

A close-up photograph of two hands typing on a silver laptop keyboard. The hands are positioned over the keyboard, with fingers pressing down on the keys. The laptop is open, and the screen is visible on the left side of the frame. The background is a wooden desk. The text "How to Work Together" is overlaid in white, bold, sans-serif font in the center of the image.

How to Work Together





... Without killing each other

Why is it hard to work in a group?

4 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 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 goes 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

- "Voluntold"
- 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. Someone brings up a solution, and the team doesn't like it. Later that day, a different team member 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 first person with the solution and the second person who brings up the same idea later, and as a bystander just on the team in general**

Scenario #3

- **Acknowledge ideas and viewpoints**
- **Amplify others**
- **Psychological Safety:** "A situation in which everyone is safe to take risks, voice their opinions, and ask judgment-free questions. A culture where managers provide air cover and create safe zones so employees can let down their guard."

Scenario #4

Someone(s) on the team stayed up all night working on the project, and you wake up to find over 1,000 new lines of code merged into the project. You feel frustration that agreed upon norms were not respected, and stressed that your contributions to the project may've been replaced by the new code (or now have merge conflicts).

- **How would you respond when this happens?**
- **How would you respond if this happened multiple times?**
- **How would you respond if this occurred in a future project in the workplace?**

The Career Success Solution

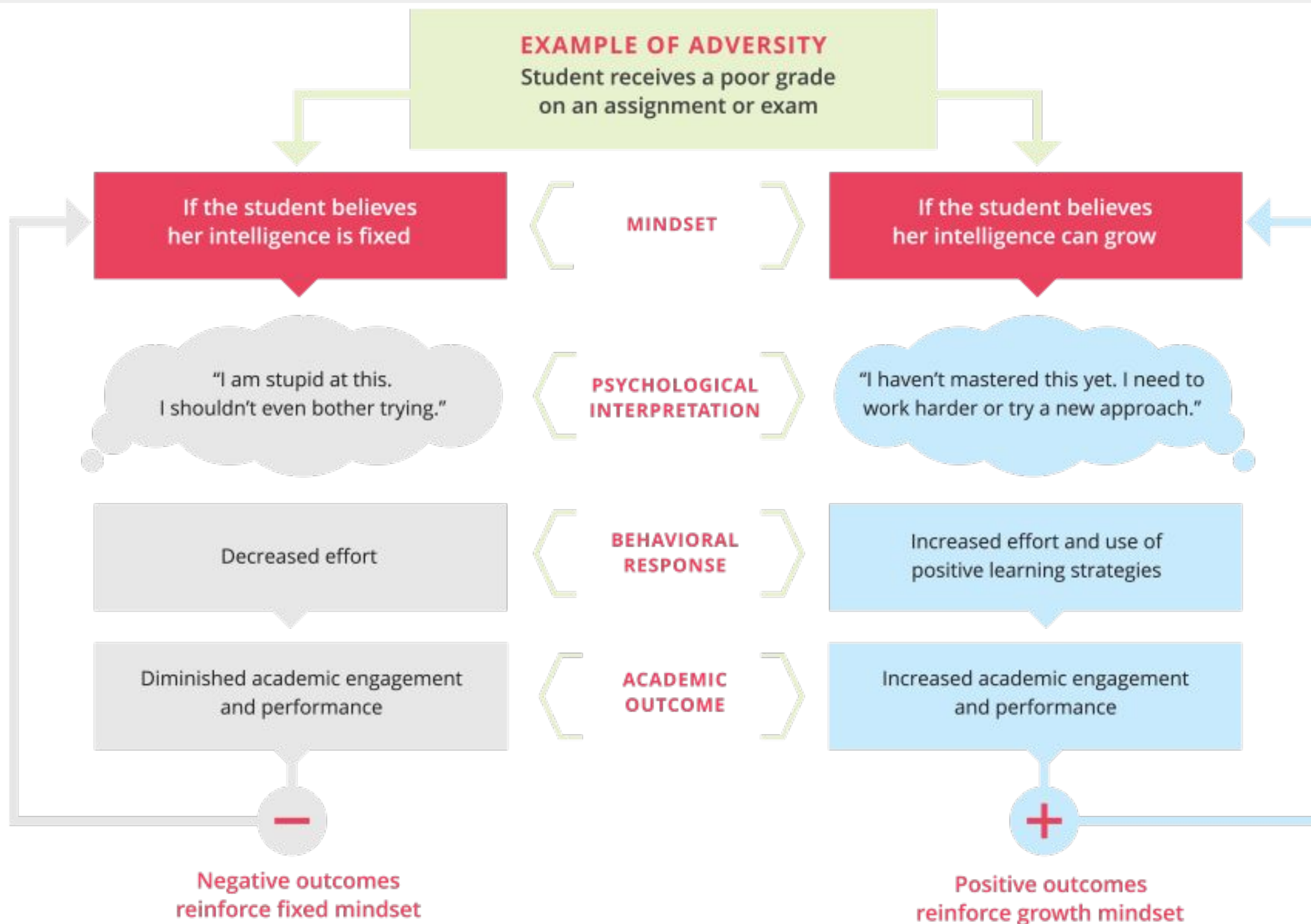
- If your team is struggling to maintain a professional level of communication, this is where you loop in your fellow and/or Program Lead
- There are clear cut strategies and practices we can implement
 - Starting with Team Contracts
- For interviews: think standups, code reviews, team contract, working with a team manager,

Before we finish

- **We have your best interests in mind and will support you to reach your goals in a reasonable manner**

**“We want to make you happy,
but we’re here to make you better”**

Adopting a *Growth Mindset*



Norms

- 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**

Starting Norms

- Create Team Contract
- Leverage our example [Team Contract](#)
- Answer the questions in [Establishing Norms](#)