Feedback Skills For Software Developers Cheatsheet (prealpha-0.0.1)

Receving Positive Feedback

A compliment is a gift to be accepted. It is not a bomb to be defused, nor a volleyball to be returned.

Thank you.

Thanks. I worked hard on that.

I'm glad you liked it.

Thanks. I appreciate that you noticed.

Thanks. I feel good about it too.

Giving Helpful Positive Feedback

Refrain from the false complement. *I found that two thousand line method very easy to understand.*

Avoid the backhanded compliment. *Your code is so good that it doesn't need unit tests.*

Resist complementing future behavior. *You're really responsive and helpful. Could you please help me solve this problem.*

Complement behavior that has occurred. *It was really responsive of you to help me with that problem on short notice.*

Be specific. *I find your use of short, single responsibility methods makes the code very readable.*

Use shaping. Reward gradual approximations of the goal. *Well done. Your pull request increased overall test coverage from 27% to 30%.*

Receving Corrective Feedback

Relax. Open your mouth slightly. Soften your eyes. Open your hands. Relax your belly.

Avoid retaliation. *Well, it isn't like it's easy to read your forced use of recursion.*

Hold back. This is sometimes called 'elective blindness' or 'elective deafness'. In other words, ignore the nasty face or sharp tone.

Consider your safety. If violence is a risk, it is often not worth being assertive.

Don't demand perfection. Understand that most people simply aren't that skilled at giving feedback. Probe for value.

Validate their perceptions. *I can see how you might find this use of recursion awkward. A `while` loop might have been more natural.*

Validate their emotions. *This is important to you. You're really quite concerned about this.*

Agree in part. *You're right. Opinionated, automated formatting does have its downsides.*

Listen and wait. Let the critic voice their concern completely. Take the criticism *in* without taking it *on*.

Narrow and specify. *What's one downside of opinionated, automated formatting that comes to mind?*

Ask for clarification. *You mentioned a preference for 160 character lines. How important is that to you?*

Explain without offering excuses. *I used recursion here because I find it easier to reason about immutable code.*

Don't try to change their mind. *Agree to disagree. It's rare for a resolution to require full agreement.*

Thank the critic. We need feedback. Thanking them reminds us of this.

Respond to the style. *That was really helpful, specific feedback - thank you for speaking one-on-one.*

Ask for time. *Thank you for bringing up my persistence on problems. How about we meet next week, once I've had some time to consider what you said.*

Acknowledge. *Yes. I was half an hour late for work today. My car battery died.*

Cloud. Use for statements that have a grain of truth but are intended mostly as insults.

- Agree in part. *Yes. I do work more than 40 hours a week.*
- Agree in probability. *It could be that I work too much.*
- Agree in principle. *You're right. If I work too much, I will burn out.*

Probe. *What is it that bothers you about [important part of feedback]?*

Giving Corrective Feedback

Choose your timing.

Watch the ratio.

Think before talking.

Talk one-to-one.

Frame the issue.

Be precise.

Include the positive in the message.

Give information, not advice.

Don't emote.

Paterson, R. J. (2000). The assertiveness workbook: How to express your ideas and stand up for yourself at work and in relationships. Oakland, CA: New Harbinger Publications.

McKay, M., Davis, M., & Fanning, P. (2009). Messages: The Communication Skills Book. Oakland, CA: New Harbinger Publications.