Publish your story on The AI Reliability Journal

We started The Reliability Journal with the belief that AI failure knowledge should be transferable between engineers-in-the-loop. That’s why we’re showcasing candid and technically rigorous production stories and the trade-offs involved in building and operating increasingly complex AI systems.

The most useful articles start with some sort of an unexpected event, like when an AI agent behaved in a way nobody in your team anticipated, or a small change caused a big problem in production; perhaps you saw a sudden spike in latency, memory usage, or errors, and you're not sure why or maybe you thought you fixed one problem, but it ended up causing another.

We're also interested in the decisions you made to prevent failures and how you handled situations that could have gone wrong. Of course, not every story has to involve a major disaster, we just want to learn from each other's experiences so we can build more reliable AI systems together.

A Reliability Journal entry should help the reader (assume it’s another fellow engineer-in-the loop, just like you) follow your journey and understand:

- ☐ **What happened?**
  What was the first signal that prompted you to think there might be a problem?

- ☐ **What was your first intuition?**
  Don’t be afraid to share the wrong debugging sessions or the dead ends you hit because those are valuable as well. We know debugging isn’t a straight line and

most "Aha!" moments are often preceded by stuff that didn't work.

- ☐ **What was actually happening?**

  We encourage you to go beyond the standard root cause analysis and explain how components interacted to produce the observed behavior.

- ☐ **What did you change?**

  Explain the fix, but also potential trade-offs.

- ☐ **What are your lessons?**

  Leave readers with a reusable "recipe" that they can take away and apply immediately to solidify their AI system reliability.

Don't forget: you write for other engineers, so feel free to include diagrams or code snippets where you feel like they can better explain the situation. You do not need to disclose confidential information (anonymize customer names, normalize numbers, and simplify proprietary implementation details) because our readers care more about the engineering path you took from failure to remediation

A typical story

> 💡*If the system behaved exactly as everyone expected, there probably isn't a Reliability Journal story yet.*

But if there is one, it usually consists of 1,500–2,500 words and follows a common thread: (1) System Architecture, (2) Signal Recognition, (3) Investigation Process, (4) Resolution Plan, and (5) Lesson Learned. This is just a recommended structure, but we're not rigid about following it as long as you make sure another engineer reading your entry understands both what failed and how you figured out the solution.



This is important: it would be hypocritical of us to demand 100% human-authored content from another fellow engineer-in-the-loop when so much of our daily work implies, in some shape or form, co-existing with an AI coding agent (sometimes even more 😁).

However, while you can use AI for editing your drafts, to stay aligned with our editorial standards, we kindly ask you for at least 75% of the text to remain human-authored according to https://www.pangram.com/ (note the Pangram score for this document).

This respects the other contributors' effort to make sure The AI Reliability Journal is mostly human-authored, as well as respecting readers' preferences to read what other fellow engineers-in-the-loop are experiencing, without sounding heavily AI-redacted - we believe 25% AI-generated content is a fair limit.

Got a story to tell? Pitch us at hi@reliabilityjournal.ai. We review every submission and will follow up within 48 hours to discuss next steps.