

Receiving 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

Barbara Fredrickson

People are like an aperture.

Negative feedback has more weight for the receiver.

Many people have a secret sense of incompetence.

Desensitize people by using the 3:1 ratio.

This takes a concerted effort!

Focus within shooting distance.

Refrain from the false complement. *Wow. I love how you put so many responsibilities into that one class.*

Avoid the backhanded compliment. *Your test coverage is so much better, not nearly so lazy as before.*

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%.*

Receiving 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 Helpful Corrective Feedback

Choose your timing. Your feedback is more likely to help if the recipient can focus on your message.

Worse: after a long, hard work day.

Better: after lunch.

Watch the ratio. Pointing out what people are doing right is more powerful than pointing out what they are doing wrong. What do we want them to repeat?

Think before talking. It's easy to get off track. What do you want you to say? How do you want to say it?

Worse: *So, um, I wanted to talk about the code, you know, naming things is hard, and, do you know that quote...*

Better: *Our code base uses whole words for variable names, and your pull request uses mostly single-character variable names.*

Talk one-to-one. Worse: In front of the whole team over zoom: *Shaun, your pull request was full of unused private fields.*

Better: Via a private message over Slack: *Shaun, could I meet with you over Zoom for a moment?*

Frame the issue. This is important if what you say is likely to be threatening.

Worse: *I would like to talk about how long it took you to complete your last task...*

Better: *Your job here is secure. I would like to talk about how long it took...*

Be precise. People need to know exactly what to change.

Worse: *I didn't like your code*

Better: *Some files have different names than the classes they contained.*

Include the positive in the message. This opens the aperture - at least a bit.

Here is what I think you did really well... here is what I thought could have been better.

Q: How does this differ from the backhanded complement?

Give information, not advice. Where possible, let the other person decide what to do.

Worse: *Put up your fly. Convert the recursion to a while loop.*

Better: *Your fly is down. Many devs on our team find it easier to reason about loops than about recursion.*

Don't emote. If you're passionate about the topic, say so, but avoid acting out your passion.

Worse: *It pisses me off about this stupid use of recursion...*

Better: *I feel passionate about using loops instead of recursion, because...*

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.

Fredrickson, B. L. (2009). Positivity: Top-notch research reveals the 3-to-1 ratio that will change your life. New York: Three Rivers Press/Crown Publishers.