This chapter takes a high-level look at effective engineering communication, aiming to frame later chapters on email, meetings, and presentations.

I’m nothing special as a coder; I work hard and carefully, but I’m slower than the norm and probably not more detail-oriented. Still, I’ve been welcomed and rewarded more than I deserve on the many teams I’ve worked on, and I believe it’s because I work unusually hard at communication. Even simple projects demand constant communication, but we’re terrible at it as an industry, prone to excessive focus on the details, omitting context, and, above all, poor listening. I think that mismatch leaves us starved for clarity, which is an opportunity—in an industry of poor communicators, you can make a peculiar superpower of understanding exactly what people mean and capturing exactly what matters in response.
Receiving

We forget the obvious: our main goal in listening and reading email is to understand exactly what someone means and why they mean it. This is both a social exercise, an attempt to model the mind of our colleague, and an attention exercise, an attempt to control our desire to talk or daydream and focus on another person. Like coding or writing, some are blessed with a greater gift than others, but it can be cultivated, too; I’ll recommend a few techniques in a moment. If we don’t succeed in understanding what a collaborator is saying and why they’re saying it, it’s obvious that we may miss important input to real-world decisions—but it’s also worth noting that any effort to persuade or inform them is harder or impossible. Without first understanding our colleagues, we’re apt to tell them things they already know or don’t care about, to fail to address their concerns, or to fail to persuade them in the terms that matter to them; in short, any goal we have for transmitting information is far harder.

We also have a major secondary goal in listening: building relationships with our colleagues and allowing them the satisfaction of being understood. That might sound new-agey, but people love feeling heard and hate feeling misunderstood (or worse, ignored), and you can make a friend or ally just by showing you respect a colleague enough to listen carefully.

Here are a few suggestions:

- **Summarize**: The most powerful technique for crystallizing our own understanding is summarizing what someone else has said in a single sentence. This serves two purposes: it forces you to distill your own understanding, and it gives you a chance to validate externally. If you get it right, the other party will confirm (and feel affirmed and grateful)—if you’re wrong, you find out fast, get a correction, and still show that you’re doing your best. A good summary might start with an explicit acknowledgment that you’re trying to confirm you’ve understood someone:

  *Just to make sure I’ve understood: I think you’re saying that we shouldn’t tackle feature B until we finish feature A, because we can build B more easily on top of A.*

- **Shut up**: Try to let people finish in meetings; when reading, read the whole darn email. This is hard, and we all fail from time to time (especially me), but we should get off the ground, dust ourselves off, and try to shut up again. First and foremost, it gives us a chance to hear and learn, but it is also signals respect and humility to our colleagues.
• **Follow-up 1:1**: Asking follow-up questions is constructive and respectful, but distracting others in a meeting or by email often isn’t. I liberally follow-up 1:1 in person, chat, or email when I need clarification.

• **Do your homework**: Understanding our colleagues demands extensive context, both general (your domain and its technologies) and specific (your company’s systems). You signal respect and improve your comprehension when you research a domain before engaging with a colleague, particularly before asking for help.

• **Ask a thoughtful follow-up question**: For the most part, this is confined to clarifying things you don’t understand. In a 1:1 setting, it can be a good opportunity to connect as a fellow human being—showing a genuine human interest in another person is the easiest way to do that. For example, “How do you feel about that?”

**Transmitting**

Whether in person, on chat, or over email, the central concerns of conveying your ideas are, in approximately priority order:

• **Relevance**: Talking about things people care about and minimizing things they don’t

• **Context**: Making sure your audience has sufficient background to understand you

• **Clarity**: Capturing the exact idea you see in your mind’s eye

• **Efficiency**: Making effective, that is, minimal, use of your audience’s time and attention

• **Sensitivity**: Considering your audience’s emotional response to what you’re going to say, sometimes known as not making people feel bad unless you need to

These concerns exist in perpetual conflict—context can require verbosity, and sensitivity can conflict with the goal of maximal clarity (sometimes you need to add a spoonful of sugar). Combining them is an art refined only by years of practice, and I won’t try to offer a general-purpose algorithm here. However, I will give you some questions to ask yourself that can help you stay on course; I ask myself these questions about every nontrivial email I write or presentation I give (really).
Relevance

- Does this person care about what I’m talking about? A product manager likely doesn’t care about technical details (but engineers insist on telling them); a project manager is likely most concerned about schedule and staffing implications; a software engineer likely wants to know about your API.
- What does this person need to get done?
- What information from me helps them get it done?
- How does my work help or hurt their goals?
- Am I talking about what I care about or what they care about (it should probably be the latter!).
- Am I telling my audience something they already know?

Context

- Do my colleagues know the terms I’m using?
- Do they understand the technology well enough to follow my argument?
- Do they know the goals and responsibilities of the key people involved in this issue?

Clarity

- Is my sentence structure simple?
- Am I using appropriately simple language for this audience? Professors may need different language than interns.
- Can my audience hear me? Am I speaking clearly, loudly, and reasonably slowly?

Efficiency

- Could I say less?
- Might they already understand part of this explanation?
- Could I link to some an explanation instead of describing this here?
Sensitivity

- What could my audience be stressed/worried about? How can I reassure them?
- Am I saying anything that might accidentally insult someone else's technology?
- Is anyone involved personally invested in a specific outcome in a way that might make them more or less receptive to what I'm saying?

Asking yourself these questions can help you stay on the right track, but there's also powerful real-time feedback available: your audience's reaction. I've watched countless engineers blunder on when their audience has already signaled that they're bored or bewildered, and while reading your audience isn't always easy, you should be trying. For example:

- What do their questions imply about their understanding of what I'm saying?
- What emotions do I see on their face? Are they confused? Frustrated? Do they have something to say?
- Does their response suggest a fundamental misunderstanding that I can correct or an area where we have a very different model of the world?
- Do they seem impatient with my exposition? Could it be that they understand things better than I thought, or care about something different than I expected?
- Should I change directions or skip part of what I'm planning to say?

Choosing a Medium

When you reach out to a colleague in today's workplace, you can choose between instant messages (a.k.a. “chat”), SMS, email, phone, and videoconference, as well as dropping by in person or scheduling a meeting. Each channel has its own practical trade-offs and subtle social implications; this section will help you choose between them and will offer a few key tips for each. One option, email, is so important that I'll discuss it in its own chapter (Chapter 13). I consider three qualities when deciding how to reach out:

---

1I'm writing this during COVID-19 lockdown, so no one is doing any dropping-by in person whatsoever, but we're all hopeful for the future.
Sync/async: Synchronous channels, such as in-person questions, interrupt your colleagues, costing them productivity; asynchronous channels, such as email, let them respond at their leisure, making you wait but minimizing disruption. You should respect your colleagues’ concentration by going async when you can afford to; excessive interruptions cost social capital (see Chapter 7). However, you may have to choose a more synchronous medium if your subject is urgent or if you need to iterate together, which is too slow otherwise.

Bandwidth: A tiny question can be squeezed into the meager bandwidth of an instant message, while a very complex question demands the richer structure of an email or speed of human voice. An overly complex question in a low-bandwidth channel will frustrate other engineers.

Formality: This quality is a slippery matter of local convention, but important. The more onerous your request, and the greater your colleague’s seniority relative to you, the more formal should be your channel (and language). Generally, email > chat > drive-by question.

Let’s zoom in on our options.

Email

Fully asynchronous; high bandwidth; formality varies.

Email is the most important medium for office communication, so much so that I have dedicated Chapter 13 to the subject. Email is asynchronous, minimizing disruption to your audience. It’s quite compatible with formality, which makes it appropriate for questions to senior colleagues, though that may be overkill for casual discussions with peers. It’s by far the easiest way to reach a large audience. Its greatest weakness is lack of interactivity; it tends to be a quite poor way to iterate on complex issues or resolve controversy compared to in-person discussion.

Chat/Instant Messaging

Semi-synchronous; low bandwidth; informal.

Chat is a fast, loose, casual game and a critical part of engineering collaboration. It’s great for communication that’s informal (low-stakes issues, people you know well), interactive (more back and forth), and semi-synchronous (I need information soon but not immediately), but its low bandwidth makes it a poor choice for capturing real complexity. If you need to carefully marshal a lot of information, email is your friend; if you need to go back and forth at length, you want to use your voice.
The principles for effective chat are similar to those of effective email—favor brevity, arm your audience with context, and reply as quickly as you can. I’ll offer three chat-specific tips:

- Many engineers become enraged when someone sends them an opening chat (“hi pete!” or “hi” or “do you have a second”) and don’t send their actual question at the same time. This doesn’t bother me personally one bit, and I don’t consider the pet peeve well-founded, but you should stay on the safe side and avoid this pattern by sending your question along with your greeting.

- Decent punctuation is advised; capitalization is not required and may come off a touch stuffy. Emojis are encouraged. I don’t know why those are the conventions, and they may change anytime.

- Chat is a poor choice for recording a decision, because decisions should be permanently recorded, easy to search, and subjectively more formal. Don’t leave a controversial decision recorded in chat only; use email.

Drive-By Questions (Going to Someone’s Desk)
Fully synchronous; high bandwidth; informal.

Walking over to someone’s desk interrupts their flow completely; it shouldn’t be done lightly. However, when you’re stuck and want to work through something tricky with a colleague, it’s the way to go. I strongly advise you to always ping by chat first; that way you at least show that you respect your colleague’s right to concentration.

- Do you have a moment to chat about how to add an endpoint to the Inventory service? I can also schedule us some time later in the week if that’s better.
- Sure, I have time now, what’s up?
- OK if I come by your desk?
- Come on over!

Once you’re standing over your colleague, you should be looking for the first sensible opportunity to let them get back to work! The engineer who comes by your desk and won’t quit talking is a cliché to avoid.
Scheduled Meetings

Asynchronous; high bandwidth; formality varies.

Scheduled meetings are asynchronous (i.e., less disruptive than a drive-by question) and appropriate for complex subjects; they are, however, very costly and should be reserved for cases that really require that level of bandwidth. See Chapter 5 for a discussion of effective meetings.

SMS

Synchronous; low bandwidth; informal.

In my experience, SMS is only used for time-sensitive communication, for example, about paperwork on a deadline, when chat has already failed for some reason. It’s great for those cases but too disruptive for nonurgent messages and not synchronous enough for something like an outage, which can tolerate no waiting.

Synchronous Phone Calls

Completely synchronous; high bandwidth; informal.

Similar to SMS, I mostly see synchronous phone calls (i.e., not planned meetings) used for extremely urgent issues, like ongoing outages.