THE PLAYBOOK

The Retro

A structured look back that produces one thing that actually changes. Why most retrospectives fail, the rule that makes them safe, and how to run one outside a software team.

Paul Prado Pacardo

Executive Assistant & Operations · me.kabuhayan.app

Contents

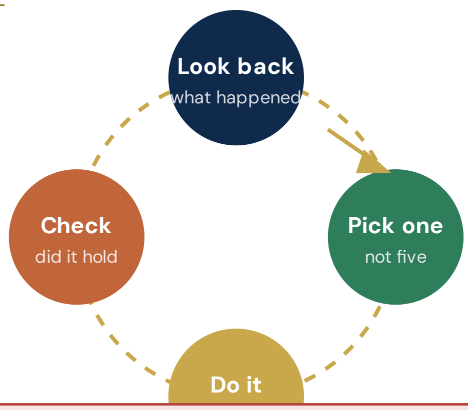
OPENING

The meeting where everyone agrees and nothing changes

A retrospective is a structured conversation about what happened, held with the people it happened to, ending in a change. The idea is sound and the execution usually is not.

The typical failure is recognisable. The team meets after a project. People say reasonable things. Somebody takes notes. Eight improvements are identified. None of them happen, and the next retrospective is attended with visible reluctance.

THE LOOP THAT HAS TO CLOSE



Most teams run the first station and skip the other three. That is why retrospectives get a reputation.

Figure: the discussion is the easy quarter. Everything that makes it worth doing is in the other three.

Most teams run the first station and skip the other three. The discussion is genuinely the easy part and it is the part everybody focuses on.

This playbook is deliberately narrow. One rule about safety, four questions in a particular order, and one action with an owner and a date. Everything else is optional and most of it is decoration.

CHAPTER 1

Why most retrospectives produce nothing

Four causes, and they compound.

Too many actions. A session that produces eight improvements produces zero, because a team that is already busy cannot absorb eight changes and will not choose between them afterwards. One is the right number and it feels insultingly small the first time.

Actions with no owner. "We should document that" belongs to everybody and therefore nobody. This is the same failure as any decision without a name attached.

Nobody checks. The action is agreed and never revisited, so the team learns that retrospective actions are aspirational. After two rounds of that, people stop proposing anything ambitious.

The honest thing goes unsaid. If the real cause of a problem is a decision somebody senior made, and that person is in the room and reacts badly, the session becomes theatre permanently. This is the failure that cannot be fixed by process.

> **A retrospective that produces eight improvements has produced a list. One that produces a single change has produced an improvement.**

CHAPTER 2

The one rule that makes it safe

The research on team effectiveness converges on psychological safety, which is the shared belief that you will not be punished for speaking up. It is well established in the academic literature and it was the strongest predictor of team effectiveness in Google's Project Aristotle work across 180 teams. Worth noting that the evidence is largely observational rather than experimental.

Whatever the strength of the causal claim, the practical version is simple: **if it is not safe to name what went wrong, the session cannot work.**

The rule I would use, said out loud at the start of every session: **we are looking at what happened, not at who did it.** And then the leader has to demonstrate it in the first session, because saying it is not what makes it true.

The moment that decides everything is the first time somebody names a decision the leader made. How that is received is the entire signal, and everyone in the room is watching for it. Acknowledge it plainly, without explaining, and the session becomes real. Explain it away and you have run one honest retrospective and will not run another.

GO FIRST

If you lead, name something you got wrong before asking anybody else to. Not performatively. One specific thing, briefly, with no justification attached. It costs nothing and it is the fastest route to a useful session.

CHAPTER 3

The four questions, and the order

The order matters more than the wording, and most templates get it wrong by starting with problems.

01 What went well that we should keep doing?

First, deliberately. It is not a warm-up. Teams routinely stop doing things that worked because nobody noticed they were working, and this is the only question that catches that.

02 What was harder than it needed to be?

Phrased about the work rather than about people, and this phrasing does real work. "Harder than it needed to be" invites process answers; "what went wrong" invites defensive ones.

03 What surprised us?

The most underused question here. Surprises point at wrong assumptions rather than at execution failures, and assumptions are cheaper to fix.

04 What is the one thing we change?

One. The discussion will produce several; the session produces one, chosen out loud with everybody present.

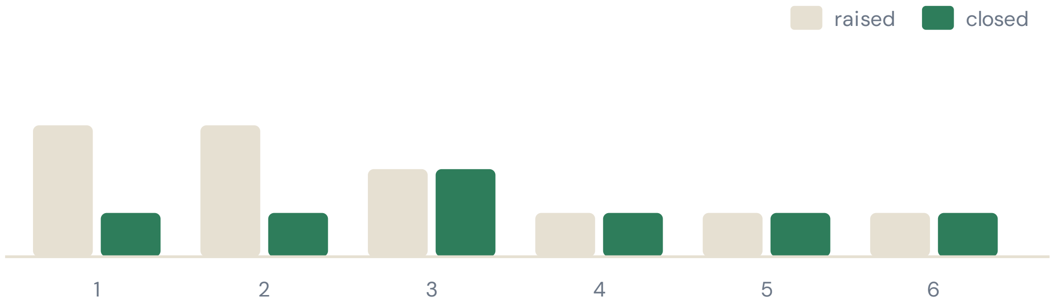
Forty-five minutes total for a team of five or six. Longer than that and the last question gets rushed, which is the one that matters.

CHAPTER 4

One action, one owner, one date

The conversion step, and the one that separates a retrospective from a discussion.

ACTIONS RAISED AGAINST ACTIONS CLOSED, BY SESSION



Sessions 1 and 2 raised three actions each and closed one. By session 4 the team raised one and closed it.

Raising fewer is the improvement, not a decline.

Figure: the number worth watching is not how many improvements were identified. It is how many actually happened.

One action. Chosen from the list, in the room, by the team. Not the most important-sounding one; the one most likely to actually happen and to make a difference.

One owner, named. A person, not a function. The owner is not necessarily the person who does all of it, but they are the person who answers for it.

One date. Specific. Before the next retrospective, ideally, so that closure is visible within the cycle.

Then open the next session with it. First item, every time: did last time's action happen? This is the check station on the loop and it is the entire reason anybody takes the process seriously.

Watch the pattern in the chart above. Early sessions raise several and close one. Later sessions raise one and close it. **Raising fewer is the improvement**, and it usually gets misread as the team losing interest.

CHAPTER 5

Running it when things went well

Retrospectives are usually triggered by something going badly, which teaches the team that the meeting is a post-mortem and produces avoidance.

Run one after something that went well. These are more useful than they sound, because success is rarely examined and its causes are consequently unknown. A team that cannot say why the last project went smoothly cannot repeat it deliberately.

Ask what nearly went wrong. The best question available after a success. There is always something that was caught late, absorbed by somebody working extra, or avoided by luck. Those are the failures of the next project, visible in advance.

And attach it to a cadence rather than to events. Monthly, or at the end of each engagement, regardless of outcome. A retrospective that only happens after trouble is a tribunal with a friendlier name, whatever anybody says at the start.

This is also what makes the difficult ones survivable. If the team has run six routine sessions, the one after a genuine failure is recognisably the same meeting rather than a special event.

CHAPTER 6

Outside software teams

Retrospectives came from software and the format transfers, with three adjustments.

Pick the unit deliberately. Software teams have sprints. An agency has engagements. A finance function has a month end. A support team has a quarter. Choose the natural boundary and use it consistently rather than trying to impose a two-week rhythm on work that does not have one.

Include the people who were actually involved, not the department. A month-end retrospective should include whoever touched month end, which may be three people across two functions. Retrospectives with an audience do not work.

Expect the first two to be shallow. Teams unfamiliar with the format will say things were fine. That is not evasion; it is a lack of practice at examining work. The third session is usually where it becomes useful, and most teams abandon it after the second.

A CAUTION ON THE EVIDENCE

The research on retrospectives outside software is thin, and the impressive percentage improvements circulating come from tooling vendors. What is defensible is the mechanism: a structured look back with a named action and a check is more likely to produce a change than an unstructured one. I would not claim more than that.

CHAPTER 7

When to stop running them

An unusual chapter for a playbook, and worth including because running them badly indefinitely is common.

Stop if actions never close. Three sessions with nothing completed means the meeting is producing a list and consuming an hour. Fix the closure problem or stop, because continuing teaches the team that improvement here is decorative.

Stop if the honest thing cannot be said. If everybody knows the real cause and nobody will name it, the session is worse than nothing: it manufactures the appearance of examination. That is a leadership problem and no format fixes it.

Pause if there is nothing to examine. Six weeks of routine work does not require a retrospective. Skipping one is fine; the cadence exists to make it normal, not to be sacred.

And stop if it has become a complaints channel with no route to action. Airing problems that nobody can address is demoralising rather than cathartic, and it is a signal that the team needs authority rather than a meeting.

CLOSING

One thing, actually done

The value of a retrospective is not in the discussion. It is in whether anything is different afterwards, and most teams optimise the part that does not matter.

Say out loud that this is about what happened rather than who did it, and if you lead, go first with something you got wrong. Ask the four questions in order, starting with what went well rather than what went badly. Produce one action, one owner, one date. Open the next session by checking whether it happened.

Run them on a cadence rather than after trouble, so that the difficult one is recognisably the same meeting as the routine ones. Expect the first two to be shallow and run the third anyway.

And watch the number of actions falling over time. That is the process working, and it is almost always mistaken for the process fading.

Paul Prado Pacardo

Senior Executive Assistant, Operations and Project Manager.
Ten years supporting C-level leadership. Available for remote work.

me.kabuhayan.app
linkedin.com/in/paulpacardo
pacardopaul18@gmail.com

Retrospective sheet

Forty-five minutes · one page · v1

SAY THIS FIRST: WE ARE LOOKING AT WHAT HAPPENED, NOT AT WHO DID IT.

Question	Notes
0. Did last time's action happen? first item, every session	
1. What went well that we should keep doing?	
2. What was harder than it needed to be?	
3. What surprised us? surprises point at assumptions	

The one thing we change	Owner	By when

One action, not five. Chosen in the room, by the team, with a named person and a date before the next session. If you lead, name something you got wrong before asking anyone else to.

Run these on a cadence rather than after trouble. A retrospective that only happens after something went badly is a tribunal with a friendlier name.