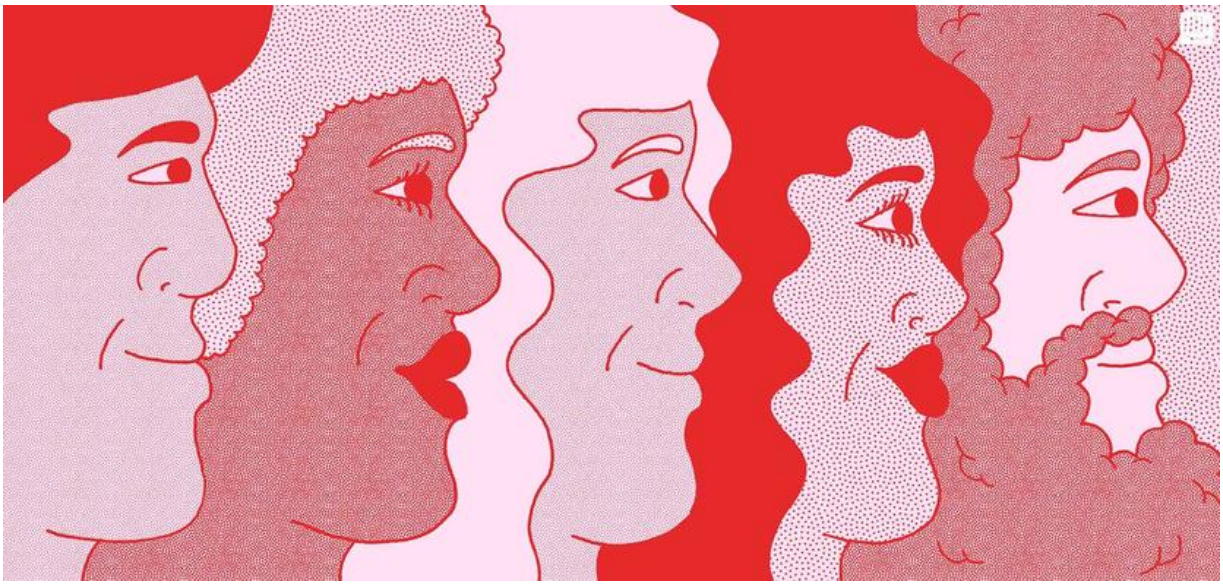
Engineering a culture of psychological safety

By **Des Traynor**, blog.intercom.com
June 20th, 2017



When I worked for Google as a Site Reliability Engineer, I was lucky enough to travel around the world with a group called “Team Development”. Our mission was to design and deliver team-building courses to teams who wanted to work together, better.

Our findings were later published as Project Aristotle. The biggest finding was that the number-one indicator of a successful team wasn’t tenure, seniority or salary levels, but **psychological safety**.

Think of a team you work with closely. How strongly do you agree with these five statements?

1. If I take a chance, and screw up, it will be held against me

2. Our team has a strong sense of culture that can be hard for new people to join.
3. My team is slow to offer help to people who are struggling.
4. Using my unique skills and talents come second to the objectives of the team.
5. It's uncomfortable to have open honest conversations about our team's sensitive issues.

Teams that score high on questions like these can be deemed to be "unsafe". Unsafe to innovate, unsafe to resolve conflict, unsafe to admit they need help. Unsafe teams can deliver for short periods of time, provided they can focus on goals and ignore interpersonal problems. But eventually, unsafe teams will break or underperform drastically because people can't introduce change.

Unsafe teams will break or underperform drastically because people can't introduce change.

Let's highlight the impact an unsafe team can have on your team's individuals, through the eyes of a recent, fresh-faced and enthusiastic graduate who finished top of their class.

This imaginary graduate, we'll call her Karen, was reading about an optimization that could reduce low-level locking in distributed databases, and realized it could be applied to the service her team worked on. She decided to test it out, it resulted in a 15% CPU saving on the test cluster, and in her excitement, decided to roll it out to production. Because it was a change to a database configuration file, it didn't go through the usual code-review process.

Unfortunately, it caused the database to hard-lock-up, causing a brief, but total outage of the website. Thankfully, her more experienced colleagues spotted the problem, and rolled back the change inside of 10 minutes. Being professionals, this incident was mentioned at the weekly “post-mortem” meeting.

1. “If I take a chance, and screw up, it’ll be held against me”

At the meeting, the engineering director let everyone know that causing downtime by chasing small optimizations was unacceptable. Karen was described as “irresponsible” in front of the team, and the team suggested ways to ensure it wouldn’t happen again. The engineering director forgot about this interaction quickly after. But Karen would never forget the exchange. She would never try to innovate without explicit permission again.

2. “Our team has a strong sense of culture, and it’s hard for new people to join”

The impact on Karen was actually magnified because no one stood up for her. No one pointed out the lack of code reviews on the database configuration allowed this to happen. No one pointed out the difference between highlighting one irresponsible act and labelling someone “irresponsible”. The team was so proud of their system’s reliability, defending their reputation was more important than a new hire.

Karen learned that her team, and manager didn’t have her back.

3. “My team is slow to offer help to people who are struggling”

As Karen was new to “production”, she had no formal training in incident management, production hygiene, let alone troubleshooting distributed systems. As her team was mostly made up of people with decades of experience, they had never needed training, or new-hire documentation. There were no signals that it was OK for a new graduate to spend time learning these skills.

Karen developed Imposter Syndrome. She didn’t understand how she passed the hiring process, and frequently wondered why she hadn’t been fired yet.

4. “Using my unique skills and talents come second to the goals of the team”

Karen’s background was in algorithms, data structures and distributed computing. She realized the existing system as a whole was suboptimal, and would never handle load spikes.

The team had always blamed the customers for going over their contracted rates, which is like blaming weathermen for rain during an Irish barbecue.

Karen proposed a new design, based on technology she’d used during her internship. Her co-workers were unfamiliar with the new technology and considered it too risky. Karen dropped her proposal without discussion. She wanted to write code and build systems, not have pointless arguments.

5. “It’s uncomfortable to have open, honest conversations about our team’s sensitive issues”

When a large customer traffic spike caused the product to be unavailable for a number of hours, the CEO demanded a meeting with the operations team. Many details were discussed, and Karen explained that the existing design meant it could never deal with such spikes, and mentioned her design. Her director reminded her that her design had already been turned down at an Engineering Review, and promised the CEO they could improve the existing design.

Karen discussed the meeting with one of her teammates afterwards. She expressed dismay that the Director couldn’t see that his design was the root-cause of their problems. The teammate shrugged, and pointed out that they had delivered a really good service for the last five years, and had no interest talking about alternate designs with the director.

Karen decided to head home early, and look for a new job. When she left, the company didn’t miss her. After all, she was “reckless, whiny and had a problem with authority”. They never realized she had the design that could have saved the product from the customer exodus that follows repeated outages.

How to build psychological safety into your own team

So, what is special about Engineering that leads us to drive away so many promising engineers, and allow so many others to achieve less than their potential?

We need to balance respect for our culture, with an openness to change it as needed.

We know that a strong sense of culture, shared understandings and common values are required to succeed. So we need to be able to balance that respect for our culture, with an openness to change it as needed. A team - initially happy to work from home - needs to change how they work if they take on some interns. A team - proud that every engineer is on-call for their service - may need to professionalize around a smaller team of operations-focused engineers as the potential production impact of an outage grows.

We need to be thoughtful about how we balance work people love, with work the company needs to get done. Good managers are proactive about moving on an engineer who is a poor fit for their team's workload. Great managers expand their team's remit to make better use of the engineers they have, so they feel their skills and talents are valued. Engineers whose skills go unused grow frustrated. Engineers given work they are ill-equipped to succeed, will feel setup to fail.

Make respect part of your team's culture

It's hard to give 100% if you spend mental energy pretending to be someone else. We need to make sure people can be themselves by ensuring we say something when we witness disrespect. David Morrison (Australia's Chief of the Army) captured this sentiment perfectly, in his "the standard you walk past is the standard you accept" speech.

Being thoughtless about people's feelings and experiences can shut them down. Some examples where I've personally intervened:

- Someone welcomes a new female project manager to the team, assumes they aren't technical and uses baby words to explain a service. I highlight the new PM has a PhD in CS. No harm was intended, and the speaker was mortified that their good-humored introduction was taken the wrong way.
- In a conversation about people's previous positions, someone mentioned they worked for a no-longer-successful company, and a teammate mocked them for being "brave enough" to admit it. I pointed out that mocking people is unprofessional and unwelcome, and everyone present understood a 'line' that hadn't been visible previously.
- A quiet, bright engineer consistently gets talked over by extroverts in meetings. I point out to the "loud" people that we were missing an important viewpoint by not ensuring everyone speaks up. Everyone becomes more self-aware.

It's essential to challenge lack of respect immediately, politely, and in front of everyone who heard the disrespect. It would have been wonderful had someone reminded Karen's director, in front of the group, that the outage wasn't a big deal, and the team should improve their test coverage.

Make space for people to take chances

Some companies talk of 20% time. Intercom has "buffer" weeks, in between some of our 6-week sprints. People often take that chance to scratch an itch that was bothering them, without impacting the external commitments the team has made. Creating an expectation

that everyone on the team should think outside the box, and ensuring that the whole team can go off-piste at the same time, is a powerful message.

Be careful that “innovation time” isn’t the only time people should take chances. One company in the transport industry considers “innovation time” to be 2:30 p.m. on Tuesdays.

Imagine how grateful Karen would have been, had a senior engineer at the Engineering Review offered to work on her design with her, so it was more acceptable to the team. Improve people’s ideas, rather than discounting them.

Make it obvious when your team is doing well

I love how my team writes goals on Post-It notes at our daily standups and weekly goal meetings. These visible marks of success can be cheered as they are moved to the “done” pile.

But we can also celebrate glorious failure. Many years ago, when I was running one of Google’s storage SRE team, we were halfway through a three-year project to replace the old Google File System.

We can also celebrate glorious failure.

Through a confluence of bad batteries, bad firmware, poor tooling, untested software, an aggressive rollout schedule and two power cuts, we lost a whole storage cell for a number of hours, and though all services would have had storage in other availability zones, the team spent three long days, and three long nights rebuilding the zone. Once it was done, they – and I – were dejected. Demoralized. Defeated. An

amazing manager who was visiting our office realized I was down, and pointed out that we'd just learned more about our new storage stack in those three days, than we had in the previous three months. He reckoned a celebration was in order.

I bought some cheap sparkling wine from the local supermarket, and with another manager, took over a big conference room for a few hours. Each time someone wrote something they learned on the whiteboard, we toasted them. The team that left that room was utterly different to the one that entered it.

I'm sure Karen would have loved appreciation for discovering the team's weak non-code test coverage, and their undocumented love of uptime-above-all-else.

Make your communication clear, and your expectations explicit

Rather than yelling at an engineering team each time they have an outage, help them build tools to measure what an outage is, a Service Level Objective that shows how they are doing, and a culture that means they use the space between their objective, and reality, to choose the work that will have the most impact.

Ask for a specific commitment, rather than assuming everyone agrees on its urgency.

When discussing failures, people need to feel safe to share all relevant information, with the understanding that they will be judged not on how they fail, but how their handling of failures improved the team, their product and the organization as a whole. Teams with operational

responsibilities need to come together and discuss outages and process failures. It's essential to approach these as fun learning opportunities, not root-cause obsessed witch-hunts.

I've seen a team paralyzed, trying to decide whether to ship an efficiency win that would increase end-user latency by 20%. A short conversation with the product team resulted in updates to the SLO, detailing "estimated customer attrition due to different latency levels", and the impact that would have on the company's bottom line. Anyone on the team could see in seconds that low-latency was far more important than hardware costs, and instead drastically over-provisioned.

If you expect someone to do something for you, ask for a specific commitment - "When might this be done?", rather than assuming everyone agrees on its urgency. Trust can be destroyed by missed commitments.

Karen would have enjoyed a manager who told her in advance that the team considered reliability sacred, and asked her to work on reliability improvements, rather than optimizations.

Make your team feel safe

If you are inspired to make your team feel more psychologically safe, there are a few things you can do today:

1. Give your your team a short survey, and share the results with your team
2. Discuss what "Safety" means to your team; see if they'll share when they felt "unsafe"

3. Build a culture of respect & clear communication, starting with your actions

Treat psychological safety as a key business metric, as important as revenue, cost of sales or uptime. This will feed into your team's effectiveness, productivity and staff retention and any other business metric you value.