

C4E - Communication for Engineers

An Interactive Course for Engineers by an Engineer

chrislaffra.com/c4e

May 2021

Introduction

The typical software engineer at large tech companies tends to be great at solving problems, analysis, coding, debugging, and testing. However, they tend to be not so good at marketing their own contributions. Being able to explain what your contributions mean to the rest of the company and how you make a difference is a crucial skill towards growth and being successful as an engineer.

To develop communication skills, this course focuses on communication as a “soft” skill to develop and nurture. The course provides concrete tips and guidelines on topics of communication that actually matter for engineers. This course was developed by an engineer with multiple decades of experience being an engineer in large corporations. We outline tips and tricks for software engineers to make them more effective communicators and as a result increase their impact at work, get more opportunities for growth, and enjoy their job and life more.

In [How Computers Learn](#), Peter Norvig, head of AI at Google, reports how Google uses machine learning for hiring and one of the discoveries was the fact that being good at programming competitions correlates negatively with being good on the job. In other words, it takes more than innate coding skills to be successful at a corporation. The missing link is collaboration and communication, which we will focus on in this course.

The course is based on the [Communication for Engineers](#) book. Students will receive a free copy of the book ahead of the course. The course will cover topics from the book, but can also be customized per needs of the organization. For instance, a decision could be made to focus more on topics such as Goals/OKRs, time management, managing emails, how to document effectively, code reviews, and how to keep on growing. A customized plan is possible.

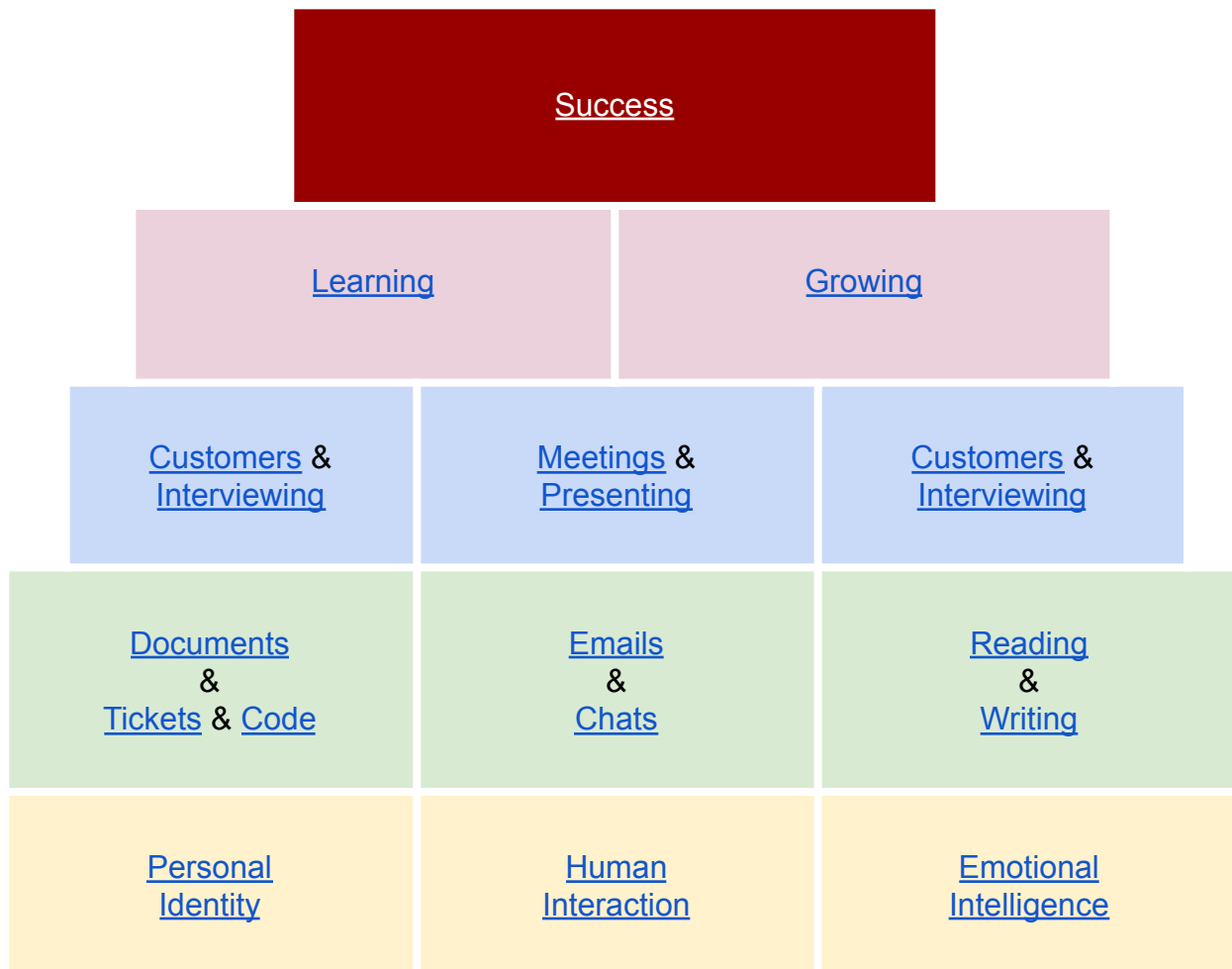
[Top](#)

[Bottom](#)

Course Format

The C4E course contains various topics, with numerous exercises, taught in an instructor-led training session. Below, see the communication pyramid of skills that is used to structure the course. The base of the pyramid is formed by basic human skills, in yellow, not specific to engineering. Above that, in green, we find objects engineers interact with and basic communication skills. In blue, we find interactions with other people, including non-engineers. Finally, in pink, we find meaningful skills that grow us as a successful engineer and that prepare us for our eventual goal, in red, which is getting recognized for our achievements.

The labels in each box are links that lead to a survey. The surveys are used to measure current level of proficiency of communication skills. During the course, survey results are discussed and analyzed with the group, to increase common understanding and team bonding.



The communication Pyramid of Skills

[Top](#)

[Bottom](#)

The course is taught in weekly sessions with a total of 12 hours using a combination of:

- Slides.
- Stories.
- Exercises.
- Interactive surveys.

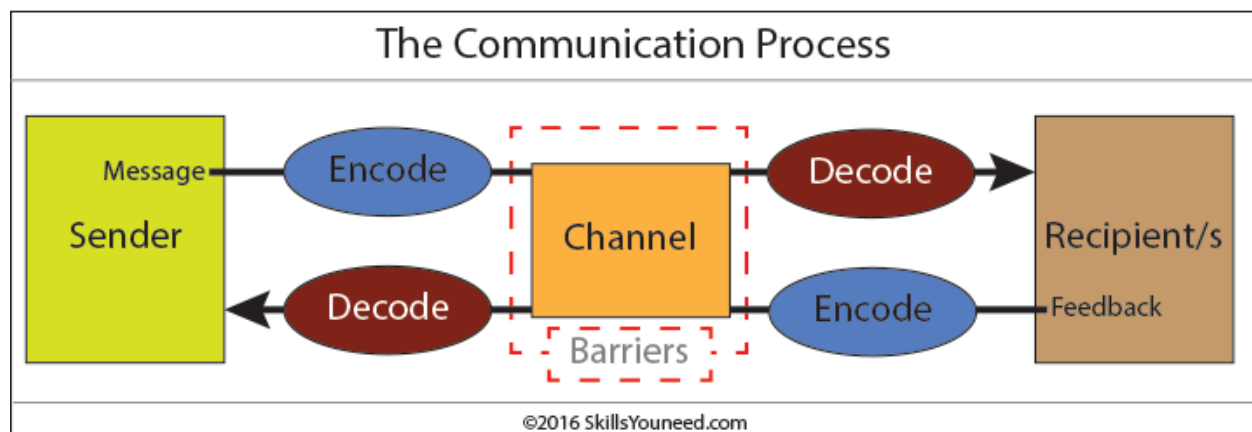
During the course, the teacher will review the surveys and dive deep into any of the topics that need further discussion.

As a follow-up to this course, the students will get:

- Access to and ownership of their survey results.
- Repeat monthly self-evaluations to check up on their progress and go red to green.
- Spaced repetition reminders on specific tips on things they can improve on.

Communication for Engineers

We will go over a number of situations where an opportunity lends itself for engineers to communicate with other humans, engineers or not. Every form of interaction with other people is a form of communication. This becomes clear when we look at the communication *process*. An engineering-friendly definition is provided at skillsyouneed.com:



By using a methodical approach to communication, attendees will become a more effective software engineer, market themselves better, prepare themselves better for promotion, and excel at collaborating inside and outside of their current company.

Emphasis on Growth

From [JHana](#): **Communication** — Listens, speaks and writes clearly and concisely. Below is an analysis from JHana on different levels when it comes to communication skills. Attendees to the course are currently at levels 3, 4, 5, or 6. The goal of the course is to lift them to the next level.

Rating	Description
4	Listens in order to clarify information, sends both written and verbal messages in a clear manner, uses clear language and avoids jargon, asks questions to test for clarity and understanding, explains complex terms, uses concrete examples
5	Provides timely, appropriate and useful information to others, determines the most effective method of communication, tailors messages to the needs of the intended audience, uses personal judgment to determine what information is useful to co-workers and clients
6	Increases the value of information by providing sound interpretation, communicates clearly and succinctly in any setting, uses personal and organizational knowledge to enhance the value of information for others, proposes uses for the information, makes his/her point with resistant audiences

Communication Opportunities

One might think that the only task of engineers is to write code. But, that's a fallacy. The role of engineers is to solve *problems*. To discover what problems they are solving, engineers need to ask others. This requires communication. Furthermore, hardly any engineer writes code all by themselves. They work in a team. Teams need to coordinate their work. Again, it is hard to solve problems as a team without any communication.

In that context, engineers have a wide variety of outlets for communication, which we will go over in this course. For now, here is a quick summary of the various areas:

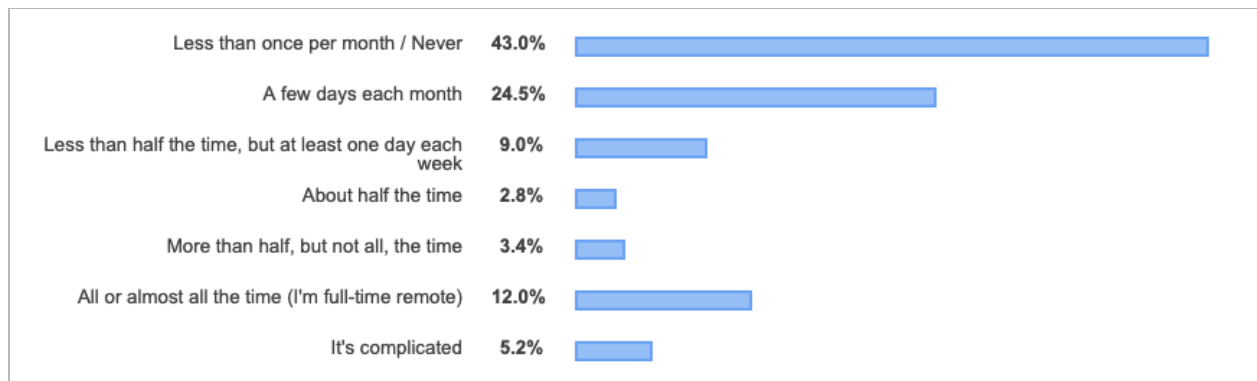
- **Identity.** How do you represent yourself well? Examples are you company's corporate directory, LinkedIn, business cards, etc. [More details](#).
- **Doing Work.** How do you communicate effectively while working? Examples are design, coding, meetings, comments, code reviews, tickets, etc. [More details](#).

- **Writing.** How do you excel in written communication? Examples are email, calendar, chats, groups, etc. [More details](#).
- **Publications.** How do you publish about your hobbies or your work? Examples are blogs, tutorials, vlogs, presentations, etc. [More details](#).
- **Growing.** How do you efficiently document your own impact? Examples are performance reviews, promotion packets, learning, etc. [More details](#).

Being Remote or Working from Home

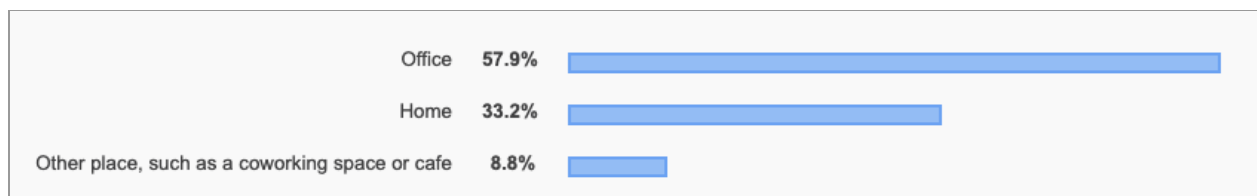
Being remote will make it even more important to focus on a structured approach to communication. However, teams with local engineers have similar communication challenges as remote teams, where applying structure to one’s communication will be useful.

According to [Stackoverflow’s Developer Insights](#) report for 2019, working remote is still the exception, rather than the norm:



The interesting answer category is the last one: “It’s complicated”. Communication skills are soft skills for most engineers and being remote, any lack of communication skills is multiplied and turns them from a minor nuisance to a major blocker for success.

Further on in the same insights, when asked, where they would prefer working if they had a choice, 40% of engineers would prefer to not work in the office:



This means that 40% of engineers would love to work outside the office, but only 12% of engineers actually get to fulfill this dream full time. Why? We can only guess this is due to missed opportunities at work, growth limitations, and not being recognized for one's contributions. All of those are caused by communication challenges. In this course we aim to analyze where opportunities arise where you can market yourself and get recognized regardless if you work in the office or not.

Emotional Intelligence

Before discussing the topic of communication in detail, let us establish the proper context and define what we actually mean with *communication*. communication is a purposeful technique of communicating with other humans to make them recognize your contributions and impact.

Unlike analytical skills such as problem solving, design, coding, and debugging, which are *hard skills*, communication is a *soft skill* requiring [emotional intelligence](#). Most engineers think they are bad at communication skills. However, emotional intelligence is mainly a balanced set of mostly *learnable* skills, being self-awareness, self-regulation, empathy, motivation, and social skills.

Self-awareness

In 1972, psychologists Shelley Duval and Robert Wicklund's developed the theory of self-awareness:

“When we focus our attention on ourselves, we evaluate and compare our current behavior to our internal standards and values. We become self-conscious as objective evaluators of ourselves.”

Self-awareness is your “inner voice” telling you about your behavior, such as when you are interrupting someone, when you sound aggressive, or when you feel grumpy. Another example is having an “outer body experience” while presenting, realizing you are saying “uh” all the time.

The ability to be self-aware is critical to developing emotional intelligence. Without it, you have no way to monitor your own behavior. In relationships, partners often supplement a lack of self-awareness by providing feedback when certain standards or values are not met.

Always listen to feedback from peers, other colleagues, or anyone else you trust. Their suggestions, comments, or even body language can compensate your lack of self-awareness.

Self-regulation

The purpose of self-regulation is to manage your own emotional state. Rather than make impulsive decisions or reacting with emotion, self-regulation is the process of recognizing what you are doing and controlling your own impulses. Self-regulation is needed to avoid abuse of physical or mental addictions. However, the same self-regulation is also needed to avoid being grumpy in the morning, reacting aggressively to investigative questions, or to protect yourself from “shutting down” when you are in any conflict situation.

Empathy

Empathy is the skill to see the world from someone else’s viewpoint. How do other people think? What impact do you have on them? Can you communicate to them in terms that make sense to them?

Motivation

The ability to use emotion effectively to maximize your goals. The C4E course provides a lot of guidance on how to plan efficiently.

Social skills

The ability to influence other people’s emotions.

Three Mindsets

From the [course description](#) for “Leadership Essentials for Engineers” from TU Delft:

A good leader needs at least three mindsets, supporting the development of essential leadership skills:

- The **analytical** mindset. This mindset is about the ability to analyze complex problems and opportunities as a basis for making strategic decisions. It is about developing a strategy in a systematic way and preparing for action. This mindset taps into the strengths of engineers.
- The **influencing** mindset. Designing solutions and strategies, however, is about more than analysis. In our interconnected and complex world,

leaders are faced with stakeholders, both internal and external, who might have completely different views. Problems and challenges are often ambiguous. This mindset looks at all these issues and finds common ground and ways of producing willing consensus in teams and stakeholders.

- The **sensemaking** mindset. The first two mindsets require a high tolerance of complexity. Leadership is also about communicating, inspiring and convincing people and teams inside and outside the organization. This mindset is about translating complexity into powerful and concise messages so people can understand and make sense of the complexity of their world.

These three mindsets are indispensable for good leaders. They sometimes conflict and they often require completely different skills. In this program, you will be familiarized with each of these mindsets and discover how they can support you as a leader.”

Engineers are great at being analytical. We often measure our value and impact into how well we can design a complex system, build it under time duress, and develop high quality code at the same time. A great engineer is measured by the number of diffs put into the system.

It is not a coincidence that technical interviews at places like Google, Facebook, Uber, etc. are often referred to as “Coding Interviews”. However, how many times will you be interviewed on your ability to influence others? Influencing others is just as important a skill to have. communication is an explicit form of influencing that is especially of value during an interview.

The sensemaking mindset is about explaining complex things in simple terms. This requires empathy, translation skills, and patience. In the words of Albert Einstein: “If you can't explain it simply, you don't understand it well enough”. This is what others will think, when you do not explain your contributions well.

Interacting with People

Audience

When you communicate with individuals, give them a personalized message. Your audience typically exists of the following types:

- Engineer
- Manager
- Director
- VP
- Product Manager
- Designer
- Data Analyst

Avoid biases or stigmas. Be considerate of someone's preferred gender.

Introversion

Whether you are an introvert or an extravert yourself, help introverts get their opinion heard. Diversity of opinion increases the value of any decision making process.

Professionalism

Avoid sarcasm and offensive statements. This applies to chats, emails, personal interactions, code reviews, etc.

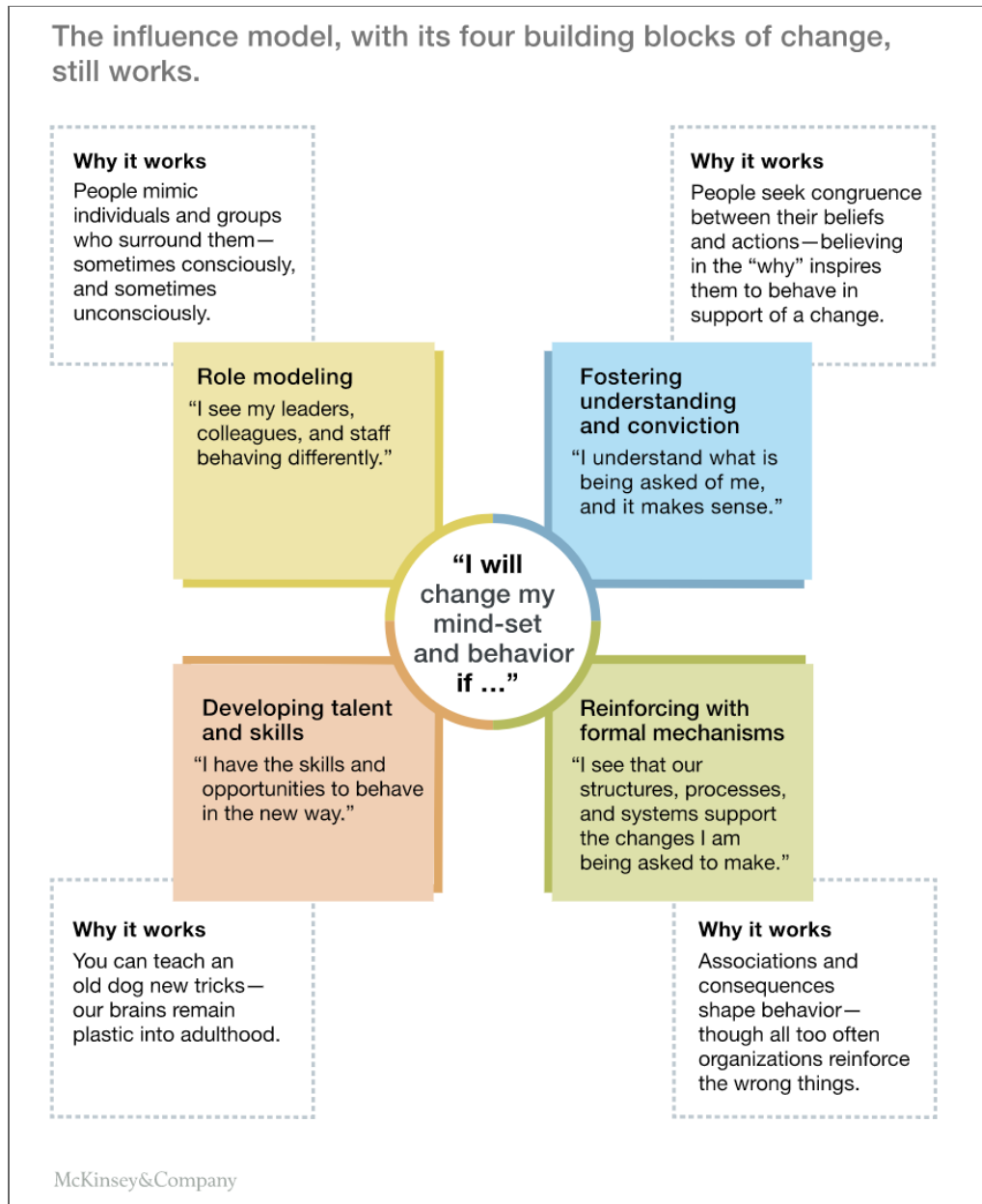
Toxicity

Great technical contributions can lose their value if others consider you "toxic". A toxic person acts like this:

- Makes everything be focused on them
- Criticizes others, whether they are present or not
- Demonstrates selfishness, lust for power, or being greedy
- Turns every discussion into a contest that they have to win
- Expresses being jealous of others.
- Land grabbing

Think of a person who you worked with that you hated interacting with. What makes you hate that person? Don't be like that.

Influencing Others



In [The four building blocks of change](#) a case is made for a practical model to be deployed when trying to influence others. If you want others to change, i.e., do what you want, you should ask yourself:

- Am I leading by example? In both my self-image and my communication, do I provide concrete examples of the intended behavior?
- Am I clearly saying what is expected from the others? Do they understand why?

[Top](#)

[Bottom](#)

- Do the tools and processes support the new approach? Am I setting people up for failure if the things I am asking them to do are too hard to implement?
- What skills need to be developed to be successful in the expected new behavior? Do I need to help them with training?

Consideration

Praise in public. Give feedback in private.

Discussions

Be hard on problems, not on people. Open debate is good.

Identity

Communication starts with a proper understanding of who you are. When you ask an engineer who they are, they often respond with *what* they are. A representative example would be “I am an Android engineer and I work on project X.” This is not what we are looking for when we ask “Who are you?”. When we drill down, we get more details out of them, such as their hobbies or their passions. Engineers should take the time to define their own identity, even a *brand*, as it will help them to motivate their actions leading to the goals they set for their life and career.

Your Personal Elevator Speech

In one paragraph, or 30 seconds, tell people who you are.

In May, 2019, Austin Geidt rang the bell at Uber IPO at the New York Stock Exchange. She joined Uber as the first intern. As part of her interview, the then-CEO, Graves, [asked her](#) to create a presentation about herself. Her deck was full of humor and pleas for Graves to give her a shot. Your personal elevator speech is exactly that. Consider it a tool to get anyone reading it to give you a shot, i.e., convince them to spend some of their precious time on you.

Employee Directory

Each company has a hierarchical employee directory. You have a page there. This page should contain your elevator speech. Explain what project you work on. Links to more things about you.

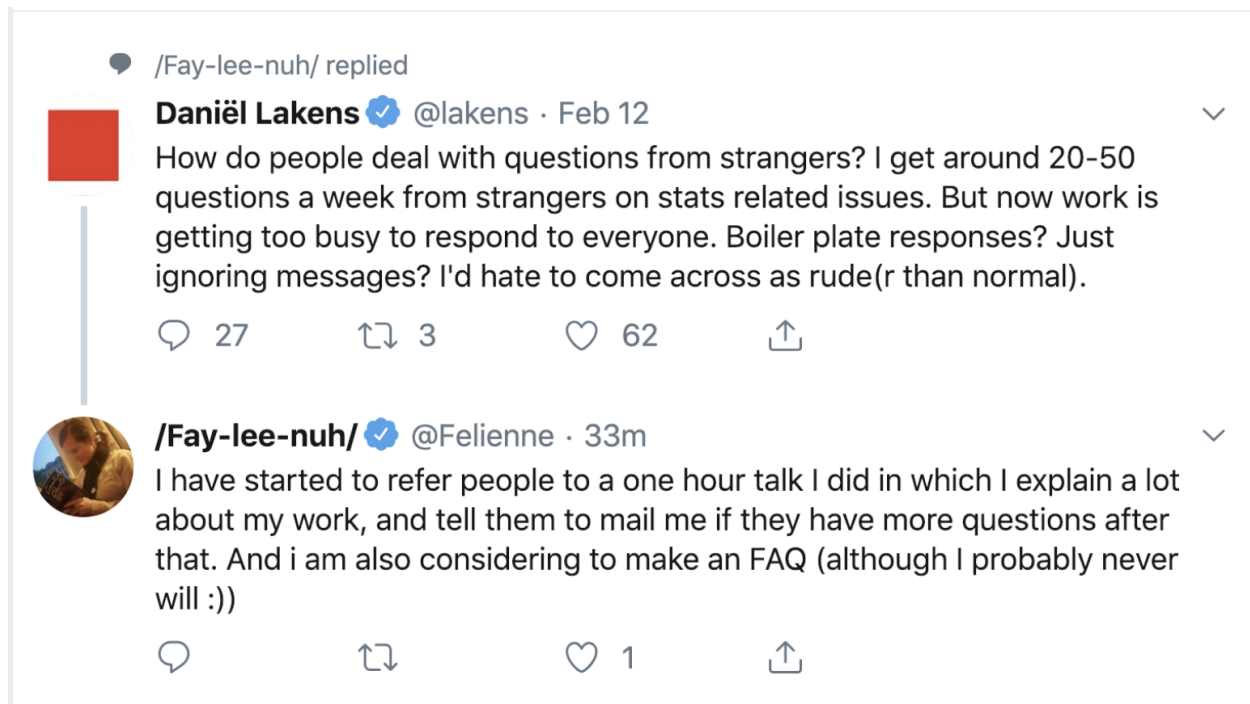
Get a good picture. Upload it.

LinkedIn

Who is your audience? Other people that want to join the company you work at. Explain what you do there.

Get a good picture. Upload it.

FAQ



The screenshot shows a LinkedIn thread. At the top, a grey speech bubble indicates a reply from "/Fay-lee-nuh/". Below it is a post from Daniël Lakens (@lakens), a verified user, dated Feb 12. His profile picture is a solid red square. The text of his post asks how to deal with questions from strangers, mentioning he receives 20-50 questions a week and is currently too busy to respond to everyone, considering boilerplate responses or ignoring messages. Below his post are icons for replies (27), retweets (3), likes (62), and a share icon. Below that is a reply from "/Fay-lee-nuh/" (@Feliene), a verified user, dated 33m. Her profile picture shows a person's face. Her text explains she refers people to a one-hour talk and asks them to email her for more questions, and she is considering making an FAQ. Below her reply are icons for replies, retweets, and 1 like, along with a share icon.

Doing Work

Standups

Prepare for standup. What did you do yesterday? What will you do today? Where are you blocked? Can you be brief? Take deep-dives offline for later. Take a glance at the roadmap. Are priorities still in check? How can you unlock others? Volunteer to be the first to start with your practiced sentences.

[Top](#)

[Bottom](#)

OKRs

Objectives and Key Results. What does your team focus on? How do your personal goals relate? How can you contribute to the creation of the OKRs?

[The Definitive Guide to OKRs](#)

Personal Goals

Set [SMART](#) goals. Lots more on goal setting is covered in detail in the course.

Project Sprint Planning

Focus on customer requirements, priority, team capacity. Prepare by taking a look at your team's OKRs. The main question you have to answer is: what can I work on in the coming Sprint? Be confident in your choices, refer to priorities. Be able to quickly explain tasks to team members. Do your homework, so you can be confident. Remain positive, humble. Know the team's vision.

Customer Engagements

Listen. Show empathy. Understand their needs, timeline, priorities. Listen. Listen. When you have a complete picture, then offer how your personal skills or your team's solutions can help them. Identify gaps. Plan for bridging the gaps. Write OKRs and design documents together.

Tickets

Add a clear explanation. Explain context. Link to design docs. Attach diffs. Make the ticket tell its own story, so you can share it with someone else and they fully understand its context.

If the ticket is a bug, include:

- How you used the product (add detail such as OS/build type) and exactly what you did
- What you expected
- What you got instead
- Optional: Why you think this is a bug

This is general advice: always respond. Even if you have nothing to say. In the course, we elaborate on this communication task in more detail.

[Top](#)

[Bottom](#)

Code

Writing Code

For engineers, code is their main form of output. People judge you on the quality of that output. Things to worry about how others perceive your output:

- Did you add tests? Do you need to update existing tests? This is not something you can do later. You have to add tests as part of your code.
- Naming things. Choose appropriate names.

Did you update relevant documentation? Fix the docs. If you cannot fix them now, open a ticket.

Committing Code

A good diff describes WHY the code was changed. The code itself shows HOW it was changed.

Add screenshots, video, or instructions on how to run the diff locally.

Work is planned and prioritized using tickets. Attach the appropriate ticket to your code change.

See [Commit Messages Guide](#).

Reviewing Code

A code review is not a boxing match. It is not the place to “win a contest”. Be nice, both to code owners and reviewers. Do what you can to make the process objective and ego-less.

Never be judgemental in review comments. Avoid starting a comment with “Why did you...” or “Why did you not...”.

WHAT - NOT - WHY - objective versus accusative

Code review comments have different focus:

- Architectural. The code is using the wrong library, the wrong service, etc. Ask for clarification. Suggest an alternative. Ask if the alternative was considered.

Follow up on all comments. Even trivial ones.

Error Messages

If you have a UI or an API, you will inevitably generate error messages. This is another area where you, as a developer, are forced to communicate with other people.

Acronyms

Engineers live in a universe of acronyms. Try not to use them when you can. If you have to, spell out the full definition at the first usage. Better yet, avoid acronyms.

Documentation

Users of your product and stakeholders such as other engineers will read your docs. The docs represent your work. Good docs usually go unnoticed. Bad docs make you look bad.

Don't let docs get stale. Aim for concise docs.

Looking Busy

Engineers make an impact by adding value, not by working hard. But, the reality is that there will always be people who still judge you on how hard you appear to work. If you are in a situation that makes you less visible to your team (whether for personal or work meetings or travel), let your team know. Over-communicate on your status. Be visible in chat rooms.

That said, be mindful of being too busy just for busy sake. Work-life balance. Learn the art of [niksen](#).

Documents

[Writing is thinking](#).

Sharing Docs

Don't wait too long. Collaboration is all about doing work together and eliciting feedback. If your doc is "perfect" and completely finished, there will be little opportunity for others to contribute and feel really part of the work being described in the doc.

Write the basic outline. Fill in. Share with small group. Use comments to elicit feedback. Share with greater audience.

If you work for someone and they asked you to capture a plan or retrospective in a doc, immediately share it with them while you are working on it. They will love to see the momentum and progress. They will understand that your doc is not yet finished or shareable.

[Top](#)

[Bottom](#)

Find your **stakeholders**. Add them as approvers.

Organizing

Most software development teams make a mess of their design docs.

If you like any doc, star it.

You *will* need to find back the design doc in the future, so go ahead and create a shared drive. Organize the drive with meaningful subfolders. Add your design doc. Make everyone on your team share their docs in the same place. Advertize the shared folder as much as you can. You can use subtle hints, such as “Hi team, I wrote a doc to plan our upcoming offsite. I put it in our shared drive.”

Design Documents

Design Documents (DD), Requests for Comments (RFC), or Engineering Review Document (ERD). Different companies use different names for the same intent: a document that describes the *execution* of a strategy or vision.




Steven Sinofsky 
@stevesi 

Replying to @stevesi

Execution is in a constant state of “diverging” as more expertise deals with more details that fewer people understand.

The act of writing forces a team of experts to share the details of goals—not just the what, but the why, what else was considered, the history, context.

 294  8:03 PM - Apr 19, 2018 · Menlo Park, CA 

 60 people are talking about this 

Structure

Use a company-defined template, if available. The ERD process used by Uber looks like this:

[Planning at Uber.](#)

Why are you writing this doc? Focus on three main things:

- What is the problem?
- How will it be solved? What architecture? What solutions?
- What is the timeline?

What to include

When creating any design doc, do this:

- Add a relevant title
- Give the document a meaningful, short name
- Add your name and a way to contact you (email or employee directory link)
- Add a date, typically month+year suffices
- Add relevant links that add context to the doc, such as your team's project
- Add an abstract or introduction that explains the goal of the doc
- Add graphics, such as a project logo, a screenshot, or a diagram

Collaboration

In Google Docs, use comments to elicit feedback from others by using comments. Assign comments as tasks. Resolve comments as soon as possible to remove noise. Follow-up on time sensitive comments and ping people. Be proactive. When you have questions, make them explicit by assigning comments as a task to an individual.

Meetings

Attending Meetings

Is there an agenda? Consider [framing agenda items as questions](#).

What is your role in the meeting? How do you make an impact?

Are there any action items to gather from the meeting?

Can you take notes?

[Top](#)

[Bottom](#)

If this is a recurring meeting, did the owner follow up on previous action items?

Why are you in the meeting? Do you find yourself checking emails on your laptop or phone during the meeting? Feel like you did not contribute enough to the meeting? Next time, decline.

Holding Meetings

Make sure you have a good title. Think about it. Make it short, but meaningful.

Add an agenda.

Take care of action items.

Attach relevant documents to the invite.

Keep notes. Send the notes to all the attendees. There may be a larger audience interested in this meeting.

Keep meetings short. The typical one hour meeting can be done in 30 minutes if it is well run.

Presenting Slides

[What is a Pitch Deck?](#)

No bullet points. Images.

Use a company defined template.

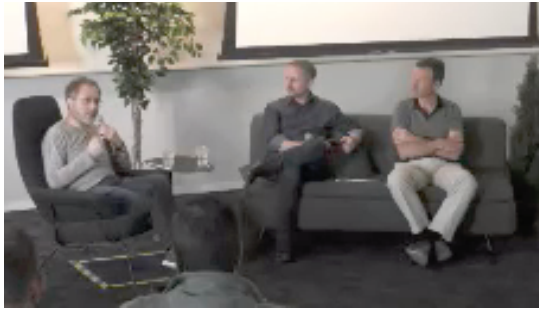
Try the [2 minute elevator speech](#).

Shared Calendars

Add all-day meetings, not ones that color the entire calendar pink or green.

Writing Well

Urs Holzle, employee #8 and CTO of Google, at a fireside chat at Uber in May 2019 when asked on how to become successful at a career shared the following [insight](#):



“What many people underestimate is that being a good writer, whether that is through emails or through documents, allows you to be more impactful. I see many engineers ignore that skill. You might be proud about your *code*. You should also be equally proud of the craft of *writing*, it is just as important. Every company, especially those that matter,

gets to scale. When you get to scale, oral communication does not work that well anymore. If you want to make something that I say more durable, you have to write it down. And, if you want it to be actually read, you have to be good at writing it. That way, they will want to read it and they will retain what you want them to retain. Writing is a highly underestimated skill for engineers.”

Later on, Urs followed up with “reading is the most important superpower. You should probably develop that skill first, before becoming a good writer.”

Email

In the course, we will discuss how to write good emails, focus on the intended audience, and how to avoid getting overwhelmed with email.

Email Audience Structure

See [JHana tips on Email](#).

In [How to start an email to your boss](#) good advice is given to add focus to an email for people that have to read hundreds of emails a day. Of course the same advice is valid for anyone. Not every “maker” gets 500 emails a day, but they may have the same small amount of attention to give to an individual email. So, structure your email effectively for the best result, regardless of who you send it to. Below are a couple of examples.

Signature

In [The Significance of a Signature](#), the authors make a point of the signature being the equivalent of a business card. The average person gets around 800 emails per month. For

engineers in tech companies this must be much higher. That said, each impression is an opportunity to market your company, your team, and yourself.

Earlier we discussed how to create a personal elevator speech. Now, try and reduce that elevator speech even more, basically into something that fits on a t-shirt. In contrast to the elevator speech that should focus on *who* you are, the signature should focus on *what* you are.

Here is an example of my signature at Uber that shows my name, linking to an internal directory, followed by my role on my project, followed by links to details about my project:

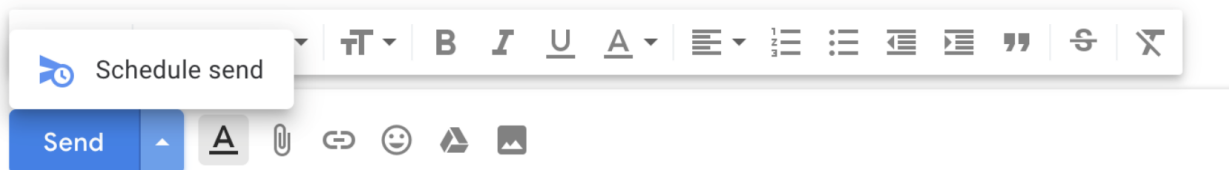
Chris Laffra, TLM on Screenflow ⇒ [Intro & docs](#) & [IDE](#).

The majority of people you interact with are internal, as is the case for my email signature shown above. If you do interact with people outside your company, your signature should of course contain your company name and not include any non-public links.

Setting up a [signature on gmail](#) is trivial. Do it now.

When to Send?

Work-life balance. Do not send emails on the weekend. Gmail allows you to schedule sending emails, to do so just click on the little triangle next to the send button which will pop-up a "Schedule send" option.



Who to Send To?

Avoid the bystander-effect. Send to individual people. Include clear call to action.

Scripting

Use Google scripts to send individual emails, rather than blast in gmail.

- Promo
- Office hours

- Promo peer feedback
- External sourcing

Chats

Collaboration is hard. And chats are [not the answer](#).

Perf and Promo

The ultimate opportunity for communication is the (bi-)yearly performance process.

Promo is related to Perf, but typically it is a separate process.

The best way to get promoted is by not trying to get promoted. Promotion basically means correction of a situation where you are demonstrably operating at a L+1 level for a duration of, say, at least six months. The promotion packet is to show to an independent set of observers what you did, how you did it, what skills you leveraged, and what skills you still want to grow. The packet is then used by the committee to evaluate you.

Communication angles to consider as an engineer:

- You need to describe what you did over a considerable period, condense it, and highlight specific contributions or actions that will make your case. This assumes you are able to write a good self-review.
- Selected peers will do a peer review, independently writing up your evaluation. Here, the communication is more personal, sustained, and heavily influenced by the many forms of communication discussed all over this course.

With regards to promo, your motivation should be on being productive and effective, growing your skills, focusing on doing the right things, and the rest shall follow. That said, communication skills *definitely* make an impact.

Anti-Procrastination

Keep notes during the year. Organize by project that you work on. Keep sections on What - How - Skills - Growth. Add links for design documents, diffs, tickets that you are involved in.

Also keep a doc on others for whom you are likely to be a peer reviewer. Again, use What - How - Skills - Growth. When you provide feedback to someone, or see them do something special, make a note.

Elevator Speech per Project

Can you succinctly summarize the projects you worked on?

What was your specific contribution? Be detailed. Have proof.

Can you help your peer reviewers say something nice about you?

Giving Feedback

Can you identify growth areas for a given individual? Can you articulate your thoughts? How do you write down negative feedback and turn it into something positive?

Growth

The course emphasises topics, such as leading without authority, coaching, mentoring, and promotions.

Interviewing

If there is any process where your communication is tested to the extreme, it would be a technical interview. A breakdown of the [interview process](#) includes numerous opportunities for communication where good preparation can make all the difference.

Learning

Increasing Personal Relevance

When you learn a new technology, you can add more personal relevance by documenting what you learned in the context of your current project.

As an example, let us assume you attend an Android Bootcamp. During the bootcamp you will learn basic concepts, see examples, and do hands-on experiments. Make them more personal by looking at your team's code. When writing example code, write it in the context of your team's app. Try and add something extra to the course. When learning how to use the debugger, debug your team's app, not HelloWorld. Make changes to your team's app, see what happens.

[Top](#)

[Bottom](#)

Plan ahead to share your learning experience:

- Start a new Google doc, keep notes, share with your team after the course ends.
- Post your experience on LinkedIn. Summarize what you learned and how awesome it is that your company sends you to training courses like these.
- Coming back from the course, set up a meeting with your team, give them a one hour summary of the course. Share your experiences.
- Based on what you learned, find ways to optimize your team's productivity or product quality. Open tickets to turn what you learned into concrete actions.
- If you work at a site that has no direct access to the course, consider doing a (shorter version of) the training you had to your colleagues at the local site.
- Consider reusing the course you followed and teach a summary at a local meetup where you live. If you learned a lot and you feel like it, consider given the same course at a technical conference.

Goal Setting

If you want to really learn something, sign up to write a book about it. You will be forced to become an expert at the subject and set yourself a concrete deadline to plan towards. If a book is too much, you can pick one of the following formats to target your learning, in increasing cost of investment and yield:

- Send a chat or an email to your team
- Write a report, summarize what you learned
- Do a presentation to your team, group, or site
- Create a github project
- Post at LinkedIn, Facebook, or Twitter
- Contribute to Open Source, add docs, fix a bug
- Write an article on a public site, such as LinkedIn or Medium
- Present a talk at a conference
- Organize a workshop at a conference
- Give a tutorial at a conference
- Write a book

Public Speaking

communication eventually involves speaking to a crowd.

Setting the Stage

Don't forget to give the audience your name. Speak slowly while saying your name. Repeat it.

Elevator speech. Make sure they can find you.

Telling a Story

Fact-based vs. Story-telling.

Slides with pictures, avoid bullet points.

Practice makes Perfect

Practice in front of a mirror. Practice among friends.

Body Language

Confidence. Eye contact. Use your hands.

Wrapping Up

Leave time for Q&A.

Finish your talk with “Thank You”.

Repeat your name. Tell people how they can get in touch with you.

Links

- [Why Emotional Intelligence Should Be In a Developer's Toolkit](#)
- [To Be More Likable and Make a Great First Impression, Science Says First Do 1 Thing](#)
- [What is Self-Awareness and Why is it Important? \[+5 Ways to Increase It\]](#)
- [Self-Regulation Lab](#)
- [Managing a Career](#)
- [What is a Pitch Deck?](#)
- [Top 10 Communication Skills](#)
- [What is Communication](#)
- [Emotional Intelligence](#)
- [Commit Messages Guide](#)
- [How to start a conversation and keep it going](#)
- [25 Common Competencies and Descriptions](#)
- [When writing a longer email put key takeaways in bold](#)

- [Edx Certificate - Delftx Leadership Essentials for Engineers](#)
- [How to prepare for technical interview at Uber, Google, Amazon, Facebook, ...](#)
- [Negotiation - Cooperation vs Confrontation](#)
- [Planning at Uber](#)
- [How to start an email to your boss](#)
- [SMART goals](#)
- [Niksen](#)
- [Executing on a goal](#)
- [Writing is an underestimated skill for engineers](#)
- [Collaboration isn't a technology problem](#)
- [Writing is Thinking](#)
- [An alternative approach to meeting agendas](#)
- [The Significance of an Email Signature](#)
- [Stackoverflow Developer Insights](#)
- [Competitive Coding vs Corporate Success](#)
- [Give and Take.](#)
- [Austin Geidt](#)
- [The Definitive Guide to OKRs](#)

Course Schedule

Classes are usually taught to groups of individuals that work at the same company or organization. The course spans a total of 12 hours of online training. The actual schedule of the course is determined by the group. Choices are:

- Two 6-hour sessions, held in two days.
- Two weekly 6-hour sessions.
- Four weekly 3-hour sessions.
- Six weekly 2-hour sessions. 👉 Recommended.

To schedule a course, or discuss more details, please contact the teacher.

Course Experience and Ratings

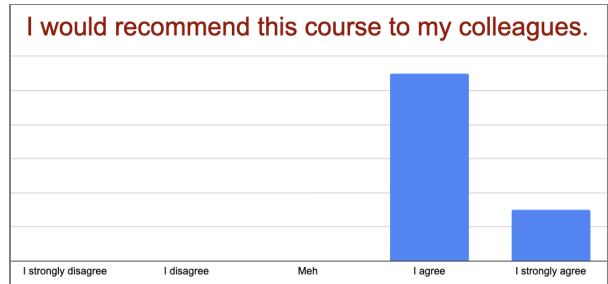
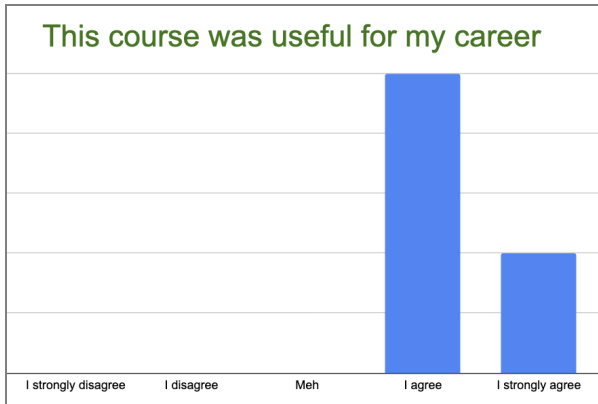
When asked what attendees learned from the course, answers included:

- Communication is not just talking to people, but it is happening all the time through different means. Becoming aware of that can help me improve my career.
- That I need to work on my spoken communication. I used words that can be replaced by better words. The "Very" word is something that I use a lot, and because I'm not a native speaker, it's hard to find the right translation. That's something that I will def keep an eye and try to improve.

- Have a brand for yourself.
- You are a brand. That it's important to think about yourself as a brand. Not only when you're planning to find a new job, but all the time.
- Communication is very important for engineers.
- Eye contact.
- Communication best-practices.
- Action items for docs, assign to other people to make them react.
- Interesting to see how my experience aligns with others.
- Marketing your name.
- That I do need to go back and read more.
- Communication is key.
- Life is better if one aims to make others happy.
- There are many areas where good communication skills can make a difference.
- Reading and writing good practices, Tada!
- Branding yourself can be fun.
- Emotion control.
- Zero inbox, help us to manage Emails better.
- Need to invest into human interactions and connections.
- It really stuck with me how I can do better in the process of code review.
- Writing is important!
- I should probably read and write more to get better.
- Keep it (your brand) simple.
- Know others.
- Tickets should be full of descriptions which can be shared by others.
- Need to work on my name brand more.
- Everything is connected. The way you reply to emails, write tickets, write documents, write code, comment on code reviews. Everything plays a role in your "brand" and in how people see you. Doesn't matter whom you're talking to in the "hierarchy chain", you need to be polite, objective and effective when communicating. Thinking about engineers (encoding, decoding, channels) make a lot of sense when thinking about communication.

Course Evaluation

Of all past attendees, 100% found the course useful for their career and would recommend to their team members:



Course Pricing

Course fees are €999 per person and include:

- Complimentary copy of the C4E book.
- Twelve hours of online training by an experienced teacher.
- Interactive exercises and assignments.
- Weekly reviews of assignments.
- Newsletters with tips and tricks after the course.

To schedule a course, or discuss more details, please contact [the teacher](#).