How to work together

without killing each other

…without killing each other

Why is it hard to work in a group?

Normalize

- Establish a set of rules that the group agrees to abide by
- These rules act as a *source of truth* for how the group behaves, manages work, and deals with conflict
- When in doubt: “What are our norms?”

Starting Norms

1. Share the mic
2. Don't interrupt - raise your hand
3. Share frustrations in a constructive matter
4. Write a GitHub issue
5. Ask for help
6. Commit frequently
7. Deploy early
8. Choose a code style and stick to it
9. Keep each other updated
10. Disagreements many, arguments few

Roleplaying Scenarios

Scenario #1

During a pair programming session, two pair partners are having trouble communicating ideas effectively. One person seems to have a good grasp of the goal and how to get there, but cannot effectively communicate it—their ideas and words are coming out jumbled. The other person is trying really hard to understand. The dynamic alternates between silence and frustration.

- **How would you respond when this happens?**
- **How would you respond if this happened multiple times?**

Scenario #1

- Communication is two-ways: everybody here has the technical ability
- If you can't communicate something well, it may be because you yourself don't understand something about it
- Be aware of your natural biases—it can be easy to blame miscommunication on language barrier or accent, when really that's a red herring, we're all learning to improve our technical vocab
- Takeaway: be upfront with your partner if you don't understand them, so that you can try to address it
  - "show instead of tell"
  - diagrams, writing it out, coding it out
  - using similar examples, pointing them out in docs / wherever

Scenario #2

In a standup one morning one team member says to another, "this'll be easy, you'll finish it in 10 minutes!" The task ends up taking that person the whole day.

- **What should standup look like the next day?**
- **If you were the team member that made the time estimate, how would you feel? What would you do after?**
- **If you were the team member that did the day-long task, how would you feel? What would you do after?**

Scenario #2

- Communicate throughout the day—did something unexpected go wrong? If so, bring it up. Even if not, there's no shame because...
- Time estimation is hard
- Programming is hard
- Needing more time than expected is not a reflection of technical skill

Scenario #3

The team has a problem with their database. A woman brings up a solution, and the team doesn't like it. Later that day, a man brings up the same solution, and the team loves it.

- **How would you respond when this happens?**
- **How would you respond if this happened multiple times?**
- **Consider your response as the woman, as the man who brings up the same idea later, and as a bystander just on the team in general.**

Scenario #3

- **Unconscious bias**
  - **Women tend to be ignored more than men**
  - **Women: expertise isn't trusted**
- **Amplify**

Up Next

- Create Team Contract
- Leverage our example Team Contract
- Answer the questions in Establishing Norms