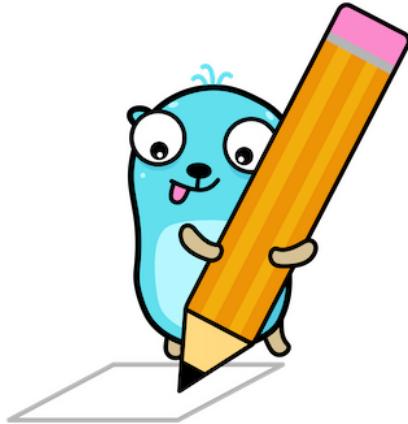


# 5. Work

*The only thing harder than trying to design software is trying to design software as a team.*

—Joel Spolsky, “[The Big Picture](#)”



Welcome aboard! You've just started a new job, maybe even your first job. What should you expect? How should you act? What do you need to know to succeed? In this chapter, let's explore the world of work.

## Starting out

You might be feeling nervous about the prospect of starting a new job. If so, don't worry, you're not alone. There are lots of things to worry about, as with any major life change, but the chief worry in most people's minds is probably:

### Can I even do this?

The fact that you're feeling this way is a good sign, oddly enough. If you're going to make progress in your career, then each new job should at least *seem* slightly beyond your current ability.

This might seem obvious, but it's something many people fail to take into account. When they look for jobs, they're looking for jobs they *can already do*. In other words,

jobs where the requirements more or less match their current set of skills.

## Impostor syndrome

So *impostor syndrome*—the feeling that you’re not really up to the job and sooner or later people will find out—isn’t necessarily a problem. In fact, I’d say it’s a sign that you’re taking on the right kind of challenges. You should be concerned if you *don’t* feel a bit of impostor syndrome in any new job.

Also, everybody else feels just the same as you, however experienced or senior they are, and however relaxed and confident they might look on the outside. Everyone is secretly thinking “I hope nobody finds out I have absolutely no idea what I’m doing!”

## You can do this

Having evaluated you carefully, and probably at considerable length, the company decided you were the right person for this job. Getting that decision wrong would be unfortunate and expensive, and companies that consistently make bad hires don’t survive for long.

Now, I’m not saying you should be *over*-confident; that would be silly. What I *am* saying is that you should regard a modest amount of impostor syndrome as normal, and that while it’ll never go away altogether, it’s also not a sign that anything’s wrong or that you shouldn’t be doing what you’re doing. Instead, just keep in mind that this is what *growth* feels like.

*And then I thought to myself, “You know, what they think of you is so fantastic, it’s impossible to live up to it. You have no responsibility to live up to it!”. I am what I am, and if they expected me to be good and they’re offering me some money for it, it’s their hard luck.*

—Richard Feynman, “Surely You’re Joking, Mr. Feynman!”

## Don’t show off

In the first days or weeks of the new job, you may feel a strong impulse to *demonstrate* to everyone around you just how skilled you are and how much you know. This often takes the form of talking a lot in meetings, making points to try and sound smart. Or, worse, pointing out what’s wrong with everybody else’s code and telling them how it could be improved.

This usually doesn’t go well. When someone in the team is constantly showing off and acting like a know-it-all, we don’t tend to assume they must be over-compensating for hidden feelings of inadequacy. We just assume they’re a deeply annoying person.

It can also be very exciting to join a team of smart people who know a lot, and sometimes this excitement can overwhelm you. If you’ve been missing some intellectual stimulation and the company of interesting people, you may find that your new position generates a lot of fresh ideas. That’s great! You don’t need to tell them to everybody

all at once, though. Capture them in a notebook and revisit them after a month or two to see if they're still as interesting as you thought.

## Listen before you speak

Spend your first few weeks of any new job or role in *listening* mode. Absorb as much information as you can about how things work and who does what. If you see something that you think isn't optimal and could be fixed, keep it to yourself—for now.

Think about how irritating it is when you tell a friend about some problem you're having, and they say “Why don't you just...?” Well, you probably already thought of that, and it doesn't work. But it's annoying that they assume you hadn't even looked at the most basic and obvious solutions.

So don't be *that* person. Instead, assume that if something seems off to you, it's probably because there's some important factor you're not aware of yet. Try to learn more about the situation and the problem. Talk to people who've worked on it. Ask questions like “Can you help me understand why...?”

## How to make suggestions

Once you've spent a decent chunk of time in listening and learning mode, and if you *still* think you have a good idea that someone else hasn't already tried, you can start gently and diplomatically introducing it. For example, you could say “I was thinking that maybe it would be good to do X. Does that sound reasonable to you, or is there some issue I don't know about?”

If the first two or three people you ask can't instantly find a reason to reject the idea, then maybe you're on to something. You could ask them to help you try it out. Note the words “help” and “try out”. You don't want to come across as saying “Hey, losers, I fixed your code for you. Yeah, you're welcome,” even if that's how you really feel. There are ways and ways of doing things.

## Settling in

If it's natural to feel apprehensive before starting a new job, it's even more natural to feel a bit overwhelmed when you do actually start work.

## Coping with “Day 1 Panic”

Don't panic. You *will* cope. Just look around you: all your co-workers must have felt exactly the same when they started, and now they're doing fine. They probably don't even look stressed. It's amazing how quickly we can adapt to new circumstances and start to thrive.

Spaaaaace engineers talk about *max q*, the point of peak dynamic pressure experienced during a rocket launch:

$$q = \frac{1}{2} \rho v^2$$

At ground level, the atmosphere is thick, but the rocket is moving slowly. Higher up, the density of the air decreases, but the rocket is also moving through it faster. So there's a point fairly early in the flight where the speed and density curves intersect, and the rocket experiences its maximum structural load:  $\max q$ .

If the vehicle is going to fail under stress, this is when it's likely to happen, and you may feel like you're going through this phase too. How do rocket scientists deal with this problem? They throttle back the engines slightly, to reduce the loads on the structure. You can do the same.

In other words, throttle back your expectations of yourself slightly. Don't assume you should be in a position to start delivering value right away, or even within a few weeks. It takes time to learn your way around a complex codebase (and is there any other kind?)

## **The trial period**

There may be a formal trial period, or probation, during which it may be easier for the employer to dispense with your services if it turns out you're not a good fit. This can add to the stress of feeling like you have to prove yourself, because in this case you do.

One way to manage it, and to improve your chances of making it through the probation filter, is to talk to your manager and have them clarify exactly what's expected of you. Will you be evaluated on specific things, or is it more of a general feeling? If people have washed out in the past during the trial period, why? What can you do to avoid the same result?

Check in regularly with your manager during the trial period to see how you're doing, and whether you need to adjust course. Don't wait for them to tell you.

Instead, be pro-active, and get regular feedback about your performance. You'll learn a lot from these conversations about the clarity, transparency, and consistency of your new employer's communications with its staff.

## **Expect to be slow at first**

When everything is new, and nothing is routine, even the simplest-seeming tasks can be a struggle. Need to make a one-line change to some documentation page? There's actually a tool you need for that. How to install it? Well, the instructions are out of date and don't work, so you'll have to find someone who knows and ask them. But they're too busy to help you right now. And so on.

“This is ridiculous! It should only take five minutes,” you may be saying to yourself. Sure: when you have all the tools you need, you know how to use them, and you're ex-

perienced enough to deal with any little glitches, it's a five-minute job. But when none of those things are true, it could easily take you all day.

So don't be too tough on yourself, and don't assume that others will be feeling impatient if you're not reaching orbital velocity just yet.

In fact, your experiences as a new person are very valuable input. Things like documentation *get out of date* because people become familiar with the processes and don't need the docs anymore. So any time you find something like this, offer to update it (with the help of someone who can explain what it *should* say).

## This is what learning feels like

Don't wait till you get stuck before asking for help. Get help so that you *don't* get stuck. If there's no help available, and you're struggling, see if you can break down the problem into smaller steps. Focusing on one small thing at a time helps reduce anxiety, and every tiny task you can check off your list increases your feeling that you're getting somewhere.

You *will* get frustrated at first, and demotivated, and maybe even depressed. But this will pass. You will feel dumb for not being able to master things right away. But this is completely normal: indeed, that's what it *feels* like to learn something new.

When you lift weights at the gym, your muscles will be sore; that's what it feels like to get stronger. When you lift weights with your brain, it feels sore, too. But little by little, it is getting stronger. And, just as your body needs rest and downtime between workouts, to recover and grow new muscle tissue, your brain needs that, too. Go easy on it, and on yourself.

## Being effective

So take heart, you *will* survive that scary "max *q*" period, where your brain feels like it's exploding every five minutes. You may feel convinced that you're going to get fired for rank incompetence at any moment, but that won't happen. At least, let's hope not.

BOSS: *Kramer, I've been reviewing your work. Quite frankly, it stinks. These reports you handed in... it's almost as if you have no business training at all.*

KRAMER: *Well, I'm just trying to get ahead.*

BOSS: *Well, I'm sorry. There's just no way that we could keep you on.*

KRAMER: *But I don't even really work here.*

BOSS: *That's what makes this so difficult.*

—Seinfeld, “The Bizarro Jerry”

How long does it take? There's no specific moment when you realise you're no longer a newbie, and you start to feel like you're getting a handle on things. But there'll come a time when you look back on yourself from a few weeks ago, or a few months ago, and think "What was I even worrying about? This is fine!"

## Social capital

A useful thing to understand about work, or any kind of group activity, is that you have a certain amount of *social capital* to play with. Being helpful and kind increases your social bank balance, if you like. Being honest, trustworthy, reliable, humble, and easy-going increases it, too. On the other hand, being arrogant or unpleasant will drain your capital rapidly.

And it's not just about being a good person, though you should certainly care about that. You *need* social capital to do your job. When you've got a problem you can't solve and you're under pressure to get it fixed right now, you can draw on your capital by asking a co-worker for help. Needless to say, if you haven't helped them in the past, they won't feel much like helping you now.

*A team is not a group of people that work together. A team is a group of people that trust each other.*

—Simon Sinek

Social capital is what makes people listen to your ideas, or give weight to your opinions. It's how people see you as a team player. It's what makes them want to help you, even when that's not strictly their job. It's the currency of trust. And trust is what makes the whole thing work.

## Managing your commitments

To be trustworthy, and earn social capital, it's important to prioritise your *commitments*. In other words, if you say you'll do something, you need to do it.

If you don't, people will gradually stop trusting you, and your social capital will dwindle away. The most dangerous thing about *that* is that you may not even know it's happening.

*People don't tell you when they stop trusting you.*

—Jerry Weinberg, “Secrets of Consulting”

The biggest threat to your trustworthiness is probably not that you're a horrible, deceitful person who tells lies. It's that a perfectly natural desire to please causes you to *over-promise* and *under-deliver*.

We all want to say “yes” to things. It makes everyone happy, until you realise you've taken on far too much, and you have to belatedly tell them you won't be able to do the thing after all. Now they'll be *unhappy*, not just because you made a promise and then let them down, but because it's probably now too late for them to find someone else to help.

## Saying “no” is a superpower

While ill-considered promises might earn you a little temporary capital, your failure to deliver will cost you more later, resulting in a net deficit. On the other hand, say-

ing “no” now might cost you a *tiny* bit of capital in the short term, but in the long term you’ll gain much more, because you’ll be known as a good commitment manager.

When someone asks you for a commitment, they don’t *really* want you to say yes unless you’re actually going to deliver on it. So think carefully about your answer. It’s fine to say “Let me get back to you when I’ve checked my schedule.”

And, if you eventually conclude that this *isn’t* something you can be 95% confident about delivering on time, you should say “I’d really like to help, but it just doesn’t look like it’s going to be possible to fit this in with my existing commitments.”

That’s a nice way to say no, because it underlines the value you *place* on your commitments, and how seriously you take them. The person you’re talking to knows that, even if you can’t help this time, when you *do* say you can, you will.

## Finding your feet

One thing that’s guaranteed when you join any new organisation, division, or even team: they’ll do things differently from what you’re used to. There will be unfamiliar and unexpected rules, some written, most unwritten, and you will break them without even meaning to.

That’s okay. If you listen more than you talk, and pay close attention to what everyone else is doing—and even more importantly, *how* they’re doing it—you won’t get yourself into too much trouble.

But there will be friction at some point and about some things, so how do you deal with that?

Fitting in with the way others work is hard, and it takes some time and patience. You have to be sensitive to what’s going on, and tolerant of things that aren’t the way you like them to be.

*A new kid in school, got to follow the rule, you got to learn the routine...*

*You got to fill out a form first, and then you wait in the line*

*It's all his design, no one cuts in the line, no one here likes a sneak*

*You got to fill out a form first, and then you wait in the line*

—Paul Simon, “[The Afterlife](#)”

When you go from working largely by yourself to working in a team, especially a big team, you’re likely to encounter lots of things that you find hard to understand, or even laughably wrong.

Coding standards, patterns, and practices vary widely: even the layout of brackets and whitespace can be a point of debate, in languages that don’t have a canonical standard format.

*gofmt's style is no one's favourite, yet gofmt is everyone's favourite.*

—“[Go Proverbs](#)”

All teams have a house style that you need to learn and abide by. You may not like or agree with all its choices, but then, your fellow team members probably feel the same. The point of standardising practices like this is not to do everything perfectly, but to stop people wasting time arguing about trivialities.

## **Don't rearrange the furniture**

A codebase is a little like someone's house: they've probably lived there for a while, and they have things mostly the way they like them. Also, they're used to the arrangements. It can be difficult to have someone new come in and start blundering around the place, falling over the furniture or breaking the crockery.

And you wouldn't be too pleased if your new roommate or partner started taking issue with your choice of decor or furnishings. You certainly wouldn't expect them to just start rearranging things to suit themselves: "I'm used to having the couch face the window—I was sure you wouldn't mind me moving it."

You'd expect them to adapt themselves to your preferences, even if it's not what *they're* used to. Now imagine those roles reversed. *You're* the guest in someone else's codebase or project. Don't complain, criticise, question, or move things around without asking. Things are the way they are for a reason, even if that reason is simply "we haven't got around to fixing that yet".

A little empathy goes a long way here. If something seems wrong to you, it might just be the way it is because other people have thought about it a little more carefully than you have. Or maybe they're fighting a desperate battle that you know nothing about, and compared with that, this issue seems ridiculously trivial. And, if something seems a little half-finished, maybe that's because it is.

## **Making the right first impression**

Don't look for problems your colleagues seem to have missed. Instead, ask *them* what problems they could really use help with. You could say "Do you have any little tasks kicking around that no one's had time to get to lately? Tackling those would help me get familiar with the codebase, while still contributing something useful." This is much more likely to get a favourable, or even rapturous, reception.

It sends out a signal straight away that you're not here to make them look bad. In fact, you're here to make them look good. You're not going to create extra work for them: instead, you're going to *lighten* their load. Once you've established this idea, you'll find your welcome a lot warmer.

## **Collaboration is your biggest challenge**

The biggest challenges you'll face at work won't be build failures or the type system, they'll be about trying to work with other people. If everyone were smart, nice, and competent, things would be easier, but you'll have to work with plenty of the other kind too.

Even when team members trust each other and get along, there will be conflicts and difficulties. Collaboration is just *hard*. This is known. And no one is a born collaborator; it's something you have to learn and practice to get good at. We're all somewhere along that journey, and throughout your career you'll meet people at every level of collaborative ability, from "zero" to "decent".

## People skills

While technical skills predominate in the job application process, once you actually make it into the workplace (whether physical or virtual), you'll find that your tech chops are taken for granted. What's going to determine your ultimate success or failure in this job is your *people* skills.

Unfortunately, not everyone realises this, and you'll find a few people in every team or company who, while they may be super smart and experienced, are also anti-social jerks. Sorry about that.

You can do your little bit to redress this problem by learning and demonstrating how people *should* treat one another, at work or elsewhere: with respect, consideration, and kindness. Not everyone will reciprocate, but that's not why you're doing it.

## Human relationships

We're told we should act "professional" with our co-workers, but that shouldn't be interpreted as treating them like robots. They're just as human as we are, at least in most cases, and the secret to a good professional relationship is having a good *human* relationship.

LISTER: *See, Rimmer, the trouble is you've never got time for people. You're too busy trying to be successful.*

RIMMER: *I have got time for people. What about all the time I spent licking up to Todhunter even though he was a total gimp? And Captain Hollister? I went out of my way to simper round him.*

—Red Dwarf, “Thanks for the Memory”

We're all very good at knowing when people are just being nice to us because they want something. It only works when you actually mean it. But then it works magnificently.

## Overcoming shyness

People are always astonished and dismayed to be told that they come across as unfriendly. Almost no one acts this way on purpose. Instead, they might just not have the knack of making light, friendly chat. By initiating the chat yourself, you'll often be able to break the ice and turn a cool relationship into a warm one.

And no one expects sparkling conversation at work. Just making the effort at all is the compliment you're paying them. “So, how about that local sportsball team?” won't win

you any prizes for originality, but it might win you a smile, and the next time will be easier.

## **Social bonding**

Socialising outside work helps a lot. Sure, this is hard if you work mostly remotely, but not impossible. Remote and distributed teams often have non-work get-togethers for just this reason. If your team doesn't do this, maybe it should. You can have a book club over Zoom or Teams, do a quiz, or even just block out the last 15 minutes of your weekly meeting for general chat. A formula that can help here is to ask each person to nominate something interesting but non-work-related that they read, watched, or heard about this week.

If your company or team doesn't organise social events outside work, try taking it on yourself to do so. Pick some activity that just about everyone can enjoy, even if it's just ironically: bowling, escape rooms, a board game tournament, go-karting, a movie marathon, a charity walk. The main thing is to have the chance to interact with your colleagues outside work: you might be surprised what a difference it makes to your subsequent interactions *at work*.

## **Listening**

Listening is another interpersonal superpower that's often unwisely neglected. Most of us are pretty good at talking, usually at considerable length, but someone who knows how to *listen* is a rare and precious find.

We tend to *think* we're listening to someone, when what we're really doing is working out what we're going to say next, and waiting for them to stop talking so that we can make our devastating point. And even if we're paying attention on some level, that still isn't quite the same as actually listening.

If you listen carefully to what someone tells you, instead of humming a tune inside your head and waiting for them to finish, you might learn something useful. Also, if you can keep quiet long enough for them to work through the process of turning their thoughts into words, *they* often learn something, too.

Experienced programmers call this *rubber-ducking*: sometimes, the simple act of explaining your problem to someone else in sufficient detail is enough to trigger the insight you need to solve it. Rubber ducks make better listeners for this purpose than most human beings, apparently.

What *we* can do that bath toys can't, though, is to add value to the discussion by telling the speaker what we think we understand them to be saying. Our understanding may be right, or it may be wrong. Either way, it's helpful for the speaker to know how we're hearing them.

## Reflection

True listening, then, is not a passive process, but an active one:

*I recall a time when I was talking with someone who seemed to ignore everything I said. “You are not listening to me!” I accused. “Oh, yes I am!” he said. He then repeated word for word what I had told him. He heard exactly. But he wasn’t listening. He didn’t understand the meanings I was trying to convey.*

—Robert Bolton, “[People Skills](#)”

Even if we’re showing all the signs of attending, the other person does not actually *know* that we’re listening unless we can show them that we’ve understood the import of what they said.

The simplest way to do that is to *reflect* it, by stating the essence of what they said, but in our own words. That usually means identifying the *feelings* they’re trying to communicate:

THEM: *That code review comment really made me angry. It was like “You’re so dumb for making this mistake.”*

YOU: *You felt attacked.*

THEM: *Exactly.*

When the speaker hears that we have successfully grasped their point, they usually say something like “Right,” or “Exactly”. If not, they will correct us. Either way, they are reassured that we’re at least trying to understand.

## Don’t make it about you

The mark of someone who’s *not* listening, by contrast, is that they jump in as soon as there’s a pause. This is like saying “Never mind that, here’s what *I* think...”

A common fault among poor listeners is that they immediately try to relate what the speaker is saying to some experience of their own. “The same thing happened to me once when I was on the way to Shelbyville. Now, I was wearing an onion on my belt, which was the style at the time...”

The listener means well, because they want to let the speaker know that they understand exactly what they’re talking about, by drawing a parallel with their own lives. But by breaking the speaker’s flow and changing the conversation to be about *them*, they inadvertently did the opposite of listening.

Instead, just let the person say what they want to say in their own time, and in their own way. And if it seems that they *don’t* want to talk, don’t try to force it: just stay quiet. If they want to fill the silence, they will. Otherwise, you can just be comfortably silent together. Not all conversation has to involve talking.

## Being a good co-worker

It's a curious feature of modern life that we're expected to have opinions on everything at all times, whether we're particularly well-informed about that topic or not. People have a bad habit of constantly probing and auditing each other's opinions, and challenging those that don't fit perfectly with their own.

The office isn't the place for this kind of thing (that's what the internet is for). Just let people think what they want, however nutty it might seem to you, and resist the urge to educate them on everything they're wrong about.

### Keep your opinions to yourself

This isn't to say, of course, that you should *have* no technical opinions about, for example, the right way to structure a particular piece of code. Naturally you will, and the more you become a master of your craft, the more such insights you'll have. But the way to use that understanding is by *doing* the right thing, not by arguing with other people about what the right thing is.

And it's always wise to remember that, however sure you may be about something, others may think differently, and that's okay. You don't need to badger everybody until they think the same way you do. After all, you've been wrong about things before, and now you know better.

You can't learn anything if you don't believe others have anything to teach you. Seek to understand, not to persuade.

### Don't be an advocate

One of the most annoying people in the workplace is the Advocate for something. You know of whom I speak. It's the person who thinks Linux is the greatest thing ever, everybody should use Linux for everything, and if you've made the mistake of buying a Mac instead, you're either misguided or an idiot.

Or maybe they're an Anti-Advocate: "Go is a terrible language, and you should feel bad for using it." Sure, buddy. Whatever you say.

These people are just *boring*, and their co-workers pretty soon learn to avoid their favourite subject, or to avoid talking to them altogether. But it's no good, because they will somehow manage to bring *every* conversation around to the topic.

### Exercising empathy

It doesn't matter how nice a person is in other ways: if they have no empathy, they'll be a bad co-worker.

PETER: *Milton, hi, uh... could you turn that down just a little bit?*

MILTON: *I was told that I could listen to the radio at a reasonable volume from nine to eleven...*

PETER: *No, I know you're allowed to, I just thought maybe as a personal favour...*

MILTON: *I told Bill that if Sandra's going to listen to her headphones while she's filing then I should be able to listen to the radio while I'm collating, so I don't see why I should have to turn down the radio...*

—“Office Space”

Perhaps this is less of an issue now that remote working is becoming more common, but when you share a workspace with other people, do be considerate about what noises, smells, or other distractions you might be emitting.

I once had a manager who played all his voicemails back on speakerphone, audible to everyone around. And he got a *lot* of them, presumably because he was the kind of person that people try to avoid face-to-face conversations with.

Everyone in the office hated this, but no one ever said anything. People don't tell you when they start hating you.

## Meetings

For some people, meetings *are* their work, but for most engineers they're simply an annoying time-suck, to be avoided whenever possible. When you can't avoid a meeting, though, here are a few tips for getting through them with as little collateral damage as possible.

### Less noise, more signal

Some people love the sound of their own voice, and ideally all those people would just get together and have one giant meeting, while the rest of us get stuff done. Until that happens, you can at least avoid making the problem worse by keeping your own contributions to a minimum.

Don't feel you need to say something just for the sake of it. Indeed, if you get through the whole meeting without being *required* to say something, then you probably don't need to attend those meetings in future. It's a win-win.

On the other hand, if you're only in meetings when you need to be, and you only speak when you have something to say, then you have a better chance of getting people's attention.

### The stakes can be high

Work meetings are an interesting social dynamic, because you're not just having a conversation with someone: you're talking to them *in front of their peers*, that is, effectively, in public. People are more sensitive in this situation than they might be otherwise. Something they might perceive as a slight, or that they think *others* might perceive as

a slight, can give offence in a meeting situation when it might not matter at all in a private chat.

If, as often happens, the boss is present too, that raises the stakes even further. Now it's not just your peers who are listening and judging you, but the person who has power over your compensation and promotion. Maybe even your continued employment. Under these circumstances, a word or two from your co-workers can do you great harm, or great good. It all depends.

The presence of a boss is like an amplifier for the risk or reward of what we say in meetings, and the bigger the boss, the greater the amplification. Group chats such as Slack or Teams are like a meeting that never ends, in this and other respects, so be careful. Treat anything you say or type in such a context as being said to everyone present, whether they appear to be listening or not.

And, it's important to add, treat anything you write to someone *privately* (for example, in an email or direct message) as though it will be shared with the whole company. One day, it might be. If you say something critical about a third party, assume they will see it sooner or later (and you may never know).

## **No one likes a doom-monger**

One of the most endearing qualities you can have as a colleague is *enthusiasm*. We all know the person who shoots down every suggestion in meetings with the response "Well, that won't work, because..." And we are not fond of that person.

Yes, it can be useful sometimes to visualise potential problems, especially if you can come up with solutions. But just rejecting all ideas out of hand isn't constructive by itself.

Even if you think the proposal is fundamentally flawed, you needn't phrase your response that way. You could say "That's a really interesting idea. I wonder if..." and you can talk about the pros as well as the cons.

## **We just want to be visible**

Have you ever wondered why some people seem to constantly speak up in meetings even when they don't have anything useful to say? The answer is that they want what we all want: recognition and validation. They want someone to say, in effect, "Yes, we see you, Bob, and we think you're a valuable member of the team."

It's easy to deny someone this validation, even when we don't really mean to. Consider an exchange like this:

FEMI: *I was wondering if it would be a good idea to store the what-nots in a shared database?*

ERIC: *What we should really do is cache that data in a key-value store local to the node, or maybe in a CDN.*

(MURMURS OF AGREEMENT)

See what Eric did? He trampled right over what Femi just said, erasing their suggestion and replacing it with his own. Even though the one idea leads logically on to the other, Eric subtly managed to suggest that Femi was wrong, and that, in future, people shouldn't listen to Femi, but to Eric instead.

Maybe he didn't consciously intend to send that message, but that's what people *heard*. We all know an Eric. How can we avoid *being* an Eric? By building others up, not shooting them down.

## **Validating and being supportive**

After this meeting, Femi may think twice about voicing their ideas in meetings in the future. They may also think, with some justification, that Eric is a jerk.

Here's a better way to handle this exchange:

FEMI: *I was wondering if it would be a good idea to store the what-nots in a shared database?*

ERIC: *That's a great suggestion, Femi. Could you talk about it a bit more?*

FEMI: (TALKS ABOUT IT A BIT MORE)

ERIC: *Yes, I like the sound of that. And what if we used a local key-value store, or even a CDN? Do you think that would increase performance?*

FEMI: *Yes, that's a great idea.*

Almost exactly the same conversation, at least on the surface, but with a radically different *subtext*. Eric first of all recognised Femi by name and validated that their suggestion was acceptable, and worth discussing. Rather than explain it further *himself*, as though it were his own idea, he invited Femi to talk about it, implying that he values their opinion.

Then he presented his own point, not as a substitute for Femi's plan, but as a natural evolution of it: "Yes, and..."

Rather than imply that he's correcting Femi because he knows better, he asks Femi for their opinion about *his* suggestion. By deferring in this way, he sends the message that he thinks Femi is qualified to pronounce on other people's ideas, as well as originating them.

Femi will remember *this* conversation very differently to the first version. They will feel that they contributed something valuable, that the contribution was recognised and appreciated by the group, and that it spurred a discussion leading to an even better plan.

As a co-worker, *this* version of Eric was supportive, listening to Femi's idea and inviting them to talk further about it, rather than trying to shut them down and talk over them.

## **"Say my name"**

When you're in a discussion like this, pay attention and watch for opportunities to do the same kind of thing yourself. If a co-worker makes a point, acknowledge it, support

it, mention them by name, and if the discussion leads to a fruitful result, show that you remember whose idea it was in the first place.

When you follow up on what someone else has said, tie it back to them by name, saying “To Helen’s point, it might be worth considering...” or “What Nasir said earlier reminded me that...”

When someone does this for you in return, take note of the warm feeling it generates in you: “I’ve been heard!”

## **Being a good co-worker is your job now**

I think we all realise that it’s bad manners to actively shoot down a colleague’s idea or code in a group setting. But ignoring them, erasing them, or even just minimising them is a more subtle, yet no less effective way of making yourself an unpleasant person to work with. The behaviour may not be malicious in intent, but that doesn’t matter: it’s the effect that counts.

In general, you should always be seeking out ways to be kind, friendly, and supportive to your co-workers. *That* is your real job, regardless of what your business card says. And it’s not difficult. All you need to do is be the person that *you’d* want to work with, if you had the choice.

## **Courtesy is a sign of strength**

Don’t expect everyone to respond in the same way, though. Some people won’t notice your kindness, some won’t value it, and some may actively exploit it. This is a shame, but there we are. Treat everybody with impeccable courtesy, no matter how they act in return, and you’ll have the high ground in every situation.

Never be suckered into tit-for-tat rudeness. When two people get up in each other’s faces, and the spittle starts to fly, no one watching will remember “who started it”. They’ll just remember two angry, rude people, and try to avoid both of them.

You can’t *like* everyone you work with: that’s impossible. But you *can* be polite to them, even those who don’t return the favour. *Especially* to those people, indeed. It’s hard to imagine that anyone actually sets out to be a rude jerk, but this just seems to be some people’s default way of interacting. It’s important to show them that it doesn’t work on you.

By being rude to you, they hope to engage you in a who’s-the-biggest-jerk contest, and drag you down to their level. Instead, you can wrong-foot them by being extra nice. That worries people, because they won’t understand what you’re up to.

*“Never think that being polite is a sign of weakness,” Chen said as he shook his head. “It is really a sign of great strength. When people are rude, they expose their weaknesses. When we are polite, our enemies cannot see our weaknesses.”*

—Phillip Starr, “The Making of a Butterfly: Traditional Chinese Martial Arts as Taught by Master W. C. Chen”

We'll have more to say about the game of office politics, and how to win it by not playing, in a later chapter. For now, though, let's turn to something that's very likely to be the limiting resource for your effectiveness at work, in your career advancement, and in your own life: *time* itself.