

Conducting Low-Friction Technical Retros

Feedback is critical to technical and professional growth. Retrospectives (retros) are simply a structured form of feedback aimed at helping you and your team improve how you work.

Many people view retros as something you only do when a system breaks or a project fails. While they are vital after incidents, the best retros often happen during normal operations or at the end of standard project phases. Evaluating regular work helps you catch minor friction, improve team processes, and spot opportunities for automation before problems occur.

Whether you run a retro on a specific incident, a completed project, or a two-week sprint, the goal stays the same: learn from what happened and make practical improvements.

3 Effective Ways to Run a Retro

You can use these methods on your own or run them as a group.

1. Timeline / Incident-Based

This approach works best when you are looking at a specific bug, system failure, or tight project deadline.

- **Map the Sequence:** List the key events in order. Note when the work started, when issues arose, and when key decisions were made.
- **Apply the 5 Whys:** Start with the surface issue and ask "Why?" up to five times to uncover the root cause.
 - *Example:* A dbt run failed in production → A column name changed → Schema drift occurred without warning → We lack automated schema testing → We rely on manual checks for source updates.
- **Identify 1–3 Changes:** Focus on technical adjustments, system alerts, or process fixes that prevent the issue from repeating.

2. Start, Stop, Continue

This simple framework works well for general process reviews, team habits, or project milestones. It is easy to use and translates well to everyday feedback outside of technical work.

- **Start:** What new tools, habits, or checks should we introduce to make work easier?
- **Stop:** What processes are wasting time, causing stress, or creating technical debt?
- **Continue:** What went smoothly that we should keep doing on future projects?
- **Identify 1–3 Changes:** Pick a few actionable items from your Start and Stop lists and convert them into real action items.

3. The 4 Ls (Liked, Learned, Lacked, Longed For)

This framework helps team members share balanced feedback across four distinct categories.

- **Liked:** What parts of the project or technical execution went well?
- **Learned:** What new technical insights, business rules, or tools did we discover?
- **Lacked:** What resources, documentation, or support were missing during the work?
- **Longed For:** What do we wish we had to make the work faster or higher quality?
- **Identify 1–3 Changes:** Choose items from the Lacked and Longed For categories to convert into real action items.

Running a Retro Solo vs. As a Group

Retros do not always require a full team meeting. Choosing the right setup depends on who owned the work and how many people were involved. Senior engineers lead by example, and mastering both retro styles shows true technical maturity.

Solo Retros

- **When to use:** Use this approach for individual tasks, small bugs, or pipelines you maintain alone.
- **Why it works:** It takes 10 to 15 minutes. You can process your thoughts quickly without waiting for calendar availability.

Group Retros

- **When to use:** Use this approach after major team projects, cross-functional releases, or complex system outages.
- **Why it works:** It brings multiple perspectives into one room. You see blind spots that a single engineer might miss.

Pro-Tip: Running Smooth Group Retros

When hosting a retro with a group, prepare your workspace ahead of time. Set up a shared document or virtual board with the categories already laid out, and add 2–3 of your own notes under each section first.

Send the link to your team at least two days before your meeting. Encourage everyone to add their thoughts before you meet. Gathering input early saves time during the call, allowing the team to focus on discussing root causes and picking realistic next steps.

How to Handle Common Retro Friction

Retros do not always go smoothly, especially if a project was stressful. Use these quick approaches to keep the conversation productive.

- **When someone gets defensive:** Remind the group to focus on processes and system setups rather than individual mistakes. Shift the question from "Who caused this?" to "Where did our checks fail to spot this early?"
- **When the discussion gets stuck on complaints:** Acknowledge the issue, then ask a direct question to push toward solutions: "What is one small change we can make next week to reduce this stress?"
- **When team members stay silent:** Give people two minutes during the call to write down thoughts silently on the retro board before opening the floor for speaking.

What To Do After The Retro

Capturing retro notes is useless if those notes sit in a folder and gather dust. You must connect your findings back to real work.

1. **Convert Actions into Tickets:** Immediately create real tasks in your team backlog for every agreed change. Do not leave them as bullet points in a summary document.
2. **Translate to Saved Hours:** Calculate how much engineering time your fixes will save. Show how process improvements reduce stress and save time every month.
3. **Share the Fix Publicly:** Drop a short update in your team Slack channel or email list. Explain the key findings and the process updates you put in place. This builds trust across teams by showing you take continuous improvement seriously.

Technical Leadership & Coaching Services

Managing modern data teams and executive expectations requires more than technical execution. Moving from firefighting to strategic leadership demands clear boundary setting, structured communication, and proactive team development. We help data and engineering leaders scale their influence and lead with clarity.

Core Leadership Focus Areas

- **Executive Alignment & Managing Up:** Translate complex technical debt, data quality risks, and architecture choices into business metrics executive teams value.
- **Stakeholder & Peer Collaboration:** Align cross-functional dependencies, set clear project boundaries, and resolve lateral friction without friction.
- **Team Delegation & Capacity:** Shift from tactical coding and reactive troubleshooting to coaching high-performing teams with clear accountability.
- **Burnout & Boundary Management:** Insulate your team from unrealistic executive deadlines and protect deep-work focus time.

Targeted Programs & Audits

- **Technical Leadership Audit:** A dedicated 60-minute diagnostic session evaluating your team structure, delegation habits, and stakeholder communication to build an actionable 14-day leadership roadmap.
- **The 60-Minute Career Audit:** A focused diagnostic session to evaluate your career trajectory, resolve immediate leadership bottlenecks, and build a 90-day execution plan.
- **1-on-1 Leadership Mentorship:** Ongoing private coaching for new and established tech leaders navigating corporate politics, team conflict, and promotion paths.

Is Strategic Support Right for You?

- You face constant executive pressure to deploy unvetted tools without adequate data foundations.
- You spend your week firefighting operational issues instead of driving long-term technical strategy.
- You want structured mentorship to transition from an operational manager to an executive-level data leader.

Ready to Build Your Strategy?

Let us discuss your current team challenges and map out your next steps.

- **Discovery Call:** [Schedule a Free 30-Minute Intro Call](#)
- **Coaching Services:** [Explore Full Coaching Programs](#)
- **LinkedIn:** [Connect with Amy](#) | [Follow Company Page](#)