The Review Playbook

Ed Schaefer · Version 1.1 · August 2026 · edward-schaefer.com/playbooks/review · Free to share

Most teams that think they're running a sprint review are running a demo. The two get combined, and the demo is what survives; the review itself never happens. This document works two ways: read it start to finish and run a real review this cycle, 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

- **Cadence:** fixed. Weekly, every other week, or monthly, depending on the team, the nature of the work, and how fast things change. Every two weeks is the default when nothing else dictates otherwise.
- **Duration:** 60 to 90 minutes is typical. It can run as short as 30, and as long as it needs to, but not past two hours. (The Scrum Guide's own cap is four hours for a one-month sprint, proportionally shorter for shorter ones — about two hours at a two-week cadence.)
- **Review closes the cycle** and preps the planning that follows it.
- **Demo is last in the cycle, and it's public.** Stakeholders, sponsors, whoever gets invited.
- **Retro sits between review and planning:** after the review, before the next planning session, team-only.
- **Stakeholders belong at review and demo.** Retro doesn't include them.
- **The first-demo rule:** the scheduled demo is never a work item's first demo. Demoable work gets shown as soon as it's ready, in standup or on a quick call, so feedback arrives early instead of waiting for the calendar.
- **A review is never just a demo.** If the meeting was only a demo, the review hasn't happened yet.

2. The default play

Run this as a work-board walkthrough — Jira, ADO, whatever the team already tracks work in — going through it item by item.

Completed work. What got finished, and what the team learned finishing it.

Started but not finished. What's still in flight, and why it didn't close this cycle.

Not started. What never got picked up, and why.

Organizational blockers. Any blocker or impediment at the organizational level that kept work from getting done, not just team-level friction.

Re-check value. For anything unfinished or not started: is this still valuable? Does it still need to happen next cycle, or has something changed?

Big-picture impact. How what did and didn't get done changes what's most important to work on next, and what the plan should be going into the next cycle.

Industry and organizational changes. Anything that's shifted in the market or the organization's own direction since the last review, and what that does to the plan going forward.

Who runs it: the scrum master or team coach, the product owner, or some combination of the two.

Who talks: the team does most of the talking about what got done and what problems or blockers got in the way. The product owner covers market, direction, and vision. Stakeholders ask questions and bring feedback or market and business news of their own.

Prepare going in, for the product owner:

- Market research and findings.
- Whether the roadmap, plan, or direction is still the same or has changed.
- The calls the team will need: for anything unfinished, keep working it, shelve it, or change what "done" means so it can finish now and get reprioritized in a few weeks.

The one question the product owner answers at every review, even a quiet one: what do we do with the unfinished work — the items not done, not started, or still in flight.

3. Script: converting a demo-only "review"

Start wherever the team already is. The conversion doesn't need a new meeting: bolt the review half onto whatever's already scheduled, the demo or standup, or run it separately if there's room for a new slot. Say something close to this, to whoever currently owns the demo-only meeting:

> "We've been running this as a demo, and it's a good demo. I want to add the other half back in: what we finished, what we didn't and why, anything that blocked us at the org level, whether the stuff we didn't finish is still worth doing, and what all of that means for the plan going into next sprint. We can run it as the first half of the demo, in standup, or as its own meeting, whichever fits. I'd like to try it for a few cycles and see how it goes."

Whichever container gets chosen, keep the two halves distinct. The review comes first and closes out the cycle. The demo is public and comes last.

4. Script: running the review

Open by walking the board, rather than going around the room person by person:

> "Let's go through the board, item by item."

Per-item questions, asked as each work item comes up:

- "Is this done? If it's not, is it still in flight, or did it never get started — and why?"
- "What did we learn finishing this, or what's blocking it?"
- "Was anything that blocked this outside the team's control — something at the organizational level?"
- "Is this still worth doing? Does it still need to happen next cycle?"

Big-picture beat, once the board is covered:

> "Given what did and didn't get done, here's what I think that changes about what's most important going into next cycle."

The PO's market and direction beat, right after. What it sounds like done well:

> "We thought we'd invest more in this feature set, but users are telling us what we have meets their needs — so we'll finish it off and pivot to the features we're getting requests for."
>
> "A competitor shipped this first, and we've decided it's not worth continuing to invest. Here's where that work is going instead."
>
> "The business has identified new pressing priorities. This is still important — that is more important right now."

Stakeholder participation prompts, opened to the room:

- "What questions do you have about what you just saw?"
- "Is there anything you're hearing about the market or the business we should know?"
- "Does anything here change what you'd expect from us next?"

Close by naming what changes. Confirm out loud whether the plan going into the next cycle is the same plan or an adjusted one. That confirmation is the direct input planning needs the next day.

5. What belongs, what doesn't

Segment	Belongs	Doesn't belong
Team portion (work-board walkthrough)	Completed items and what was learned; started-not-finished items and why; not-started items and why; organizational blockers; a value re-check; the big-picture read on next cycle's plan	Status theater: percent-complete numbers with no discussion of why or what changes next
Stakeholder / PO portion	Market and direction updates; industry or organizational changes and their effect on the plan; stakeholder questions and feedback	A market and direction beat that gets skipped for time; stakeholders who never engage
Demo	A lead stakeholder or key owner working the product hands-on	The team narrating a walkthrough in the stakeholder's place

Review content that never got covered, because the meeting was only a demo, and then surfaces for the first time in planning instead — that's a clear sign the review itself isn't happening.

6. The demo

The scheduled demo is never a work item's first demo. Work that's ready gets shown as soon as it's ready, in standup or on a quick call — I want feedback as early as possible, and I don't want to wait for it. The fixed-cadence, public demo, the one on the calendar with stakeholders and sponsors and whoever else is invited, comes later, as a showcase rather than a first look.

Who drives it. A lead stakeholder or key owner, someone closer to the end user than the team is, works the product hands-on instead of watching the team walk through it. Have the work ready to go, and have that person actually drive. If they can drive it, the work is more likely to succeed, and the pain points and trouble they run into surface in the room. If it works, that's a good sign it will work for the user. LeSS's framework guidance lands nearby, preferring hands-on interaction by real users over a passive team demo (less.works), and handing the stakeholder the keyboard surfaces problems a scripted walkthrough hides (Verwijs).

Feedback routing. A bug or defect goes straight into the backlog, usually worked in the next cycle or planned ahead for a future one. Anything else, an enhancement or a new idea the demo surfaced, goes into the enhancement backlog for prioritization: worth doing, or not, decided separately.

Where the sources disagree

Not every source agrees on who should drive the demo, and it's worth stating the disagreement plainly rather than smoothing it over. Mike Cohn's guidance is relativist — no hard rule, experiment with what works. James Shore's default is a third answer: the product manager drives. Jenni Jepsen and Ole Jepsen argue a different axis entirely: the team should keep the controls, so the team builds its own skill at eliciting feedback.

All three positions are defensible. My default stands: a lead stakeholder or key owner drives, because closeness to the end user is the signal that predicts whether the thing will actually work for them once it ships.

7. Failure modes and fixes

The collapse cascade. The most common failure, and the one behind most of the others. Combine review and demo into one meeting without protecting both halves, and only the demo survives: the review content doesn't happen, so it has to happen in planning, with planning, which usually means less planning actually happens. Fix: protect the work-board walkthrough as its own segment inside a combined meeting, the same way the demo already gets one.

Slides stand in for the software. The meeting presents decks and screenshots instead of the running product. Fix: nothing gets presented that isn't running.

The meeting becomes a numbers report. Velocity and percent-complete get read out instead of a conversation about what changed and what it means. Fix: the work-board walkthrough asks why, not just what. A number alone doesn't answer any of the seven steps in the default play.

The review runs as a go/no-go gate. It functions as a sign-off checkpoint instead of a session that actually adjusts the plan. Fix: the re-check-value and big-picture steps are where adjustment happens. If nothing about the plan is allowed to change coming out of it, it isn't a review.

Skipped entirely. The team moves straight from work to planning with no read on value, no market or direction beat. Fix: start with the conversion play in Section 3, bolted onto whatever meeting already exists, rather than waiting for an ideal standalone slot that never gets scheduled.

No stakeholders show. The team runs it as internal-only, and the market and direction beat disappears along with the demo. Fix: invite deliberately, every time, and treat the market and direction beat as reason enough for stakeholders to be there.

8. Variations

No-sprint / Kanban teams. Some Kanban teams still work on a schedule; others run reviews ad hoc, as needed. My default holds regardless of framework: a fixed cadence, whether that's weekly, every other week, or once a month, with the actual frequency set by the team, the nature of the work, how fast things change, and how long work takes.

Review and demo, one meeting or two. Whether they run together or separately depends on audience and context, and one of the defaults above decides part of it for you: the demo is always public, so a team-only review forces two meetings — a public event can't contain a private one. If the review runs public, one meeting works; hold the two halves distinct inside it, review first, demo last. Either way the sequencing holds: the review closes the cycle, the demo is the cycle's last event.

Public or private review. The review portion can be public or team-only, depending on the organization. The stakeholder role in the default play is a reason to lean public: if stakeholders are expected to bring market and business news into the review, they have to be in the room for it. A team-only review still works, but the market-and-direction beat then rides entirely on the product owner.

Where the frameworks land. The job a review does is universal: inspect what happened, decide what it means for the plan. A scheduled event for doing that job isn't — Extreme Programming folds the function into its Weekly Cycle, and Shape Up replaces it with a forward-looking Betting Table. What that means in practice: a team on one of those methods doesn't need to add a meeting to use this playbook. It needs the seven default-play questions answered somewhere, on a regular rhythm. If they already have a home, the job is covered; if nobody can say where they get answered, that's the gap to close.

9. AI-era

Two things change as agents take on more of the work. First, review starts doing a job it didn't used to have: checking agents' work, not just people's. Second, the cadence itself may need to tighten, more frequent reviews and more frequent demos, because agents can move through more work in the same amount of calendar time. Resolve feedback before moving on to the next thing.

DORA's 2025 research on working in small batches backs the instinct: the practice is tied specifically to counteracting the delivery instability that shows up as AI accelerates throughput.

What doesn't belong here: ideas about AI inside the retrospective. That's a different event, with its own playbook.

10. 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.