

Jonathan Dursi

# Calibrating Expectations:

## Giving frequent, useful, feedback



# Preface - Feedback is Expectation-Setting

When talking with new managers, I generally prefer to avoid the word “performance”, and instead talk about communicating expectations.

- It avoids unproductive framings like blaming the employee for “doing a poor job” or praising them for “doing a good job.”
- It takes the focus away from the past and what was or wasn’t done in that now-immutable time, and puts it on expectations of the future;
- It connects quite different things—immediate short-term task work and long-term goal setting—into a framework for talking about expectations;
- It provides a framework for accountability conversations between peers;
- It places the burden, properly, on managers for explicitly communicating expectations, and for having reasonable expectations in the first place;
- It correctly equates *not* giving feedback to withholding knowledge of your expectations from the team member.

Almost all of the team members you will ever work with *want* to meet your expectations, and are or can become capable of meeting them. Short-term feedback, or even long-term goal-setting and review, are about describing your expectations, and recognizing team members’ efforts as meeting or not meeting those expectations. The resulting conversations are about helping your team members meet your expectations in the future. In some cases—especially as you find yourself working in new areas—it will also partly be about recalibrating your expectations.

# Table of Contents

<b>I Bet You Don't Get Enough Feedback From Your Boss</b>	<b>5</b>
Neither Do Your Team Members	5
Feedback is How We Grow	5
Feedback Helps Us Rely On Each Other	6
Frequent Small Feedback » Few Big Conversations	6
<b>The Benefits of A Feedback Formula</b>	<b>8</b>
Permission, Observable, Expectation, Impact, Next Step	8
Some Feedback Examples	9
<b>Most Feedback Should Be Positive</b>	<b>11</b>
Aim for 80-90% Positive	11
Feedback Sandwiches Are Off The Menu	11
You May Be Surprised At The Effectiveness of Positive Feedback	12
<b>All Feedback Should Be About The Future</b>	<b>13</b>
The Past Is Gone. Feedback Incrementally Improves The Future.	13
This Means Letting Go Of Things That Happened In The Past	14
<b>Observable Behaviour or Output: Not Interpretations</b>	<b>15</b>
Focus on The Data You Have, Not Your Interpretation	15
Don't Give Feedback You Don't Have Supporting Data For	17
<b>Expectation: The Ruler Being Measured Against</b>	<b>18</b>
Consider the Underlying Expectation	18
Make It Explicit	19

Expectations (and Impacts) Can Be Less Observable	19
It's Fine For Expectations To Change Over Time	20
<b>Impact: The Why</b>	<b>21</b>
Future Impact Is What Matters	21
<b>Next Steps: Forward Looking Question</b>	<b>22</b>
Feedback Works To The Extent That It Improves The Future	22
Ask for More or Less of Something, or For Better Solutions	22
<b>Get Started</b>	<b>23</b>
Think About Your Expectations	23
Begin Noticing Behaviours and Outputs	23
Start With Positive Feedback	24
How to Give Feedback: Soon, Not Over Text, One-on-One	24
Slowly Introduce Negative Feedback	25
<b>Variants</b>	<b>26</b>
Debugging Discussion	26
Peer Feedback	27
<b>Resources</b>	<b>28</b>

# I Bet You Don't Get Enough Feedback From Your Boss

Some day I'll probably meet a new manager who feels that they get enough clear, helpful feedback from their boss. I'll hear that they routinely understand what's expected of them, know where they are and aren't meeting those expectations, and where they should focus on growing next.

It hasn't happened yet, though.

And it's not just us. You don't receive enough useful feedback, but the thing is...

## Neither Do Your Team Members

Your team members aren't significantly different from you. We all want to know where the bar set is so we can clear it; we want recognition when we do clear it; and when we miss, we want to know so we can improve.

We have data on this! Software developers, for example, feel more productive when they get regular feedback<sup>1</sup>.

## Feedback is How We Grow

Part of why we want lots of feedback is because that's how we develop our skills. Feedback lets us know where we're already doing well, and what we need to work on. Again, there's lots of data on this<sup>2</sup>; feedback helps us develop professionally.

Working in an environment where you're not getting feedback means knowing there's a bar there, somewhere, but not being able to see or feel it, and so not having any idea whether your jumps are high enough. It's disorienting and

---

<sup>1</sup> ["What Predicts Software Developers' Productivity?"](#), Murphy-Hill *et al.*, Transactions on Software Engineering (2019)

<sup>2</sup> See for instance ["Relationships between Feedback and Self-Development"](#), London *et al.*, Group and Org. Management (2016)

unsettling and unsatisfying. We don't know what's good enough and what we should be working harder on.

## **Feedback Helps Us Rely On Each Other**

What's more, none of us works alone. We work together as a team, and meet (or don't meet) goals as a team.

Teams rely on each other. That's what distinguishes a team from a random collection of people with similar email addresses. Relying on each other means being able to hold each other accountable to some shared set of expectations.

Communicating those expectations and how well we're doing against those expectations is the responsibility of every team member - but the responsibility begins with the team's manager. Teams with managers who model how to effectively deliver good, useful, helpful feedback tend to be teams that learn how to hold each other accountable.

## **Frequent Small Feedback » Few Big Conversations**

Because feedback is so important to our skills development, and to our team's being able to rely on each other, feedback is too important to "save up" for special occasions. We know a lot about what works well for feedback, and small, specific, timely, and frequent is much better than big, general, eventual, and occasional.

We don't learn skills by trying something new for a while, then waiting until something's gone tremendously wrong or right to assess our progress. We learn new skills – whether technical skills or skills working with a new team – by paying attention to lots of small signals.

That's how feedback works, too. We aim to give lots of small feedback about things done well and things that didn't meet expectations. This is much more effective, and much less stressful, than waiting until A Big Conversation is

needed to address a situation that's developed. In fact, having lots of small feedback conversations can often keep things on track enough that the need for Big Conversations are prevented.

Giving lots of little feedback takes practice, and may be uncomfortable to start. But small feedback is easier for you to give, and it's easier for the recipient to receive. It's more effective for the receiver since it's more frequent and more easily acted on. It's more effective for you since you get more practice giving it. The biggest part of the job of leading a team is keeping everyone pointed in the same direction, and it is much easier to give people small nudges about small things early on than to wait until something big happens.

I've heard "Oh, it's too small thing to bring up" from new managers more often than I can count. Good or bad, being a small thing makes it easier to talk about, not harder.

How would you feel if your boss was letting you fail to meet their expectation on something minor again and again, but simply decided to withhold that information from you because they just couldn't be bothered to have the conversation? If you wouldn't like that very much – and most of us wouldn't – then there's no justification for treating your team member that way either.

And if it's something small but positive – well, wouldn't you rather hear about that than not? I would.

In the next section I'll give you a simple templated formula you can use to give feedback. Having a ready-made model makes it much easier to structure effective feedback and give it often.

## The Benefits of A Feedback Formula

Having a specific formula for feedback helps both you as the feedback giver, and your team members as the feedback recipients.

For you, having a simple formula gives you guardrails to follow when you're getting used to giving feedback. It gives you a repeatable structure you can get very practiced at both constructing and delivering.

Having a regular format for the feedback you give also makes it easier for the receiver to hear and process the feedback. It can be uncomfortable hearing even positive feedback. Having it come in a familiar "shape" each time makes it that much easier to listen to.

Note that as you get more and more experienced giving feedback, you can riff off of the formula structure, modifying it for each particular case. But I'd suggest sticking to a formula consistently until it becomes second nature, first.

### Permission, Observable, Expectation, Impact, Next Step

The table below lays out the formula I've been using with managers I work with. The heart of any effective feedback model<sup>3</sup> is a focus on a specific, observable **behaviour** or output, and the **impact** of that behaviour or output. The best then take that and bookend it between questions.

I increasingly find it important to be explicit about the expectation that underlies the feedback; it helps the manager delivering the feedback be more explicit and consistent about the expectations they have, and it gives more guidance to the team member receiving the feedback, so they know what they're aiming for in the future.

---

<sup>3</sup> The blog post by Jacob Kaplan-Moss, "[Three Feedback Models](#)", compares three formulae - CCL's Situation-Behaviour-Impact, Manager-Tools Feedback Model, and Lara Hogan's Feedback Equation - all of which have been instrumental in shaping the one I use. Any of those are excellent models to follow and have been used widely to great success. Being consistent with any of them is more important than the benefits of any one in particular.



## The Feedback Formula

Part	Examples
<b>Permission to Start</b>	<p>Can I give you some feedback about...?</p> <p>Can we talk about the ....?</p> <p>When would be a good time to discuss how the ... went?</p>
<b>Observable: Behaviour, Output</b>	<p>When you... [specific observable thing they did]</p> <p>I noticed that ... [something observable and specific about their work product]</p>
<b>Expectation</b>	<p>As a team, we expect...</p> <p>Our aim is to...</p> <p>I expect..</p> <p>I want us to...</p>
<b>Impact</b>	<p>Otherwise, the team has to...</p> <p>The effect on the team if we don't meet that expectation is...</p>
<b>Next Steps (Forward-looking Question)</b>	<p>Can you ... next time?</p> <p>Is there anything I can do to help you .... in the future?</p> <p>Could you share the way that you succeeded in .... in the team wiki/next team meeting?</p>

## Some Feedback Examples

We'll go deeper into each point shortly, but let's start by looking at a few complete examples.

Can we talk about the training course for a second? The examples you chose were very relevant to the students taking this session. We aim with these courses to have them apply what they learn to their work; that showed them how to do so very effectively, and they're more likely to use the material as a result. Great job! Could you write a few words about choosing good problems on the teaching wiki?

Can I give you some feedback about the last sprint? It was a week before you submitted Aesha's code review. Our team aims for 2 day reviews.

The review held up Aesha's progress for half the sprint. Could you meet the 2-day time next sprint?

Can I give you some feedback on the project work? The tasks you had were due by the end of last week, but they're just getting finished now. Our team expectations are that we meet our commitments to each other on time, or in exceptional circumstances we let each other know ahead of time that there's a problem. We're going to be late getting the results back to the researchers. Can I count on you to meet the deadlines next time?

Can I talk to you about the meeting? You interrupted Carol three times in the planning discussion. On the team we make sure everyone gets a chance to express themselves. Team members are less likely to contribute if they feel they're going to be talked over. I want to keep hearing your input! But can you wait for other team members to finish before making your contributions?

Can I give you some feedback about the meeting? I've noticed how much less you've interrupted in the last few meetings. That lets everyone feel like their contributions will be heard and valued, and that's what we aim for letting everyone express themselves. Thanks so much! Is there anything I can do to help you keep up the good work?

In the following sections we'll talk more about the components of this feedback formula. But first, one important note about how much of each kind of feedback you should be giving ...

# Most Feedback Should Be Positive

## Aim for 80-90% Positive

That's not a typo. For every piece of corrective or negative feedback<sup>4</sup>, aim to give 4-9 pieces of positive or reinforcing feedback. Some weeks it may be more and some it may be less, but that's the goal.

There's basically no plausible amount of positive feedback you can give that's too much. And most of what your team members are doing is at least meeting expectations, right?

Remember, more small feedback is better than less big feedback. Positive feedback should not be reserved for some heroic act that goes above and beyond the call of duty. It can be for any specific thing that was done well, or even well enough. We want to encourage people to continue meeting expectations, and it feels nice to hear that your work was good, so by all means, tell people when something was good.

Certainly if you've given someone negative feedback about something, and they improve, give positive feedback about the improvement. But don't limit yourself to cases like that where there's some change you want to see. Getting in the habit of noticing specific things done well is a painless way to start giving feedback, it's good for your own perspective (as managers and leads we tend to focus too easily on the negative), and it's good for your team members to hear about so they know to keep up the good work.

## Feedback Sandwiches Are Off The Menu

People *hate* the "feedback sandwich"<sup>5</sup>, where you "wrap" negative feedback in positive feedback. It's seen, reasonably enough, as being insincere, and more

---

<sup>4</sup> This battle is long lost, I know, but I don't really like the term "constructive feedback". All feedback should be constructive!

<sup>5</sup> For data on this and what works, see for instance "[How do performance feedback characteristics influence recipients' reactions? A state-of-the-art review on feedback source, timing, and valence effects](#)", Jonas Lechermeier & Martin Fassnacht (2018) and references therein.

about the comfort of the person giving the feedback than about utility to the person receiving it.

Yes, for negative feedback to land well and be well received, it has to be seen as being fair. One way to achieve that is balancing it with positive feedback. But you can do that by just... giving positive feedback. When you're frequently telling a team member what they've done well, they're much more likely to take a piece of negative feedback as it's intended: something to work on as suggested by a manager who notices when they've done well, too.

## **You May Be Surprised At The Effectiveness of Positive Feedback**

People *want* to do well. Telling them specific things that they've done that meet or exceed expectations gives them a tool they need to continue doing well. And it gives them a bit of a motivation and confidence boost, too.

If you're reading this, there's probably a team member who's being less than completely successful, and you want to have some tough feedback conversations with them. Fair enough, and so you should. But give *them* positive feedback too, about things they *are* doing adequately well. I can't promise it will turn them around, but you will very likely see some kind of result.

Personally, once I've received an "attaboy" for some effort, the bar has been raised and I will make quite sure that I'm doing at least that well in following efforts. You probably feel the same. So do most of your team members. Positive feedback is very effective. We want to do well.

# All Feedback Should Be About The Future

## The Past Is Gone. Feedback Incrementally Improves The Future.

Changing the past is impossible. Incrementally improving the future is actually pretty easy.

Feedback is **not** about what happened. It's triggered by something that happened, but it's not fundamentally *about* that event. It's *about* what *can* happen next time. The only purpose of feedback is to incrementally improve the future, by encouraging behaviour and outputs that meet your and the teams' expectations.

So good feedback is more than just praise or blame. Talking about something that went poorly without discussing a diagnosis and a path forward is complaining, not giving feedback. "Good job" is at least nice to hear, but without specifics about what was good and why so that the good work can be repeated, it's just cotton candy – sweet, nutrient-free fluff.

Feedback matters to the extent that it improves the future. And you can do several things to make it more likely that it *will* improve the future; the feedback formula template will help. The keys are:

- Sharing specific, actionable details – the feedback should be specific enough, and grounded in observed behaviour or output, that people know exactly what's being discussed and what to keep doing or change
- Give the expectations – explain what team expectation is relevant in this case, so they know they bar that's being measured against
- Describe the impact – explain the goal of the expectation, and why it matters
- Ask for next steps – keep the discussion grounded in the future by asking about what happens next.

## This Means Letting Go Of Things That Happened In The Past

Feedback is about affecting the future. That means it's about discussing something that's happened very recently as a way of reinforcing or changing that behaviour for the next time. You can't address the future by dredging up the past.

I'd encourage you to read this and start applying it immediately, to new events, as they occur, starting from scratch. Bringing up something that happened months ago will feel like an ambush, and won't work as feedback. If there's been a pattern of bad behaviour in the past, and it recurs, then you can address *the new instance* with some feedback.

That doesn't mean you can't take action based on something that happened some time ago ago, if it's serious enough to warrant it. Maybe that means a discussion (perhaps the debugging discussion below), maybe that means something else. But we're now talking about something different than feedback.

# Observable Behaviour or Output: Not Interpretations

## Focus on The Data You Have, Not Your Interpretation

You and I are experts in our respective fields.

But we don't have the ability to peer into the state of mind or intentions of our team members. Any interpretation we have of what their actions mean is *our* interpretation. It's not much more than a guess.

And really – it doesn't matter, right? What we care about is external, not internal. We care about future behaviour and output, and the impact of that behaviour and work on the team.

Not Observable	Observable
Disrespectful	Interrupted three times Dismissed idea immediately without hearing the case for it Said "it's very technical, they wouldn't understand" Arrived 10 minutes late for meeting
Collaborative	Sought input at multiple stages of the project Incorporated their concerns about .... into the result Kept ... informed of the work's progress
Sloppy	The work that was done was incomplete and didn't meet all the requirements The deliverable was completed well behind schedule Several mistakes were found
Conscientious	Helpful tests were committed with the code The amount of detail in the talk/blog post... The follow-up with the client... Identifying a data problem early on in the process

So don't be tempted to give feedback on *your interpretations* concerning a behaviour or output. If what you see is incomplete work, focus on that, don't

jump to “you’re not taking this project seriously”. Some examples of what I mean are given in the table.

Focus completely on the actual specific behaviour, or an objective aspect of their work output. Focus on the data, not your interpretation of the data.

First, the actual observable data is relatively objective, and your interpretation is not. Was showing up late for the meeting an act of disrespect? Well, maybe, but maybe not. After all, only the team member (if even them) really knows what they were feeling at the time. Focus on the behaviour you want to reinforce or change. It’s not arguable that they were ten minutes late, and if expectations are that we show up for team meetings on time, that’s enough to give feedback about.

Second, the interpretations are almost always vaguer than the observed behaviour or output, and so harder to do something about. “Keep being conscientious”? Ok, sure, but how specifically? About everything? “Don’t be so sloppy”? Same deal - how? When?

Finally, interpretations are often charged and value laden, and get understood as statements about the person rather than the behaviour or output. For negative feedback, that leads (reasonably enough) to defensiveness, hurt feelings, resistance, and arguing. And even for positive feedback, it’s just not helpful.

The more specific and objective what you’re giving feedback about is, the more likely it is to have the desired outcome of incrementally improving the future. Be laser-focussed on specific, observable, changeable things to give feedback on.

My experience is that this is the piece of any feedback model that new managers have the hardest time with initially. It’s just so easy to jump to our interpretation of what happened. But it doesn’t take very much practice to get quite good at focusing on the observables. We are people of science, after all,



and we know how to objectively collect and observe data. That's better for feedback, and for your own perspective.

## **Don't Give Feedback You Don't Have Supporting Data For**

At some point you'll be asked to give feedback on behalf of someone else. A team member or client has an issue that they don't feel comfortable raising themselves, so they tell you, in the hopes that you will pass it on.

Giving this hand-me-down feedback feels like the right thing to do at the moment. It's not; it goes poorly. (Ask me how I know!) Feedback lands well when it's trusted. If it starts to feel like gossip or third hand, not only will that feedback not have its intended effect, future feedback will be less trusted, too.

That doesn't mean you can't do anything. If a recurring behaviour has been pointed out, you can make a point of looking for it so you can give first-hand feedback. Or sometimes the state of the relationship with the third party can be the observable output itself, and you can give feedback about that. Or you can encourage the other person to provide the feedback, *if* that's appropriate to the situation.

You can also discuss the topic on its own merit. Just don't phrase it as feedback if it's not something you yourself have observed. Have a less structured conversation about it.

## Expectation: The Ruler Being Measured Against

### Consider the Underlying Expectation

Some teams have pretty well-established internal expectations of how they work together; otherwise, there are always fairly basic standards of professionalism about quality or timeliness or courtesy that might be surpassed or violated.

Either way, an item of behaviour or a piece of work output caught your attention because it exceeded or failed to meet some expectation of yours or the teams.

It's useful as a new manager to put a some time into thinking about those expectations. It will help you be more consistent about giving feedback, applying a consistent set of expectations to behaviour in the team.

If you're constantly responding only to individual events, it's easy to wonder if *this* occurrence really warrants some mention. For instance, I've seen managers who were ambivalent about giving feedback about (say) work being handed off late – “well, but this time it genuinely was a bit of a harder task” – but when there is a team expectation that “we deliver work to each other on time, and if we can't we let each other know well ahead of schedule,” it's much easier to know to give that feedback.

Observable	Expectation
Interrupted three times	We listen to what each other has to say
Incorporated their concerns about .... into the result Kept ... informed of the work's progress The follow-up with the client...	We take the time and effort to communicate with our team, clients, and stakeholders
Arrived 10 minutes late for meeting	We arrive on-time to team meetings

Observable	Expectation
The work that was done was incomplete and didn't meet all the requirements  The deliverable was completed well behind schedule  Several mistakes were found  Helpful tests were committed with the code	We deliver quality work to each other, and do so on time

## Make It Explicit

If having expectations clear in your own mind helps you, it's even more helpful for the team members.

The expectation underlying some piece of feedback may be so obvious to you that it doesn't even seem worth mentioning. But different teams doing different work value different things. Team members don't join your team automatically knowing what's important there.

Clarity of expectations is too rare, and incredibly valuable. Letting people know what's expected of them reduces uncertainty and worry. It lets team members know what matters and what they should be focussed on. It can help to foster a sense of collaboration and trust, as team members know what to expect of each other. It helps set a consistent team culture.

## Expectations (and Impacts) Can Be Less Observable

When we talked about the behaviour or output that inspired the feedback, I stressed the importance of it being objective, and observable. Implying to someone that you know what they were feeling or intended by something (a) may be wrong, and (b) will be interpreted as a comment on them as a person. It leads to defensiveness and argument, and so derails otherwise useful conversations.

But when you're talking to someone about the expectations of their behaviour or output *for other people*, or impacts of their behaviour or output on *other*

*people*, we don't always have to be so rigorous. Objective is still a little better, as it sets an unambiguous bar for the team member to meet or exceed; but we can also have expectations like "we ensure our team members feel respected" or "our clients feel heard and valued."

"Stop being disrespectful at team meetings" lands *very* differently than "Frequently interrupting team members can make them feel disrespected." The first will get someone's back up, the second generally won't.

## **It's Fine For Expectations To Change Over Time**

The value of expectations is that they can be consistently applied, but that doesn't mean they're carved in stone.

You expectations may change. The team, discussing their own expectations, can evolve them over time. As a team grows, or its work or priorities change, or just as the team finds new norms<sup>6</sup> for working together, expectations can shift. That's a good and healthy development. It does mean that what you give feedback on will have to change, too, to support the new expectations.

---

<sup>6</sup> Tucker's team stages, "Forming, Storming, Norming, and Performing" (e.g. [https://en.wikipedia.org/wiki/Tuckman's\\_stages\\_of\\_group\\_development](https://en.wikipedia.org/wiki/Tuckman's_stages_of_group_development)) are no longer understood to be a linear progression of development, but they are still a useful way to think about how teams learn to work together.

## Impact: The Why

### Future Impact Is What Matters

We care about the behaviour or aspect of output that's triggering this feedback because we care about the future impact on the team, its work, and its clients.

Observable	Expectation	Impact
Interrupted three times	We listen to what each other has to say	Otherwise, people will contribute less, and we'll lose out on their insights when making decisions.
Incorporated their concerns about .... into the result  Kept ... informed of the work's progress  The follow-up with the client...	We take the time and effort to communicate with our team, clients, and stakeholders	Our client feels valued and has more confidence in the work we did for them.
Arrived 10 minutes late for meeting	We arrive on-time to team meetings	Otherwise we have to repeat ourselves and it's harder to end meetings on time.
The work that was done was incomplete and didn't meet all the requirements  The deliverable was completed well behind schedule	We deliver quality work to each other, and do so on time	Otherwise the team may not be able to meet our commitments to clients.

When we were individual contributors, in the weeds and focussed on our own work, it was sometimes hard to see how what we did or didn't do mattered in the big picture. One of the jobs of a manager or lead is to connect those dots for team members, so they understand the impact of their work and how they do it.

Connecting their behaviours and outputs to impact on the team and its work is what will make the feedback "stick" for your team members. It's a "why" for the feedback that genuinely matters.

## **Next Steps: Forward Looking Question**

### **Feedback Works To The Extent That It Improves The Future**

Our feedback was inspired or triggered by some past event, but it's *about* the future.

So the final step in the feedback formula is really the key, the whole purpose, of the feedback. We're asking for (continued) good outcomes *in the future*.

### **Ask for More or Less of Something, or For Better Solutions**

We close out the feedback formula asking for more or less of some behaviour or output, or for the team member to propose better solutions or approaches for the future, or to build on some success.

This will frequently be something as simple as "Can you make being on time a priority in the future?", or "Can you deliver on time next time?".

If the team member didn't meet an expectation, you can ask them to come up with a way to meet the expectation next time, for discussion later. "Can you come up with a way that will work better next time? We can discuss it in our next one on one.", for instance. Or "What steps can you take to meet this expectation in the future?"

Asking what you as a manager can do to help will work very well with some team members; "What can I do to help you to keep up the good work?" or "... to meet this expectation in the future?"

## Get Started

I'd encourage you to start right away by thinking about your expectations and noticing behaviours, and then as soon as you're comfortable, begin by giving positive feedback.

## Think About Your Expectations

If you're like most managers, you haven't yet put a lot of thought into what your detailed expectations are for team behaviour. This is a great chance to find concise ways of expressing those expectations that you probably haven't explicitly talked about before.

This exploration of expectations doesn't have to be comprehensive, or even especially deep at first. As you notice behaviours and outputs (below) you'll probably notice pretty quickly that a fairly small number of expectations are implicated in the majority of the behaviours or outputs that might inspire feedback pretty soon. That's great; now start thinking of succinct ways to express that expectation that would resonate with your team. What are impacts of meeting or not meeting those expectations?

This isn't (yet) the time to start coming up with new expectations with would require big changes in the team's behaviour. To start, focus on expectations that are often or usually met, and that you'd like to see met or exceeded more consistently.

## Begin Noticing Behaviours and Outputs

While thinking about expectations, start paying very close attention to behaviours or aspects of outputs that you want to see more or less of. We're *not* going to give feedback on these at this point! We're just training our senses of workplace observation.

It takes a little (but not a lot) of practice to get good at describing behaviours and outputs in very objective, observable terms. Keep doing this until you feel

pretty comfortable quickly identifying extremely specific behaviours or work outputs that you want to encourage more and less of.

And make particularly sure that you're noticing lots of specific *positive* behaviours and outputs. We're managers or leads; we feel the weight of responsibility, and because of that, noticing problems comes more easily for many of us than noticing things going well. Develop those positive-behaviour-detecting muscles, too.

## **Start With Positive Feedback**

Once you're comfortable with describing some of your fundamental expectations, identifying specific observables that meet or exceed those, and the impact of those observables, start giving *positive* (and only positive) feedback.

You're going to be giving lots of positive feedback anyway, so practicing it first is useful. Also, the positive feedback conversations are just more robust to minor trip-ups. You won't be super fluent with the feedback formula yet; it's good to start with conversations that are almost certainly going to go well.

If it would make you more comfortable before starting a new kind of conversation with your team members, let them know at an upcoming team meeting that you'll be starting this practice. It doesn't have to be a big deal; your team members want feedback. Tell them that you're pleased with how the team is doing, but you've realized that as a manager you're not giving as much feedback you should. You're going to start giving small feedback more frequently, most of it positive.

## **How to Give Feedback: Soon, Not Over Text, One-on-One**

Feedback works best when it's timely. Give feedback very soon after the event that's triggering it; within a week is a good rule. The sooner after the event, the easier it is to reinforce or change the underlying behaviour.



Give the feedback “face-to-face”; if that means over a video or audio call for your team, that’s fine. But don’t do it over email or Slack or other text media, especially when you’re getting started. You lose too much communication context via text. If it works well in your situation, you can start the feedback discussion by doing the “permission to start” step over text, but the rest should be in person or over a call.

When you’ve built a good feedback relationship with your teammates – when they’re used to hearing feedback from you, and you’re used to giving it – then Slack or equivalent can work for the occasional minor positive feedback. But in general, these conversations benefit from hearing each other’s tone of voice and seeing each other’s expressions.

Have these feedback discussions one-on-one. Public positive feedback *can* work, but a lot of people are uncomfortable receiving positive feedback in public. And you don’t want to make too big a distinction between positive and negative feedback, delivering them very differently. Once everyone’s used to feedback, and if you know (preferably by asking) that a team member is comfortable receiving positive feedback in public, you can give the occasional positive feedback in public. That can particularly make sense if the feedback is to a number of people for a team effort. But one-on-one should be the default.

Speaking of one-on-one, regularly scheduled one-on-one meetings are an ideal venue for providing feedback as well as other professional development discussions. I highly recommend them.

## **Slowly Introduce Negative Feedback**

When you’re feeling comfortable giving positive feedback, and your team members are getting comfortable hearing it, it’s time to start introducing negative feedback.

Don’t overdo it! You might be tempted to now deliver a bunch of negative feedback that you’ve “saved up” over the past weeks. That won’t work. You’re

still going to be mostly giving positive feedback, with targeted negative feedback inspired by in-the-past-week events here and there.

You'll likely feel uncomfortable the first few times you give negative feedback. You can help yourself by giving your first few pieces of negative feedback to team members you have good relationships with, who are probably your higher performing team members. They want negative feedback! They want to get better and build their skills.

Don't feel awkward about these initial discussions. It is good and proper to discuss expectations for team members, and the impact of their work on the team and the clients. Keeping the feedback small and focussed on specific observables will make things easier.

## **Variants**

We've mentioned that the feedback formula isn't the right structure for all conversations about expectations and behaviour. The components of the formula, however, are widely applicable.

## **Debugging Discussion**

Sometimes the discussion you want to have is a bit more of a debugging session. Something happened that didn't go the way you expected, you're not sure why, and you think digging into what happened is important.

In that case, you can use the structure of the feedback formula but make the final question more of an introduction to a longer discussion. Lara Hogan's question, "What are you optimizing for?"<sup>7</sup> is a good example of an open-ended question that can start a discussion to find out what's going on.

In these discussions impact may or may not be relevant, but the opening and observations will still be key. The expectation you discuss will be a little different than a team expectation, but the idea is still the same.

---

<sup>7</sup> <https://larahogan.me/blog/what-are-you-optimizing-for/>

## Peer Feedback

It's perfectly reasonable to have expectations of your peers, or even stakeholders or clients. Certainly their actions will have impact on you and your team.

Your relationship with them is different than that with your team members, though. You can't necessarily ask for behaviour changes directly.

But you absolutely can use the other parts of the feedback formula to talk about behaviour and impact, and what you were expecting.

## Resources

Here are my current favourite resources for managers giving feedback:

- [The Leader Lab Book](#) by Tania Luna and LeeAnn Renninger PhD is a casual and well-thought out introduction to some foundational people-management skills, and how to combine them in different combinations. And the [Leader Labs model](#) is similar to what's described here.
- [Rands Leadership Slack](#) is a 20,000+ strong community of people who are interested in management, with a heavy (but not exclusive) tech focus. There are always very interesting discussions going on from which you can learn a lot, on any topic. There's a [#research-computing-and-data](#) channel.
- I quite like the [Manager-Tools Basics Podcasts](#), and their [book](#), although I don't agree with some of the material in the later podcasts.
- Lara Hogan has some key advice - [don't soften tough feedback](#), and [tough love for managers giving feedback](#).
- Men, we particularly tend to soften feedback to women and other groups not represented well in our teams; [this hurts their career growth](#).
- My 10-minute talk [Help, I'm a Research Software Manager!](#) touches on feedback in a broader discussion of management responsibilities.
- Finally, if the was helpful, you might find my [Research Computing Teams Newsletter](#) valuable. And you can always [email me](#) or [arrange a quick coaching call](#) if you have questions.



Research Computing Teams is a newsletter covering the work of managing a research technology team - updates on managing individuals and teams, emerging technology, product management, and working with research communities. Find out more at [www.researchcomputingteams.org](http://www.researchcomputingteams.org).