



Am I the problem? Interviewing another team to find out.

2026-08-07

I'd like to thank [ngrok](#) for supporting my "art" (weird ranty tech blog posts). They paid me to write this post, but the opinions and stories here are my own.

Many years ago, I was working in DevOps at Cisco. This was my first job out of college, so I had all the energy and wisdom of a sledgehammer. I fondly remember eating those Sysco burrito bowls from the Cisco cafeteria and getting flakes of fried tortilla all over my \$3,000 laptop.

A good friend of mine referred me to that job, and it was great working with a friend, for a while at least. Unfortunately, we just argued about so much stuff: git branching strategy, fields on Jira tickets, the "definition of done", architecture, security, timelines, priorities, everything. After two years of this, I couldn't handle it anymore. I left that job, got a therapist, and lost contact with that friend.

Have you had your fair share of technical arguments? With software being so abstract, I feel like there are so many abstract hills to die on. But maybe that's just me? Or just the teams I'm on? There's one sure-fire way to find out: interviewing a similar team at another company. Surely this simple activity won't cause an existential crisis and recontextualize the last few years of my life or anything.



Soir Bleu (1914) by Edward Hopper.

I work on an infra team as my day job, but back in April, I got an opportunity to interview the infra team at [ngrok](#). Y'know, the thing that lets you serve a website on localhost? Lately they've diversified into other developer networking tools. Anyway, my plan was to write about how other teams approach technical disagreements, since my past work experience has involved a lot of those. Once we sorted out conflicts with our respective on-call rotations, I met with all four of them and ended up with thirteen pages of notes.

Unless you're a consultant, it's pretty rare for us to meet people doing our same job in their natural habitat. You might meet some domesticated tech workers at a conference, showing off their tricks in fancy slide shows. Always presenting solved problems, while the unsolved problems remain, like worms under an unturned rock. So I was excited to enter a parallel dimension and talk to another infra team, which mirrored my own team in many ways.

Name	Fav emoji	Interests
James		Green builds, DevEx, cats
Stacks		Ownership, standardization, infra

Name	Fav emoji	Interests
Alex		Pragmatism, <i>Crucial Conversations</i> , infra
Sabrina		Standardization, CI/CD, tennis

My first interview was with James 🐱, who had been with the company for four years. He's into photography, cat fostering, and sushi, an extremely synergistic set of hobbies. He's fostered hundreds of cats, sometimes as many as seven at a time. His Slack display name right now is literally "Seven Cats in a Trenchcoat". To feed all those hungry mouths, he keeps the developers happy by working on pipelines and other developer tooling.

How was I supposed to get a guy who has room in his heart for hundreds of cats to spill the tea about workplace conflicts? I was up for the challenge. James brought up that the infra team recently had a discussion about whether to adopt [Argo Rollouts](#) or whether to build a custom in-house tool, augmenting their existing [Buildkite](#) CI/CD setup. This must be it! We found some conflict. In my notes, I created a subheading called "Argo-gate" and eagerly pressed James for more details.

The context is that ngrok has a tricky update rollout process. They have a go-lang service called Mux which runs in each region and is holding hundreds of thousands of TCP connections. Replacing this go-lang process with a new one requires waiting several hours for connections to drain. In the early days, this update process was controlled by a script running on a developer's laptop. If their wifi cut out, the update would fail. *Anything* was better than that, so they ended up with the foundations of their current Buildkite CI/CD system. But they aren't super happy with it now that the company has more products using that same system. The team agrees the CI/CD needs improving, but disagree on how to do it. More custom tooling around Buildkite would be easier to adopt, but using Argo would force a rewrite of those shaky foundations.

"Who is on each side?" I asked, like a bloodthirsty Roman Vestal asking about a gladiator bout. It's not so simple, according to James.



Pollice Verso (1872) by Jean-Léon Gérôme.

- Stacks 🐢 doesn't want to burn innovation tokens on custom tooling, so prefers Argo but understands there are other factors at play.
- Alex 🙋, ever the pragmatist, thinks it'd be too difficult to gather buy-in for a rewrite, preferring to build on the existing system. Alex and Stacks 🐢 have been friends since high school, so it's not like they are opposing factions.
- Sabrina 🚀 hasn't used Argo before, but usually sides with Stacks 🐢 on the importance of standards. She's prepared for either outcome.
- James 🧐 often sides with Alex 🙋 when it comes to developer happiness (like avoiding a rewrite), but he also sees the pros and cons and would be happy with either decision too.

Maybe these people are just genuinely okay with either outcome? Surely not, right? At this point I was worried I wasn't getting the full story, since it was so unlike my own experiences of workplace conflict. I had no evidence, but my brain just couldn't imagine that a topic like this wouldn't lead to some arguing and hurt feelings. I figured he must have been over-simplifying or straw-manning them, or downplaying the level of disagreement.

The next interview was Stacks 🐢. Maybe he would spill the beans I'm looking for? Apparently not, he corroborated James' account of the situation and we had a lovely conversation about team boundaries and ownership. How frustrating!

Next was Alex 🙋. Alex opened up and even recounted big-picture topics discussed in his most recent meeting with his manager, which was super interesting but it wasn't conflict like I was looking for.

And finally was Sabrina 🚀, who mentioned she does have a lot of opinions about how things should work based on being ex-Amazon, but emphasized she was being open minded and

listening on the Argo subject.

Maybe they would get deadlocked on the decision and that would cause conflict? Apparently not, while writing this post they came to an agreement not to adopt Argo and have been making great progress since then.

Every single team member agreed it's a tricky situation. They all agreed there were pros and cons to both sides. Their descriptions of their own viewpoints all align with their descriptions of each others' viewpoints. They aren't even mad at people who disagreed with them!

Reflection...

As soon as my last Zoom call ended, I closed my laptop and laid on my bed, arms crossed behind my head, staring at the ceiling. What was I supposed to do now? I searched so hard to find conflict that I expected would be there, which made me wonder: *why* did I expect that? The only response to my question was the indifferent whirring of my desk fan, keeping my upstairs bedroom/office tolerable in the heat of summer.

I started thinking through my experiences with conflict, starting with my first internship, even before that Cisco job. I was hired as a software engineer despite being just a college freshman, with only one semester of computer science courses under my belt. As far as conflict resolution skills, my only tactic was complaining to a teacher. At this job, there was a lot of conflict. The details are lost to time, but there were definitely whiteboard diagrams being pointed to emphatically and disagreements over Jira settings. Being the precocious junior dev that I was, I definitely initiated my own share of arguments. Like one time, a senior coworker mocked out the database calls in a unit test. All of our other unit tests don't do this, so I requested changes on his pull request. He dismissed my review and merged it anyway. I was inconsolable. The old tactic of complaining to the teacher (my manager) didn't work that time; he expected me to deal with the conflict like an adult. So instead of doing that, I just continued to build animosity toward that coworker for a year before quitting.

I wearily sat up in bed, making eye contact with myself in the mirror hanging slightly askew on my closet door. The whirring of my desk fan, while unchanged, suddenly took on a different timbre to me, more like a droning or nagging reminder of the inescapable heat.

After another string of conflicts at Cisco and one lost friendship, I went to work for a university, picking more fights along the way. What domain name should we pick? What should our

password policy be? Should we pick this vendor or that vendor? More arguments, more 1:1s, another job search, and another resignation letter, just like the last two jobs.

The mirror reflected myself back at me, but it also reflected my surroundings: A desk, a dresser, an over-full laundry basket, a stack of un-filed documents, a box I haven't unpacked yet.

What about my current job? I like my teammates well enough. Our infra team is about the same size and tenure as theirs. I work for a startup in a similar phase, which is also making a similar AI play and experiencing similar growing pains. So why does my team have more conflict than theirs?

If I unfocus my eyes, the image of myself in the mirror starts to blend into the background. My team has me on it. Their team doesn't.



Stańczyk (1862) by Jan Matejko.

Are you the argumentative person on your team? Or know someone who is? If they're like me, they aren't totally clueless. I've known I'm an argumentative person. The therapist I got right after that Cisco job was quick to point that out. But I hadn't really seen it revealed in such stark contrast before.

For the last few years, I have been reading a lot about conflict, relationships, burnout, volunteer organizing, and other fields. I thought I was doing it for general self-improvement, but this experience of interviewing another team has recontextualized that as part of a broader journey of managing conflict in my life. Fortunately now I can see the ways in which I've been making progress and which pieces of advice have been most helpful. I hope these tips prove as useful to you as they have been to me.

Tips from the field of relationship counseling

If you squint, our relationships to our coworkers look similar to our romantic relationships in many ways (hopefully not all ways). Therapists the world over have stable jobs because we're shit at both. I'm sure therapists have different methods of dealing with different kinds of relationships, but I'm no therapist.

One popular method of relationship counseling is the [Gottman Method](#). [Lots of therapists](#) specifically advertise it as one of their specialities. It's hard to describe since it's basically just a bunch of stuff that Drs. John and Julie Gottman recommend based on their long and prolific careers. I'm just going to highlight two tips from the Gottman Method that I've found helpful for workplace conflict: the Gottman ratio and "paraphrasing".

The "[Gottman ratio](#)" says you need a five-to-one ratio of positive to negative interactions in order to have a healthy relationship. It's based on longitudinal studies showing couples who consistently fell below that 5:1 ratio were significantly more likely to divorce within six years (although the predictive power of this metric [is suspect](#)). The thinking is: if we have a lot of positive interactions, then it makes us more able to view neutral interactions as neutral, and negative interactions as mistakes with good intentions, keeping the ratio high in a positive feedback loop.

When applied to the workplace, this emphasizes the need to intentionally create opportunities for all those positive interactions. In remote workplaces, this could involve using something like [Donut](#) to get random 1:1 meetings with coworkers, having a "social" meeting where work-talk is forbidden, and having processes that minimize those negative interactions, like enforcing good [pull request etiquette](#). Multiple people on the ngrok infra team cited their good rapport with each other, and also mentioned their social meetings, such as building [tier lists](#) together or playing [GeoGuessr](#) cooperatively.

First Gone												
A												
B												
Last Gone												
Straight into the trash												
Idk what these are, weren't in my neighborhood												

A tier list of Halloween candy that the whole team agreed to within 15 minutes without major argumentation. When I see this, I want to scream that Junior Mints are just small York Peppermint Patties.

"Paraphrasing" is a communication technique which is a small piece of the broader [Gottman-Rapoport Intervention](#). Relationship counselors use this intervention when people seem to be talking past each other, to break them out of a cycle of constant conflict. There are a bunch of rules and steps if you were doing the activity for real, but I've found success just doing the paraphrasing part. It's simple: when discussing a divisive topic, before either of you can respond, you have to paraphrase or summarize what the other person said to their satisfaction. Skilled listeners are able to do this naturally, like how the infra team members were able to easily summarize each others' perspectives on the Argo Rollouts issue. For others like me, practicing paraphrasing has been helpful for honing my listening skills, and I'll often use it in 1:1s with managers.

Tips from the book *Crucial Conversations*

While the Gottman Method is not typically used in the workplace, the authors of *Crucial Conversations* proudly claim their methods can be used for either personal or workplace relationships. "[Crucial Conversations](#)" are conversations that are high-stakes, but are approached strategically with certain attitudes, mindsets, and techniques that the book describes. You can find descriptions of the techniques online, but I recommend reading the whole book to give you the full picture.

In college, I had a class devoted to software engineering where we were taught about this book. Too bad I ignored that advice until about a year ago when I finally read it. Since then, I've had many successful crucial conversations. Like that time I healed a rift in trust with another team, that time I resolved a conflict over AI usage, and that time I possibly saved hundreds of thousands of dollars in wasted effort, each with a single conversation. I could go on.

Without my prompting, Alex 🙋 brought up this book during our interview. The whole company read the book together in the past, and Alex referenced it as something he wants to utilize more to drive alignment. It's one thing for a team to know that certain ideas exist, but it takes active involvement from people like Alex to put the ideas into action.

Tips from the field of mutual aid organizing

About a year ago, I started getting involved in mutual aid work, specifically food distribution. The corporate world with all of its hierarchies and coercion couldn't be more diametrically opposed to the concept of mutual aid, but I think mutual aid communities have a lot to teach us. If you're struggling to get a bunch of similarly-minded tech workers to work together *for money*, imagine getting a bunch of wildly different people to work together *for free*, such as crews of people clearing debris and mucking flooded homes in the wake of [Hurricane Helene](#).



Illustration from *Mutual Aid: An Illuminated Factor of Evolution* (2021) by [no bonzo](#).

The guy who literally wrote the book on the subject (well other than [Kropotkin](#)) is Dean Spade, whose book *Mutual Aid* spends nearly half of the pages discussing how to deal with conflict. Spade emphasizes that conflict is a normal and necessary part of making sure everyone's voice is heard, but it's extremely important to handle it properly. Unlike a job which may be your only way to pay rent or afford healthcare, volunteers can simply decide to leave if a group doesn't handle conflict well. I'll focus on just two of the many tips in that book: how burnout relates to conflict and the value of building consensus.

Burnout

Late last year, I started researching books about burnout to help myself get through a nasty bout of it. In all my research, I never found a better explanation than in *Mutual Aid*:

Burnout is the combination of resentment, exhaustion, shame, and frustration that make us lose connection to pleasure and passion in the work and instead encounter difficult feelings like avoidance, compulsion, control, and anxiety.

Burnout is created or worsened when we feel disconnected from others, mistreated, misunderstood, ashamed, overburdened, obsessed with outcomes, perfectionist, or controlling. Burnout is prevented or lessened when we feel connected to others, when there is transparency in how we work together, when we can rest as needed,

when we feel appreciated by the group, and when we have skills for giving and receiving feedback.

Giving and receiving feedback is one key area I've struggled with (like my coworker who mocked out the database calls), but getting it right at my current job has helped me immensely. Spade seems to agree it's key, since he mentions feedback *everywhere*, even in other books he's written. When we can't effectively give and receive feedback, we get into more conflicts, which lead to the resentment, shame, and frustration indicative of burnout. When we can do it effectively, such as using tips from the Gottman Method or *Crucial Conversations* we avoid burnout and thus reduce conflict.

Burnout didn't come up in my interviews, but Spade points out how important it is to notice the signs of burnout before it sets in, since burnt out people can inadvertently cause a lot of damage that could have been fixed earlier with a few crucial conversations or some extra PTO.

One time, I got into an argument with someone about using unsupported infra tools. I was already pretty burnt out and my entire team was too. I started noticing the signs: we started gossiping about this person, feeling misunderstood, frustrated, and disconnected. There was only one way out of this spiral. So I sucked it up and sent a DM to them, using techniques from *Crucial Conversations* to ensure I created space for them to share their feelings too. I'm not gonna say we fully see eye-to-eye now, but I think we understand each other to the point where interacting with them is usually positive rather than burnout-inducing like it was before.

Building consensus

Mutual aid groups often operate based on [consensus-based decision making](#). In contrast to ["the boss decides"](#) or ["majority rules"](#) which we might be used to, consensus-based decision making requires all members of a group to agree to a decision. This may seem unheard-of, but most pull requests involve a kind of consensus, since a single member can "block" the PR. This respects the autonomy of all members of a team, and ensures everyone's on-board with the outcome (maintaining that code).

One example where consensus is useful is when dealing with this common issue facing infra teams: enforcing standards. Standards can be as broad as "We only use AWS as our cloud provider" to as narrow as "Every git repo must contain a `glorp.yaml` file". The ngrok infra team cited this as a common source of conflict with other teams. I know the feeling on my own team. Rather than relying on corporate hierarchies or purely technical solutions (like extra CI/CD checks), consensus-based decision making offers a novel but kinda obvious solution:

just talking to teams. Stacks 🦋 mentioned doing this in staff meetings regarding silencing some infra alerts to reduce [alert fatigue](#). Sabrina 🚀 mentioned meeting with development teams during the rollout of some CI/CD changes to understand and adapt to their needs. Alex 🐶 also identified the need to build consensus, mentioning the concept by name.

Coming to decisions like this, in a way that respects the autonomy of your coworkers, can minimize conflict and also result in better adherence to the standard moving forward, as Spade explains:

When other people make decisions for us and we don't get to raise concerns or disagreements, we are less likely to want to implement them. ... When we get to look at a proposal together and tell each other how it might be improved, hashing out our best ideas until we have something that we all like or at least can live with, we are more likely to vigorously do what we all decided, instead of drifting apart or failing to follow through.

When I finally broke my gaze from my mirror and rolled out of bed, I opened my editor and started writing this post, asking the question, "am I the problem?".

Yes, I am.

Just like with Alcoholics Anonymous (a famous example of mutual aid), the first step is admitting that. This post is me working through step four, making "a searching and fearless moral inventory" of myself. Step nine involves making direct amends to people we have harmed. Remember that friend from Cisco I stopped talking to? I had been thinking about messaging her again for like a solid year, but I really struggled to overcome the shame of getting into all those pointless arguments and ruining a friendship as a result. Would she be holding a grudge of her own? Would she even see my message? Well last week, I decided to chance it.

We had a lovely conversation: life updates, venting about our current jobs, AI woes, discussion of mutual friends, and she even told me how I could hack the product she works on. What did she think of all that head-butting we did at Cisco? The least I could do is give her the final word: "Lmao nah I barely even remember the ol' Cisco days."
