



Reliable Software Systems

Week 9: Team Culture



March 7, 2018
Alyssa Pittman
University of Washington Allen School

Motivating Example: Titan II

In 1980, the Air Force had a missile silo with a nuclear warhead in Arkansas.

Workmen doing maintenance forgot the torque wrench required by the official process, and used an old socket wrench instead.

The socket fell off and punctured the side of the fuel container.

Alarms went off but the control room couldn't pin down the cause of the leak.

Eventually the workmen told what had happened, but too much had leaked to be able to fix it.

...long story short, they evacuated, and BOOM. Read the rest:

<https://www.thisamericanlife.org/634/transcript>

What is company culture?

What is the mission/vision of the company?

How are decisions made?

How do people communicate with each other?

How do teams decide what projects to do or technologies to use?

What expectations are there for teams or employees?

What is the workspace like?

Psychological safety

“A shared belief held by members of a team that the team is safe for interpersonal risk taking” *

Why is this important?

- Asking questions helps us all understand a system better
- Suggesting changes makes a system better
- Admitting failures helps us all learn from them
- ...and fix them
- Giving feedback helps others grow
- Receiving feedback helps us grow

Design

Design first

Review designs!

Design for failure

Document the “why” since the code has the “what”

The price of reliability is the pursuit of the utmost simplicity.

- C.A.R. Hoare, Turing Award lecture

Simplicity

Solve a complex problem in the simplest way possible.

- YAGNI
- Stable not “boring”
- Don't write code for code's sake
 - Delete it instead!
- Solve problems rather than create problems
- Use frameworks and libraries

Be warned that being an expert is more than understanding how a system is supposed to work. Expertise is gained by investigating why a system doesn't work.

- Brian Redman, Bell Communications Research

Oncall is a team sport.

- George, a Google SRE manager

Emergency response

Don't panic!

You've tested your recovery plans and you can do this.

Or, it's a tough problem and you'll do the best you can.

Accept, or ask for, help.

Communicate.

Post mortems

A post mortem explains what happened during an incident, often including:

- A timeline
- The impact
- The root cause
- Context
- Lessons learned: what went well, what went poorly, what was lucky
- Action items

But...don't include finger-pointing and blame!

Operational load

Time spent on oncall response, bug fixes, maintenance tasks, etc rather than on design and feature work.

It takes resources to engineer or automate it away, but operational load grows as your service scales.

Don't let it grow too large, that frustrates the team and slows progress.

END