

# Lecture 6

## First impressions

- First impression is the single most important thing that makes or breaks your time in a company/team.
- Unfortunately, most people are full of biases. If you are working really hard for the first 6 months and deliver an outsized amount of work for your role. You can often coast on the good name indefinitely unless you do something drastic to break it.

## Communication

- Professional communication is an iterated game. Any one heard of the prisoner's dilemma? Basically you are asked to pick between choices where everyone is better off as a whole if everyone is altruistic. However if someone else is altruistic and you are not, you can get slightly further ahead at the expense of making someone else much worse off.
- If it's just one round, you can afford to not be altruistic, however in an iterated game, you are constantly interacting and people will react to you based on how you behaved in the past.
- On the other hand, if you behaved well in the past, then people will give you more benefit of the doubt and give you more leeway if you are having a bad day.
- Therefore, this whole lecture we will focus on workplace communication and making sure everyone understands the value of first impressions, how to make them, how to work in a team, and how to give feedback.

## Disagreements

- It's common to have disagreements, and it's common to also not know how to deal with it. Professional disagreements are tricky to navigate. If you dislike someone who you met socially, it's not impossible to avoid that person in the future, and therefore sidestepping the whole issue.
- Professionally it's much more difficult to avoid the issues and often you are asked to deal with disagreements head on but also gracefully.
- Maybe you are not a confrontational person by nature. Maybe you think you don't know enough to disagree. Also maybe you just don't care enough to have a different point of view.
- Some scenarios:
  - What would you do if you feel like someone has a terrible structure to their code?
  - ... someone is missing some edge cases?
  - ... disagree with the product direction?
- Overall, there are two types of disagreements

- specific/concrete/provable: something typically is wrong with their code. In these cases, it's mostly just a simple matter of pointing out things that they are not considering, like a test case that would break their code, or questions about why they made certain choices. A lot of these disagreement happens in engineer's comfort zone, over text in a code repository.
- subjective/opinion based: some of these things might be code related, such as subjective design decisions or stylistic choices. In those cases, someone needs to make a decision based on relative merits and move on. Other aspects relates to decisions maybe you don't have as much knowledge/opinions about, but you don't feel great about the result or proposal. In those cases, it's often useful to take a pause and think about the exact area you might have disagreements about. Then open a dialogue with the person responsible, often times they will educate you on the context around the decision and you will feel much better about the result. Sometimes you may be able to bring a new perspective to the table and change the course of the decision.

## Confusion

- when you are not sure what was said or whether you heard correctly. Often the best time to raise questions is now. I've seen my shares of where someone was confused about a topic and feel like they missed their chance to ask a question. Don't be, if you are confused, put your hand up even if that person has moved on, and ask your question. Especially if you are in a large group, someone else might also be shy and appreciate your bravery.
  - Can you please clarify...
  - Didn't get the part about...
  - What do you mean by...
  - How would that work?
  - What's the next step?
  - So my understanding is that...

## Feedbacks

### Giving

- Give feedback early and often, don't just focus on the negatives, make sure you recognize both positive and negative behaviours
- Be very specific when you give feedback, include plenty of examples and evidence, especially if you are giving constructive feedback.
- Know that most feedback you are simply representing your own perspective. you very likely don't have complete information, you also don't know the motivation and context for why they did the things that they did. When giving constructive feedback, aim to

understand first and foremost. Try to get to an shared understanding of the whole situation and then take steps to address the feedback

## Receiving

- When receiving feedback, it's difficult but try to not take it personally. Understand that in most cases, the feedback giver wants you to succeed, and wants you to improve.
- First thing, acknowledge the feedback. Take a pause and process what you've heard, and then repeat what they said to make sure you understand correctly and are on the same page.
- Next, ask for clarifications, are there any part you didn't know about? Anything you are confused about? Anything that you feel like came out of the blue?
- Then own the feedback, act on suggestions for improvements, and take an active role in shaping the plan for improvement.

## Activity

### Retro - What has gone well so far?

- Did everyone do their work?
- Could you deploy what you wanted to deploy?
- Is your partner responsive, helpful, and reasonable?

### What has gone wrong?

- Did you have communication issues with a team member?
- Did you have challenges with your partner?

Take a few minute to answer, write them down if you have to.

## Activity: Scenarios

- Discuss each one of the scenarios with your group or another person
- Decide on good and bad actions to deal with each scenario
- Choose one scenario and practice communicating to deal with that scenario
- Choose a representative on behalf of the group to share with class

-