Saal: Simulacron-3	12:00	00:30	Day: 2	Track: Failosophy	nA

Title: **Building a culture of courage**

Subtitle:

Speaker: **linse**

Short: *Starting my new job, I was surprised about my colleagues reactions to my mistakes. No blame! People trust me to own my mistakes and fix them! Complex systems are brittle. Developers work in a constant state of new things-not-quite-working-yet. Programming in pairs and code reviews help us. But what about the quality of human interaction? How can we build a culture of blamelessness and why is it crucial for learning and for brave decision making?*

Long: On the first day of my new job, I contributed program code that went live to the website we are building. After a month, I contributed some code that caused alerts. I felt bad and it was reverted quickly, never making it to the site. Still, I was surprised about how my colleagues reacted. They did not blame me! I felt support. These people have got my back. I experienced that it is okay to make mistakes and people trust me to fix them. After my years in gradschool this was a refreshingly new experience! Just two weeks ago, somebody within the company received the annual prize, our three-armed-sweater, for the most spectacular way to bring down the site. We joked that if somebody does not break the site to get the prize, they are not daring enough in their coding. I am still nervous when pushing new code to the site, but it feels like an exciting anticipation and respect for my work. People working with complex systems know they are brittle, because they are complex. Code that works well is nice, but working code also means everything is done - how boring! Programmers and builders live in a constant state of things-not-quite-working-yet, pushing this boundary every day. Even though it might be just software, this can be a challenging environment. Programming culture is rife with stereotypes of lone-wolf coders. On hacker news, people like to ”well-actually” each other. I did it, and maybe you do it too. Quite often, we face our coding questions all alone. With computers, we fight unforgiving machines, and usually the mistakes in our systems are based on human decisions. Programming in pairs and code reviews are techniques to improve these decisions by adding an extra set of eyes and opinions. These tools are excellent to use when working on a challenging complex system together and are known techniques by now. But what about the quality of human interaction? How can we build a culture of blamelessness and why is it crucial for learning and brave decision making?