scrum-of-Scrums teams
- **100+ leadership and board-level respondents**, including VPs of Engineering, CTOs, Heads of Product, founders, and independent board members with operational oversight of engineering organizations
- **Global distribution**, with meaningful representation from North America, Western Europe, and Asia-Pacific technology hubs

Respondents were surveyed using a combination of structured, quantitative questions (frequency of retrospectives, tools used, confidence ratings) and open-ended qualitative prompts (what gets lost between sprints, what a "good" retrospective looks like, what would need to change for retrospectives to feel more valuable). This mixed-methods approach allows the report to surface both broad, comparable trends and the texture of how different roles actually talk about the problem in their own words.

It is worth noting, in the interest of transparency, that self-reported survey data of this kind reflects perception as much as objective process — respondents were describing how retrospectives *feel* to them, not an independently audited account of what happens in every room. That said, the consistency of certain themes across otherwise very different organizations, team sizes, and geographies is itself a meaningful signal, and forms the backbone of the analysis that follows.

3. The Central Finding: Velocity Is Outpacing Memory

If this report could be reduced to a single sentence, it would be this: organizations are optimizing for velocity at the direct expense of institutional memory.

Across nearly every organization surveyed, respondents reported that their teams are shipping faster than they were twelve to eighteen months prior. This acceleration is attributed to a combination of factors: leaner teams operating with tighter headcount, the mainstream adoption of

AI-assisted coding tools that compress implementation time, and increasing competitive pressure to reduce time-to-market. Whatever the specific driver, the effect described by respondents was strikingly consistent — sprint cycles feel shorter, decisions get made and reversed faster, and there is less institutional patience for ceremonies that do not visibly accelerate delivery.

The retrospective, positioned as a reflective pause in a cadence that otherwise rewards forward motion, has absorbed much of this pressure. It has not disappeared from the calendar — the overwhelming majority of respondents confirmed their team still holds some form of end-of-sprint retrospective. But its substance has visibly thinned. Retrospectives that once ran for a full hour with structured facilitation are, in many organizations, now compressed into fifteen or twenty minutes wedged between other meetings. Written retrospective notes, when they exist at all, are frequently abandoned after the meeting ends, with no clear owner responsible for translating discussion into action.

The recurring complaint across open-ended responses, phrased in dozens of different ways, converged on a single idea: teams are moving too fast to remember what actually happened, and even when they remember, they rarely have time to act on it before the next sprint has already reshaped priorities. This is the irony at the center of the report. Teams are, if anything, having more retrospective-adjacent conversations than ever — scattered across Slack threads, ad hoc debriefs, postmortems, and stand-up asides — but fewer of these conversations happen inside a structured, retrievable space. The cumulative time spent reflecting on work may not have decreased. What has decreased is the traceability of that reflection: the ability to return to a decision three sprints later and reconstruct why it was made, what alternatives were considered, and what was learned.

Speed, in other words, did not eliminate the retrospective. It eliminated the paper trail.

4. Section One — Engineering Teams: A Tax on Delivery Time

Of every group surveyed, engineering respondents were the most likely to describe the retrospective in explicitly transactional terms — as a cost weighed against the value of shipped work. For a function measured primarily on velocity, throughput, and the state of the backlog, time spent discussing *how* a sprint went can feel structurally disconnected from what determines whether the next sprint succeeds.

This did not mean engineers rejected the value of reflection outright. Rather, engineering respondents consistently asked for a narrower, more concrete version of the retrospective — one anchored to specific technical outcomes (a deployment that slipped, a defect that escaped code review, a dependency that blocked a release) rather than open-ended discussion of team dynamics

or process sentiment. Several respondents described a preference for retrospectives that functioned more like lightweight root-cause reviews than group therapy sessions, with clear artifacts — a ticket, a follow-up PR, a documented decision — as the expected output.

A related and frequently mentioned frustration was the disconnect between the retrospective and the tools engineers actually work in daily. Retrospective notes typically live in a separate document or board from the codebase, the ticketing system, and the CI/CD pipeline that generated the actual signal being discussed. This forces engineers to manually reconstruct context that, in principle, already exists somewhere in their tooling — commit history, pull request discussions, incident logs — but is not surfaced automatically into the retrospective conversation. Several engineering respondents specifically noted that AI-assisted development has made this gap more visible: as code changes accelerate, the manual overhead of translating engineering activity into retrospective-ready context has become proportionally larger, not smaller.

5. Section Two — IT and Platform Teams: Incident Memory, Not Sprint Memory

IT and platform teams surveyed described a meaningfully different relationship to the concept of a "retrospective" than product-facing engineering teams. For many of these teams, sprint velocity is a secondary concern; operational continuity is the primary one. As a result, their most trusted form of retrospective is not the standard two-week sprint ceremony but the incident postmortem — a structured review triggered by an outage, a security event, or a significant operational failure.

This distinction matters because it reveals a structural blind spot in how many organizations think about retrospectives. Sprint retros and incident retros, in the overwhelming majority of organizations surveyed, live in entirely separate systems, run by different facilitators, and are rarely cross-referenced. A platform team might produce an exceptionally detailed postmortem after an outage, capturing root cause, timeline, and remediation steps — but that document typically has no connection to the sprint planning process that determines whether the remediation work actually gets prioritized. Respondents described this as one of the most persistent sources of frustration: high-quality retrospective content exists, but it is siloed away from the planning process it should be informing.

IT and platform respondents also raised a distinct concern around on-call handoffs and knowledge transfer. As team composition changes — new hires, contractor rotations, or reorganizations — context that was captured in a retrospective or postmortem months earlier often fails to reach the people who now need it most. Several respondents described relying on informal channels — a senior engineer's memory, a pinned Slack message — to fill gaps that a properly connected retrospective system should, in principle, have already closed.

6. Section Three — Solution and Product Teams: Retrospective as Roadmap Input

Solution architects and product-facing teams offered a third distinct lens on the retrospective. For this group, the ceremony's core value is not primarily about process improvement or technical root-cause analysis — it is about ensuring that lessons learned during execution actually shape what gets built next. In other words, product and solution teams are most likely to treat the retrospective as an input to planning, not simply a reflection on the sprint that just ended.

This framing produced a distinct set of frustrations. Rather than complaining about the retrospective conversation itself, solution and product respondents most frequently pointed to what happens — or fails to happen — afterward. Action items are raised, discussed, and generally agreed upon in the room, but a substantial share of respondents reported that these items are rarely tracked with the same rigor as feature work, and often quietly disappear from the next sprint's planning session. The retrospective, in this framing, is not broken because the conversation is unproductive. It is broken because the organization has no reliable mechanism for converting a retrospective insight into a prioritized, resourced piece of backlog work.

Several respondents in this group specifically pointed to a lack of ownership as the root issue: action items generated in a retrospective frequently have no single accountable owner, and without an owner, they default to nobody's responsibility. This is a process and tooling gap as much as a discipline gap — most retrospective formats surveyed do not have a built-in mechanism for assigning, tracking, and resurfacing action items in subsequent sprints.

7. Section Four — Scrum Masters and Agile Coaches: Guardians of a Ritual Losing Buy-In

Scrum masters and agile coaches, unsurprisingly, emerged as the group most invested in the retrospective as a formal discipline — and, at the same time, the group most acutely aware that organizational buy-in for that discipline is eroding. Many described spending a disproportionate share of their time simply defending the retrospective's place on the team calendar, rather than facilitating the substance of the conversation itself.

This group's responses carried a distinct undertone of institutional fatigue. Several respondents described a pattern in which leadership publicly endorses the value of retrospectives in principle,

while simultaneously deprioritizing them in practice — shortening their duration, allowing them to be skipped during "crunch" periods, or treating them as optional once a team feels sufficiently senior or self-managing. Agile coaches in particular expressed concern that as AI tooling accelerates development cycles and encourages more asynchronous, distributed work patterns, the traditional whole-team, synchronous retrospective format is becoming structurally harder to sustain — even though, in the view of most coaches surveyed, the underlying need for structured reflection has not diminished at all, and may in fact be increasing.

A notable minority of scrum masters and agile coaches described actively experimenting with new formats in response — asynchronous written retrospectives, rolling retrospective boards updated continuously rather than in a single sprint-end session, and lighter-weight formats designed to fit into fifteen-minute windows. The consistent theme across these experiments was a search for something that preserves the reflective function of the retrospective without requiring the same block of synchronous meeting time that made the traditional format increasingly difficult to defend.

8. Section Five — Stakeholders and Leadership: Visibility Without Detail

Leadership and board-level respondents offered perhaps the most structurally distinct perspective of any group in this report. Almost without exception, this group expressed interest in the *outcomes* of retrospectives — recurring blockers, team health trends, delivery risk signals — without any expectation of engaging with the retrospective process itself. Their consistent request was for a rollup: a clear, aggregated view of what changed team-over-team and sprint-over-sprint, without needing to read raw notes from dozens of individual sessions.

This request exposes a capability gap present in the overwhelming majority of organizations surveyed. Retrospective data, where it exists in any durable form, is almost always scattered — spread across different tools, formats, and levels of detail depending on which team produced it. There is, in most of the organizations surveyed, no reliable mechanism for aggregating this data into the kind of organization-wide trend view that leadership respondents said they wanted. As a result, many leadership respondents reported relying on informal signals — anecdotes relayed by a VP of Engineering, sentiment picked up in skip-level conversations — rather than any systematic view of retrospective data, despite retrospectives nominally happening across every team in their organization.

Several leadership respondents also connected this gap directly to risk management. Recurring technical or process issues that surface repeatedly in individual team retrospectives often fail to reach leadership until they have already caused a visible, costly failure — precisely because there is no aggregation layer surfacing the pattern earlier. In this framing, the absence of a unified

retrospective system is not simply an efficiency problem; it is a blind spot in organizational risk visibility.

9. Cross-Cutting Theme: Company Size and Retrospective Maturity

Retrospective maturity did not track cleanly with company size, but it did track with a specific inflection point that recurred across the data: the transition from a single, informal team to multiple, semi-autonomous teams. Early-stage startups surveyed, even with very small headcounts, generally reported high satisfaction with their retrospective process, largely because a single team, a single tool, and a single shared context made informal reflection sufficient. As organizations grew past roughly three to five engineering teams, satisfaction and confidence in the retrospective process dropped sharply — precisely the point at which context needs to travel between teams that no longer share a single room, a single stand-up, or a single informal Slack channel.

Larger, more established organizations surveyed had generally invested in more formal retrospective tooling and facilitation training than early-stage startups, but this investment did not straightforwardly translate into higher confidence. Several large-organization respondents described a paradox in which more formal process produced more retrospective *documentation*, without necessarily producing more retrospective *impact* — a large volume of recorded notes that nonetheless failed to meaningfully change subsequent sprint behavior.

10. Cross-Cutting Theme: Regional Observations

While the core themes of this report held broadly consistent across regions, some directional differences emerged. Respondents in North America were somewhat more likely to describe retrospectives as compressed or deprioritized under delivery pressure. Respondents in Western Europe more frequently emphasized team psychological safety and facilitation quality as the primary lever for retrospective value, independent of speed pressure. Respondents in Asia-Pacific technology hubs, particularly at fast-growing startups, were the most likely to describe experimenting with AI-assisted or asynchronous retrospective formats, consistent with generally higher reported adoption of AI development tooling in that region during the survey period.

These regional differences are directional rather than definitive, given sample sizes within each region, but they suggest that the retrospective's central tension — speed versus traceability — plays out somewhat differently depending on local team culture and tooling maturity, even as the underlying tension itself appears to be a global phenomenon.

11. The Unification Problem

Every group surveyed for this report — engineering, IT and platform teams, solution and product teams, scrum masters and agile coaches, and executive stakeholders — converged, independently, on the same word when asked what they wanted most from their retrospective process: unification. A single, shared source of truth for sprint context, decisions, and follow-through, rather than the fragmented patchwork most organizations currently operate with.

And yet, when asked how confident they were that their organization could achieve this unification within the next twelve months, confidence dropped sharply and consistently across every respondent group. Scrum masters and agile coaches expressed the highest relative optimism, though even within this group, confidence was far from universal. Leadership respondents were the most skeptical of all — a number specifically cited prior, unsuccessful attempts to standardize retrospective tooling or process across their organization, typically abandoned once the internal champion driving the initiative moved to a different role or was pulled onto a higher-priority project.

This is the tension that sits underneath the entirety of this report's findings: near-universal agreement on the destination, combined with near-universal doubt about the path to reach it. Respondents most frequently cited three specific blockers to unification: fragmentation across existing tools (with retrospective-relevant context spread across chat platforms, ticketing systems, documents, and human memory), inconsistency in how different teams facilitate and document their retrospectives, and the sheer organizational coordination effort required to get five structurally different functions — engineering, IT, product, Scrum leadership, and executive stakeholders — to agree on a single shared process.

12. Recommendations

Based on the patterns surfaced across this report, several recommendations emerge for organizations seeking to close the gap between retrospective aspiration and retrospective reality.

Anchor retrospective context to existing engineering activity. Rather than treating the retrospective as a standalone conversation disconnected from the codebase and ticketing system, organizations should look for ways to surface commit history, pull request activity, and incident data automatically into the retrospective, reducing the manual reconstruction burden that engineering respondents identified as a core frustration.

Assign explicit ownership to every retrospective action item. The single most common reason cited for action items disappearing after a retrospective was the absence of a named, accountable owner. A retrospective without ownership assignment functions as a conversation, not a process improvement mechanism.

Bridge sprint retrospectives and incident postmortems. IT and platform teams identified a clear structural gap between operational retrospectives and sprint-level planning. Organizations should look for ways to connect these two forms of reflection so that operational lessons are visible during sprint prioritization, not siloed in a separate system.

Build an aggregation layer for leadership visibility. Leadership respondents consistently wanted trend-level visibility into retrospective outcomes without needing to read every team's raw notes. Organizations should invest in some form of rollup — whether a lightweight report, dashboard, or periodic summary — that surfaces recurring patterns across teams before they escalate into visible failures.

Treat retrospective format as a function of team context, not a fixed template. Given how differently engineering, IT, product, and leadership respondents described their ideal version of a retrospective, a single fixed format is unlikely to serve every function well. Organizations may benefit from allowing format to vary by team while ensuring the underlying context remains connected and retrievable across the organization.

13. Looking Ahead to 2027

Several trends observed in this year's data are likely to intensify rather than resolve on their own. As AI-assisted development continues to compress implementation timelines, the gap between how fast decisions are made and how slowly organizations currently capture the context behind those decisions is likely to widen further, absent deliberate intervention. Organizations that treat this as purely a cultural or discipline problem — asking teams to simply "take retrospectives more seriously" — are likely to see limited improvement, based on this year's findings. The data suggests the more durable path forward involves connecting retrospective context directly to the tools and activity generating it, and building the aggregation and ownership mechanisms this report identifies as consistently missing across organizations of every size and stage.

14. Closing Statement

The retrospective is not dying. If anything, the data gathered for this report suggests it is needed more urgently than at any point in recent memory — teams are moving too fast to trust their own

recollection of what happened even a few sprints ago. What is breaking down is a narrower assumption: that a single, generic ceremony, run the same way for every function, can adequately serve engineering, IT, product, Scrum leadership, and executive stakeholders all at once. Speed has exposed that assumption as no longer sufficient. Based on the patterns identified in this report, the organizations most likely to pull ahead through 2026 and into 2027 will be those that find a way to preserve context and traceability without treating it as a trade-off against velocity — rather than those that simply try to run their existing retrospective process faster.

15. About This Report

The State of Sprints 2026 is published by letRetro, an AI-powered retrospective platform designed to keep sprint context, decisions, and follow-through connected across engineering, product, IT, and cross-functional teams. This report is based on primary survey data collected from more than 500 individual respondents across more than 50 organizations worldwide, including more than 200 IT and cross-functional teams and more than 100 leadership and board-level participants. For more information, visit letRetro.com.