PLAYBOOK · MAINTAINED REFERENCE

The Standup Playbook

Ed Schaefer · Version 1.0 · August 2026 · edward-schaefer.com/playbooks/standup · Free to share

Most standups have become status meetings. Everyone takes a turn, says what they did yesterday and what they're doing today, and the meeting ends with the same work sitting in the same place. It's all about the people and not the work. This document works two ways: read it start to finish and run tomorrow's standup differently, or keep it open as a live reference once the shape is familiar. Point an agent at it and it can do either job.

1. The 60-second version

- **Who's in the room:** the team doing the work, plus any subject-matter experts holding information the work needs.
- **Duration:** target fifteen minutes; I almost always block thirty.
- **The core move:** walk the board right to left, work closest to done first. On a board that runs to-do, in progress, review or QA, then UAT (user acceptance testing, where finished work waits for someone to confirm it does what was asked), you start at UAT and move left. Column names differ by team; the direction is what matters.
- **The question that does the work:** for each item, "what would stop this from being done by tomorrow."
- **Facilitation:** most teams benefit from having a facilitator; the engineers get to focus on the work.
- **Where it sits:** planning populates the board this walks; review closes the cycle; retro sits between review and planning. Standup is the daily sync inside the cycle.
- **The default failure:** a per-person round of the three questions, "so it's all about the people and not the work."

2. The default play

Check in with everyone. A short check-in before the work starts.

Open floor. "Anything pressing or urgent or top of mind" gets said before the walk begins, so nobody sits on something urgent waiting for their turn.

Walk the board, right to left. Start at the column closest to done and move left: UAT, then QA and review, then in progress, then to-do. The work that's closest to being done is the highest and most important focus.

For each item, the questions are about the work: what happened with it since we met yesterday, what the plan for it is by the time we meet tomorrow, and whether anything is blocking or impeding it. Then the one I lean on hardest — what would stop this from being done by tomorrow. Section 4 has all of these as sayable lines.

The flow-metrics literature arrives at the same question from a different direction: ActionableAgile's guidance reframes the standup question as "what's not moving?" rather than "what are you working on?" (Section 6 has the metric underneath that, and when to reach for it.)

Move-questions at each column boundary. "Can we move from UAT to done? Can we move from QA to UAT?" Each one is an explicit ask about moving something forward.

Blocked items. Check them by name, every time.

Stalled items. For anything that hasn't moved: "who can take ownership of it, or what needs to happen next."

Overflow, in the room. When a thread starts running long: "can you connect about this right now or right after, and if not, when can you connect; can we get that scheduled right now." The follow-up gets booked before the meeting ends.

The 16th minute. Mike Cohn's convention of ending at fifteen minutes and saving deeper discussion for "the sixteenth minute" is mostly a crutch for teams holding the fifteen too tightly. I'd rather have that time available in the first place (Section 8).

Facilitation. Some teams genuinely run their own standup, and rotation works if a team prefers it. My default: most teams benefit from a facilitator trained for it, because it lets the engineers focus on the work instead of on running the meeting.

Where the sources disagree

Problem-solving inside the standup is officially discouraged by Scrum guidance, and empirically it's the most valued part. Stray, Moe, and Sjøberg observed over a hundred stand-ups and found teams spent nearly a third of the time elaborating problems and discussing solutions, then called that problem-focused communication necessary for effective decision-making. My own read is the same as the evidence: it can be good or bad, depending. A thread that pulls in the right people and resolves something the work needed is the meeting doing its job. A thread that recaps one person's day in detail for ten minutes is not.

3. The anti-pattern: the three questions

The round where each person answers "what did you do yesterday, what are you doing today, any blockers" is the single most common failure shape I've seen, and I treat it as the anti-pattern, not as a coequal option. It turns the meeting into "I was working on this yesterday, I'll work on it more today, and no blockers," repeated per person, all about the people and not the work.

This isn't only my read. The Scrum Guide carried the three questions for years, then downgraded them to an optional example in 2017 and dropped them entirely in 2020. What the current text says instead: the developers "can select whatever structure and techniques they want, as long as their Daily Scrum focuses on progress toward the Sprint Goal and produces an actionable plan for the next day of work." The board-walk sits squarely inside that. Two honest limits on how much that proves: a permissive clause is permission, not endorsement, and there is no head-to-head study comparing the board-walk against the three questions — Jason Yip, the clearest published voice for walking the board, says so himself. What follows is my practice and my reasoning.

If your team insists on keeping the questions, keep the goal-tied version: what did I do yesterday in service of the sprint goal, what am I doing today in service of it, and what blockers impede *the team's* success rather than only my own. That version is close to what the Guide itself required before it cut them.

The person-level failure has a documented mechanism. Stray and colleagues found that when the facilitator's behavior made team members address the facilitator, the stand-up became a status-reporting meeting; one of their teams reported their best daily meeting on a day the scrum master was absent, which says something about where people were pointing their attention rather than about the person. Jason Yip names the same pattern as Reporting to the Leader, with structural fixes: rotate the facilitator, break the eye contact. The mechanism is who the room is talking to. My preference: don't go by person at all.

4. Script: running the standup

Check-in.

> "Quick round: how's everyone doing today?"

Open floor.

> "Before we walk the board: anything pressing, urgent, or top of mind?"

The walk, per item.

> "What's happened with this since yesterday, and what's the plan for it by tomorrow?"
>
> "What would stop this from being done by tomorrow?"

Move-questions.

> "Can we move this from UAT to done? What about QA to UAT?"

Blocked.

> "This one's flagged. What needs to happen to unblock it, and who's taking that?"

Stalled.

> "This hasn't moved in a few days. Who can take ownership of it, or what needs to happen next?"

Overflow.

> "That's worth more time than we have here. Can you two connect right after this, or should we get something scheduled right now?"

Announcing the change to the team. Deliver it live, in the standup itself, and keep it short:

> "I want to change how we run this. Instead of going around the room with what we each did yesterday, we're going to walk the board — start with what's closest to done and work backwards, and ask what would stop each thing from finishing by tomorrow. Same time, same slot. The reason is that I care more about whether the work is moving than about what each of us was busy with. If it doesn't work after a couple of weeks, we'll change it again."

Expect the first few to feel awkward, and expect at least one person to give a per-card status report out of habit. That fades.

If you're not the one running it. A standup that's gone to status theater doesn't fix itself. Asking for better is a script too, built from the failure modes in Section 8:

> "I think our standup could work harder for us. Right now it feels like we each report our own status and then we're done. Could we try walking the board instead — start at the right, look at what's closest to done, and ask what's stopping each item from finishing? Same fifteen minutes, and it keeps us on the work."

If that goes nowhere, you still have a move: when your own turn comes, answer about the work rather than about yourself — name the item, say what would stop it finishing, and ask who can help. One person doing that consistently changes the shape of the meeting more often than a proposal does.

5. What belongs, what doesn't

Segment	Belongs	Doesn't belong
Check-in	A short check-in before the work starts	A long round of personal updates that eats the meeting
Open floor	Anything pressing or urgent, said up front	Sitting on something urgent until your turn comes around
The walk	Right to left, closest to done first, per-item questions about the work	Going person by person through yesterday-today-blockers
Blocked and stalled items	Named every time, with an owner or a next step	Flags that sit on the board unremarked day after day
Overflow	Connected right now, right after, or scheduled in the room	"Let's take that offline" with no booking attached
Facilitation	Most teams benefit from a facilitator; rotation is acceptable	Status reporting addressed to whoever's running the meeting

6. How many cards, and how old

You can't walk forty cards a day, and if the board has forty items on it, that itself points toward a bigger problem — too much work in progress, or too big a team. There are exceptions: sometimes you can walk all of them, when the items are closely related or when most of them plainly aren't moving right now.

The fix inside the meeting is age. Prioritize by going oldest items first; items younger than a certain age can get looked at less often. The threshold comes from the team's own data: check your current cycle time — how long items actually take from start to finish — and use that as the starting point for what counts as concerning age. For a team with no data yet, two or three days is enough for an item to get flagged. Work-item age is the leading indicator here, the early warning that something is languishing, where cycle time and throughput — how long finished items took, and how many finish per week — only tell you about work that already completed.

7. When it works

The tell I trust most: "even if the work didn't actually move across the board, the team was moving, and you could tell the right things were being done to keep progress happening."

Up close: everyone came in prepared and thinking about the work; when something was blocked, someone else offered to help unprompted; people connected right after without anyone scheduling it. The feeling in the room is "this is the one time a day we all sync up and get aligned, and then we actually do it."

8. Failure modes and fixes

The cargo cult. The root failure under most of the others: doing it a certain way because that's how it's done, instead of the way that works for your team and the work. The timebox, the standing, the three questions are all means that get treated as the point. Fix: hold the purpose — inspect progress toward the goal, adapt the plan for the next day — and let the mechanics serve it.

The militant 15. Enforcing exactly fifteen minutes as if the number is the virtue, cutting off real coordination to protect the timebox — less valuable than letting it run naturally, and the "sixteenth minute" exists as a patch for exactly this. Fix: target fifteen, block thirty.

The rambling hour. The opposite pole, and just as common: completely laissez-faire, takes thirty minutes or an hour and is just all over the place. Stefan Wolpers names this one "monologs"; Geekbot describes the same pattern as monologues. Fix: walk the board (Section 2), which bounds the discussion by the work, and enforce the overflow rule.

Status theater. The dominant failure shape, and Section 3 covers it in full: a per-person round where everyone performs progress for whoever's running the meeting. Fix: don't go by person at all; walk the work.

Forcing daily when the work doesn't warrant it. "Forcing every day is an anti-pattern." Every-other-day, twice a week, virtual, or weekly are all legitimate when they fit the work. Stray's team found the same thing empirically and recommend it plainly: in a well-communicating team, stop meeting daily if three or four times a week is sufficient. Fix: match the cadence to the work, and make any change a deliberate team decision. Wolpers names the drift version — skipping without a routine — as its own slippery slope; both are true at once.

The walk turns into status theater anyway. The board-walk is not immune to the failure it fixes. If people address the facilitator instead of each other, you have moved the status report from people to cards and changed nothing. The same is true when the walk becomes a round of "mine's fine" per card. Fix: watch who the room is talking to, and if the answer is "whoever's running it," rotate the facilitator or hand the walk to the team.

The board is stale, so the walk is fiction. The walk only works if the board reflects reality. A team that updates cards right before standup, or not at all, will have a tidy meeting about work that isn't where it says it is. Fix: treat a stale board as the finding, and fix that before blaming the meeting.

The work isn't on the board, or the team doesn't share work. A very small team, or a team of people working on genuinely unrelated items, gets less from a walk than from talking to each other — there is no shared flow to inspect. Fix: for unshared work, the per-person round is honestly the better shape; use it, and keep the goal-tied version from Section 3. The walk earns its place when the team's work is actually connected.

The 16th minute as a crutch. Reserving the real coordination for after the meeting, every day, as a standing practice — the deferral becomes the routine. Fix: the overflow rule in Section 2 — connect right now, right after, or scheduled in the room — instead of a daily pattern of deferral.

9. Variations

Duration and cadence. The Scrum Guide sets fifteen minutes flat for the Daily Scrum, not "at most" fifteen, which is a hedge from earlier revisions. My target is fifteen and almost always block thirty, because if it's the one time we all meet each day, I'm not worried about it running past fifteen. The meeting-science literature on huddles typically lands on ten to fifteen minutes; my thirty is my own calibration for remote teams, not a research finding. Cadence: match it to the work (Section 8), and change it deliberately when the work changes.

Remote and virtual. Virtual formats and skip days are fine. This used to run on a whiteboard with stickies; now it's whatever board the team lives in, screen-shared. Some teams run a sixty-to-ninety-minute morning block: coffee and catch-up, the standup, then overflow and technical problem-solving in one slot. Some run an afternoon stand-down instead — the same sync at the end of the day, written up as a named format by Atomic Object's Dylan Goings.

Async status. The sources genuinely disagree here. The async-tool world treats written check-ins as a clear win for distributed teams; Honeycomb's Ben Darfler killed them outright, reporting that no one reads async stand-ups. Stray's team lands on the hybrid that matches how I think about it: async for the status layer, synchronous for the discussion. Written updates carry what happened; they don't carry the moment someone says "wait, that blocks me."

10. AI-era

With humans on the team, the standup still earns its slot — daily or semi-daily.

Mixed teams of humans and agents. Agents could participate directly, or the humans represent their agents, almost more like a scrum of scrums (the pattern where one representative per team carries that team's status into a shared meeting), each person speaking for the work their agents are doing.

A solo builder with a fleet gets a daily brief. I run this myself. What the brief carries: what work was done yesterday, what work is still in progress, whether anything stalled or failed, problems found, pipeline failures, what's planned for today, which agents are working on what, and what to expect by end of day. For someone like me, a solo builder with an agent fleet, that's still extremely useful.

A lot of people wouldn't bother, and would just ask on demand in real time. The brief earns its keep more as the tooling for autonomous work grows.

11. This is a fit question, not a recipe

The shape above is a strong default. Where a specific team actually sits against it, and what that's costing, is what the Engineering Ways of Working Diagnostic is built to find.