

Blameless Culture: The Path to Agility in IT

You've inevitably heard the word blameless used in the context of technology and tech culture. The foundation of DevOps and agile and many other methodologies is around blameless post-mortems. Retrospective meetings and recaps that allow for trust and honesty when discussing what went wrong and what went well.

Here's the issue: being blameless is really difficult. Like, really, really difficult.

Battling Human Nature

Blameless meetings in many organizations are more like "don't blame me" meetings which erodes the very foundation of what blameless culture is all about. A societal challenge is getting past the "Not it!!" mentality and embracing the issues without seeking a person to pin it on who isn't ourselves. People often mistake not taking responsibility with being blameless. Just because you don't assign blame, constantly un-assigning yourself from it is not a blameless approach.

It's not easy to admit we made mistakes. Trust me...

A Personal Story: "Wait, that was production?!"

It was 9:30 AM on a Tuesday. Two Active Directory administration windows open, one for production, and one for the test domain with the identical OU structure (to look just like production). As I clicked the option to upgrade the Active Directory version to Windows 2000 native mode and proudly clicked the Change Mode button (clearly stating "this operation cannot be reversed") and the following acknowledgement that it was irreversible, I sat back in the chair and smiled...for a moment.

Then I checked the other window to what I thought was production...and saw the domain version was mixed mode (legacy). But then I also noticed the domain name was the test domain. I had updated production in the middle of business hours during the potentially busiest login period of the day.

I stood up and took a lap around the cubicle. Once I stepped back in and confirmed it had happened, I called my systems architect who worked with me. "James, I made a mistake. I just updated production instead of test."

James calmly said "Ok. Let's get some folks looking at it and figure out if anything happened and what our options are if something broke". James understood blameless culture. We survived the change (without issue, luckily) and I learned that rallying the team around an issue was better than seeking blame. We would work together on other issues that did have far reaching effect at other times, and the blameless culture and teamwork got us through those events.

Hire the Person that Made a Big Mistake - They Won't Want to Make Another One

Making mistakes and learning to change process and recovery procedures meant building better IT. As a systems architect and operations team member, our culture grew stronger under blameless acceptance of issues. I also witnessed the very negative results of not using this tactic. A manager at one time told me that even though we had a large, preventable situation occur from human error,

we don't remove the person. We teach them to not have it occur again. If you make the same mistakes again...and again, well, there isn't a blameless team around who will not take notice and perhaps have to make some changes in your role.

Test-driven development and test-driven infrastructure are wrapped around the foundations of finding the error first and then working back from there. Seek a problem and then find resolution. The same goes for teamwork and production deployments. To move faster, you have to be confident that you are testing and also able to work together as a team without fear of retribution when issues occur. Some issues are avoidable, and some are not. Be blameless before you find the solution and then when you find the cause, accept that it happens...but should happen less. The real leaders then ask "how can we do it better next time?"

Leadership Defined: We Succeed; I Fail

Acknowledging failure or just challenges is healthy, Celebrating success as a team is also extremely positive. My way of encapsulating what it means to be a leader of people in a team goal is "We succeed; I fail". In other words, celebrate successes as a team, because we all did this together. When looking for where we may not have succeeded, look into yourself for what you feel could have been done better. There may be others involved in the tactical things that went wrong. The best outcome is a learning experience for everyone throughout the team.

Even when things go well, we should always ask how it could have gone better, or what would you do differently? This is the beginning of embracing a culture of experimentation. You have to know that things can fail and we can recover, without blame.