# Care and Feeding of a DARPA Project

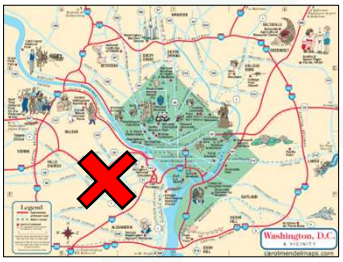
Irwin Reyes

February 26 2024

Hi I'm Irwin

twosix TECHNOLOGIES

twosix TECHNOLOGIES




**Principal** **Research** **Engineer**

2

I work at a medium-sized R&D firm called Two Six Technologies just outside of Washington DC. Government agencies like DARPA pay us to solve problems for government people.

I'm a Principal Research Engineer there, and I run Two Six's involvement in a larger DARPA effort spanning multiple companies.


How did I get here?

I've been at Two Six for 4.5 years now, half of which in a leadership position in charge of a team of ~6.

Before Two Six, I was an academic in California researching privacy violations in mobile apps.

Before that, I was a research engineer at Dell prototyping mobile authentication methods for two years.

Before that, I was a software engineer at Johns Hopkins Applied Physics Lab working on ballistic missile defense for three years. (definitely a big pivot coming from that!)

And before that, I was a college student just like you.


4

And before that, Allie and I were in the same section in high school marching band.

My speaking here has been in the works for a while.

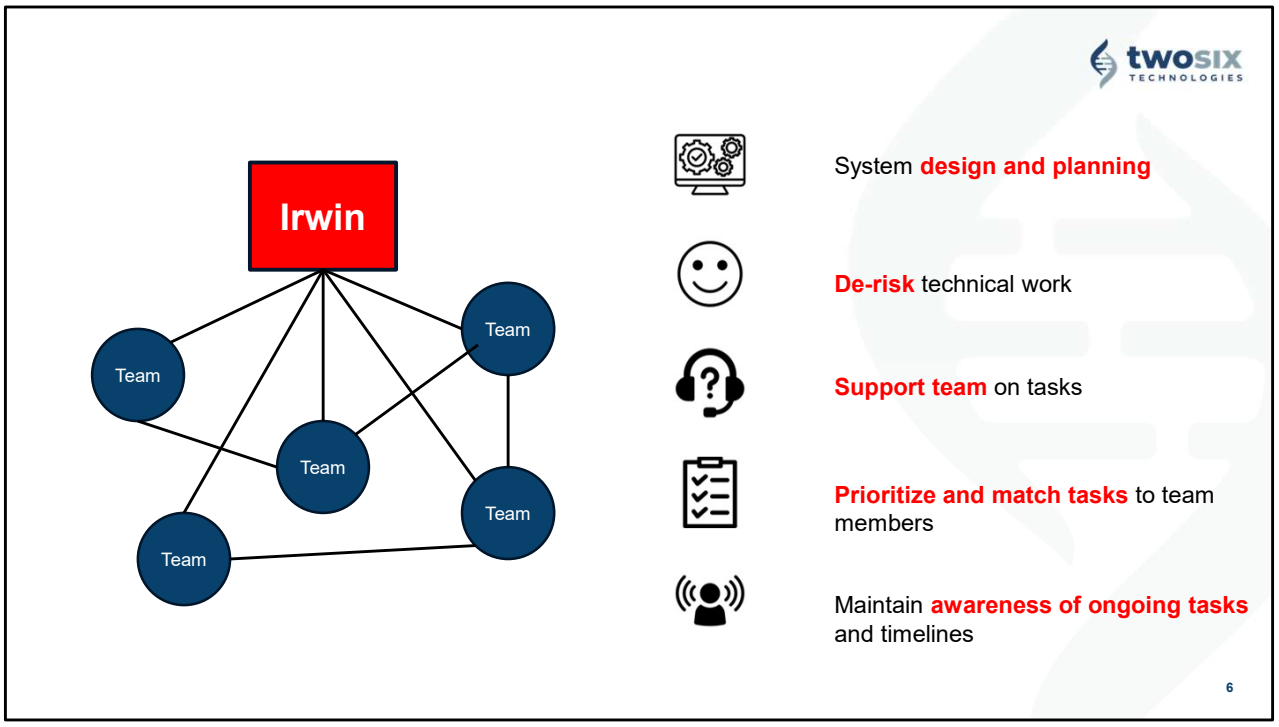

I mentioned that I run part of a large DARPA contract.

For background, DARPA is the Defense Advanced Research Projects Agency. They fund research for various defense and government purposes.

You use at least one of their projects. The ARPANET program kicked off in the late 1960s to connect computers at government, industry, and university laboratories. This laid the foundation for the modern Internet.

A few years ago, I submitted a proposal to DARPA for my team and I to research novel security and privacy methods for mobile devices.

DARPA selected us for funding, as part of a broader program with multiple companies pursuing that goal.

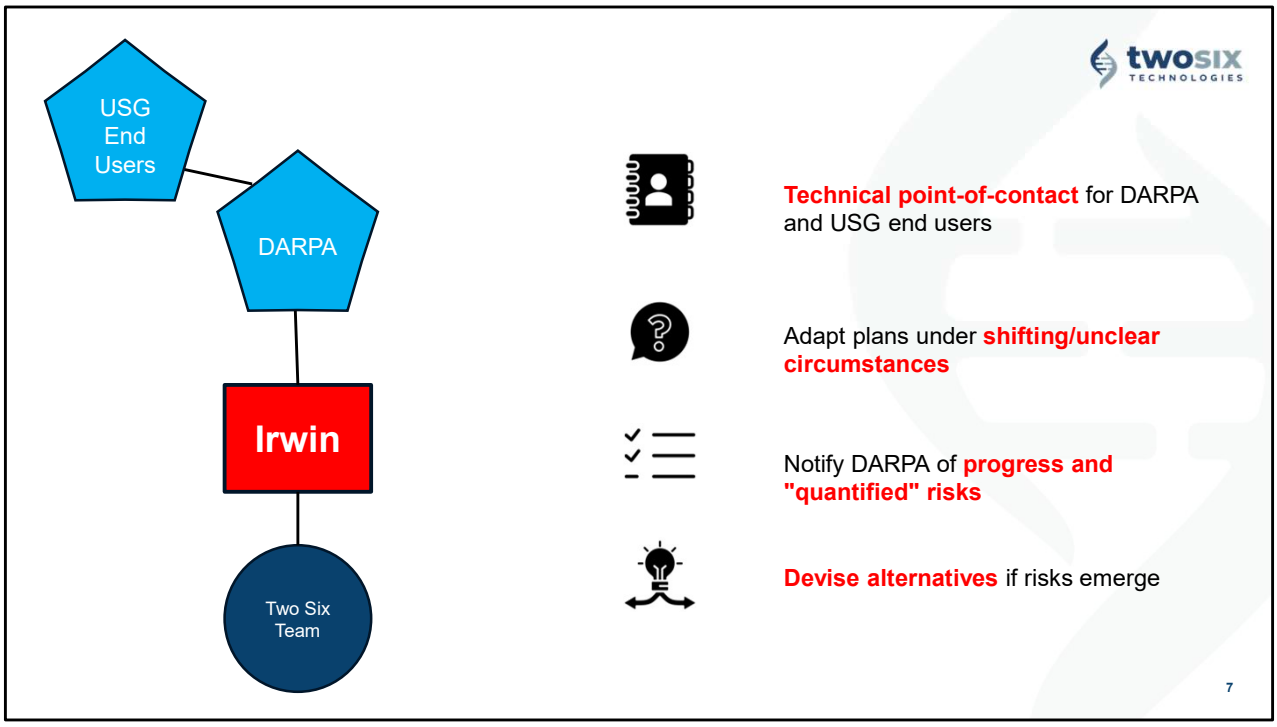



My position as a Principal Research Engineer requires me to juggle many roles and relationships in my DARPA program.

Day-to-day, managing my team takes up the bulk of my time.

This includes a lot of what you may have encountered in your class projects: software design (e.g., APIs), identifying tasks and assigning them in manageable chunks.

But it also includes a lot of non-technical "soft skills:" keeping people motivated, finding growth opportunities for folks, and making sure people have the support they need---whether from one another, from me, or from someone outside the team.

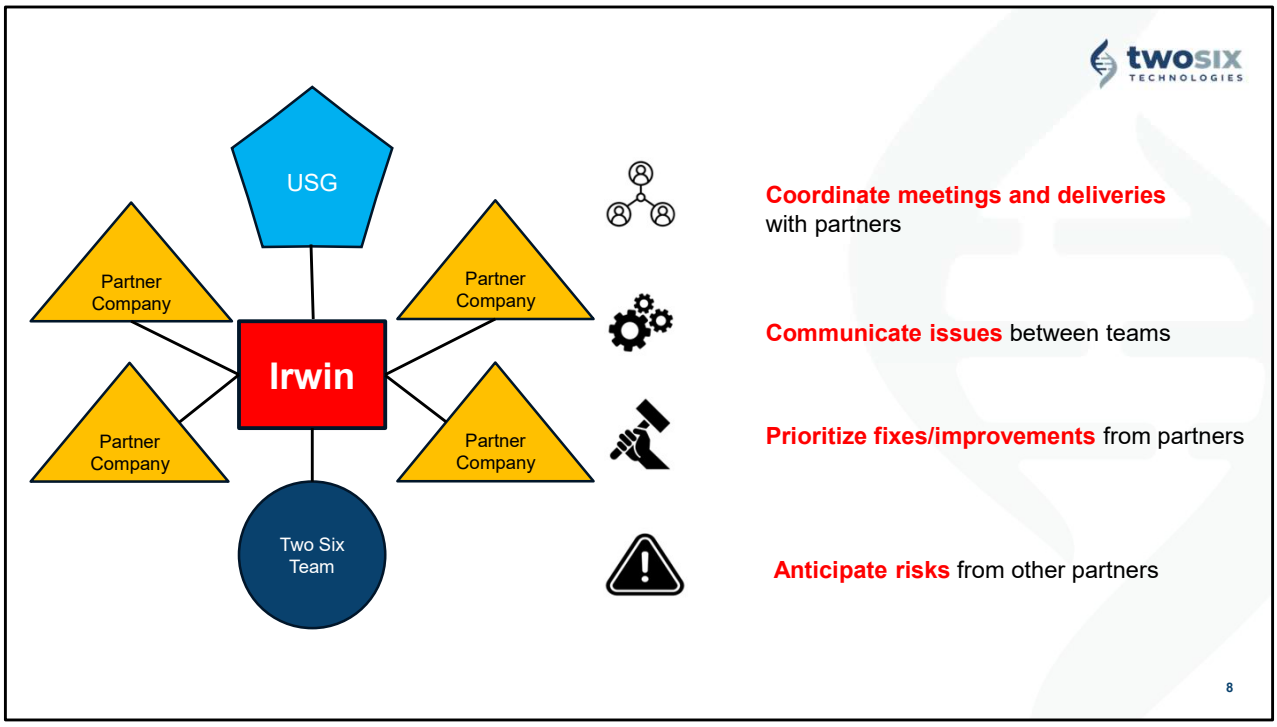



I may be a “team lead” but that doesn’t mean I’m on top of the totem pole. If anything, I’m right in the middle.

I’m the intermediary between my team, and our customer (DARPA and other US government people).

A huge part of my job is to make sure my team has clear priorities that line up with customer needs. Unclear or constantly shifting priorities are very bad for team morale.

Sometimes customers have unclear needs, or requirements with unknown trade-offs, or requests that need more time/effort than anticipated. I drill down into those details to avoid risk.

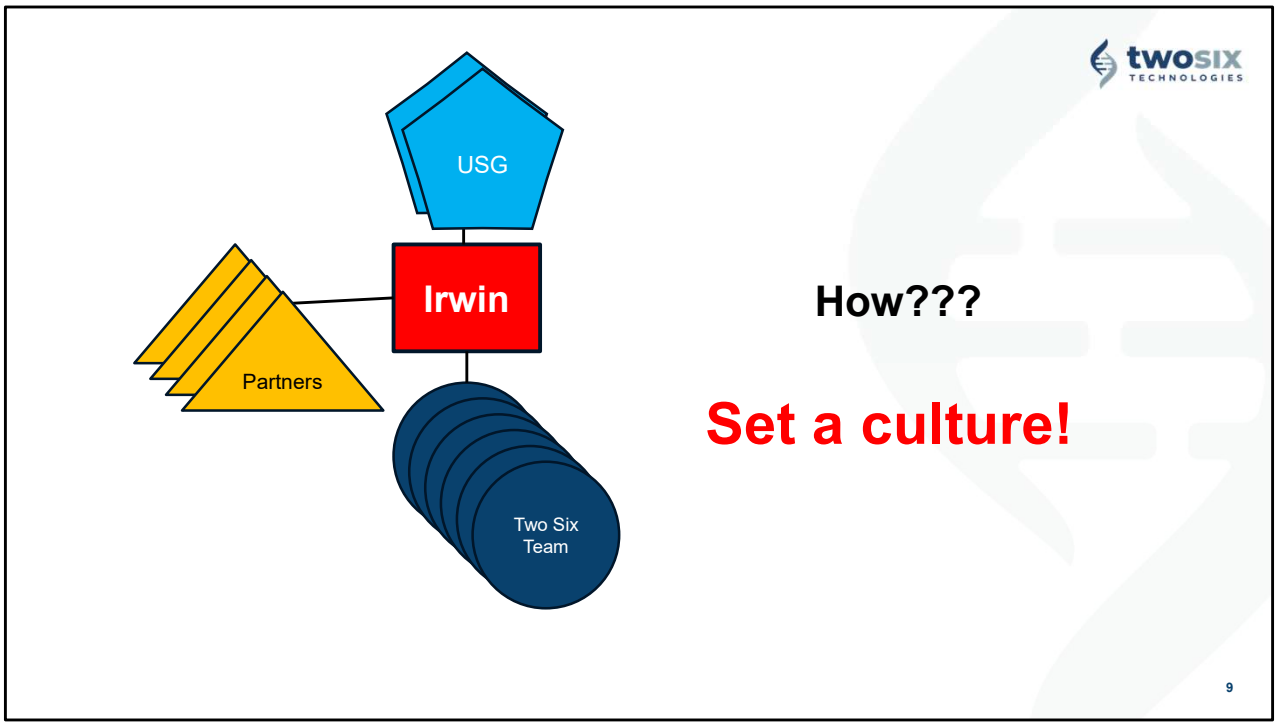



I don't just manage "up" and "down" the totem pole. I also manage the relationships among the other companies on the contract.

Our DARPA contract is structured so that each company contributes a major component to the program.

In my program, one company is responsible for novel hardware. Another builds machine learning models. Another works on user interface. My team is responsible for bringing it all together, testing everything for usability and validating correct function.

We keep everyone else honest and on the same page. This involves a lot of coordinating communications. If I sense that there might be a miscommunication or mismatched assumption somewhere, it's my responsibility to bring people together and clear it up.




I juggle many roles as part of my job.

All those roles may sound overwhelming.

But I handle it by boiling it down to one overarching task that guides me in all that: my job is to set a culture.

twosix TECHNOLOGIES

My team's culture:

→ Be boring, no surprises ←

10

The culture I try to set is quite simple.

Be boring, no surprises

Now, "boring" doesn't mean lack of interest, or low-effort, or strict adherence to initial ideas.

"Boring" in my team means rock solid reliability, so that we can do our job sustainably long-term with minimal self-inflicted stumbles.

It means that my team and I strive to deliver no surprises. The things we say we're going to do, we do. The risks we identify are real, but also reasonably well-understood with a variety of practical options for addressing them.


How do I set this culture? By creating an environment where people talk to one another so there are no surprises.

I take a “soft touch” approach for this. Instead of scheduling tons of meetings, I strive for presence and awareness.

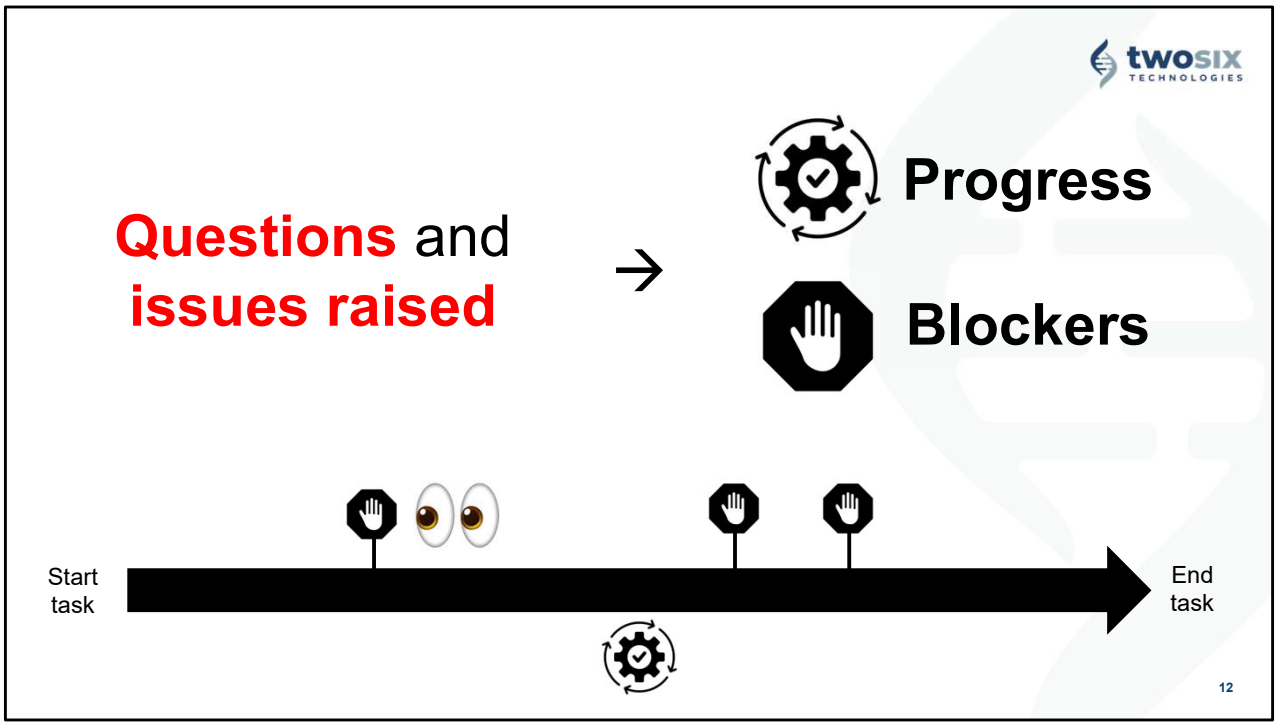
Although Two Six doesn’t have an official in-office policy (many folks are hybrid), I’m in the office for four or five days a week.

I keep an open door policy and encourage my team to stop by to talk about whatever whenever they’re around.

It doesn’t need to be just about work: if you overheard us, you’ll pick up on gym talk, trashy reality TV, cheesecakes, and puzzle games.

But those conversations are valuable because people will naturally bring up their work with a team that they feel a part of: what’s frustrating them, what they’re proud of, what they want to learn more about.

When they communicate clearly and openly and support one another, I'm able to put them in positions to support other relationships too: as experts for our DARPA/USG customers, or as technical leaders to talk to our partner companies.



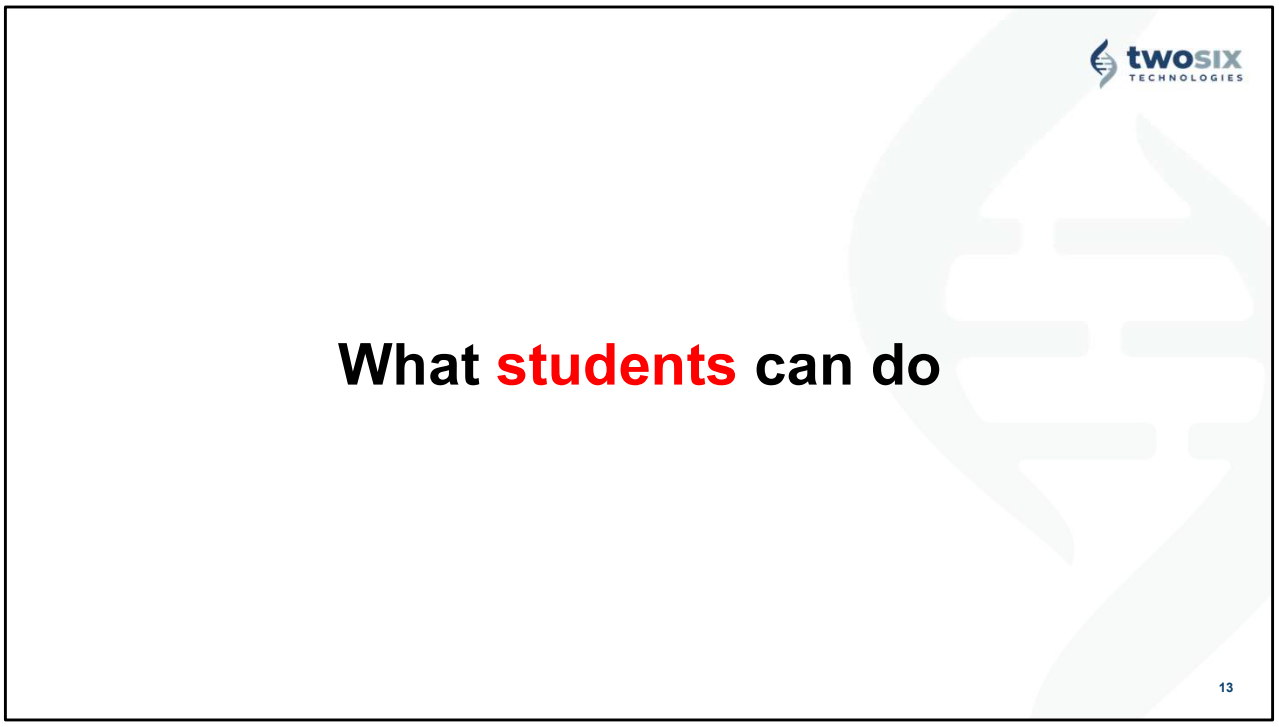

The questions and issues people raise about their work tell me a lot because those indicate progress and blockers.

When I divvy up tasks for the team, I keep a rough mental timer for how long it should take them to finish it.

That depends on the complexity and open-endedness of the task, and the teammate's strengths and interests.

I anticipate what challenges they might encounter and when. And check in on them if they don't ask questions when I expect them to.

This is how I maintain awareness of where everyone is in their work, all without being a mean guy---that wouldn't be fun for me, and it's not great for morale.

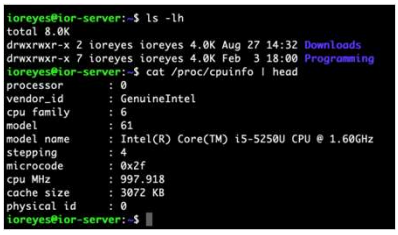



If my job sounds like a big firehose of things to do and keep track of, don't worry: you don't have to manage up, down, and side-to-side, or set a culture for a team for a long time.

I got to where I am gradually, over the course of 12 years of experience.

Whether you want to run a team someday or not, there are a couple things you can do now to set yourself up well to establish yourself in a software engineering team.

# #1 Learn the tools of the trade




## Command-line Linux

Basics: File navigation, text editing

Bonus: ssh remote access, git version control


## Git principles

Basics: `clone`, `pull`, `commit`, `push` operations

Bonus: branches, merge requests, code review

14

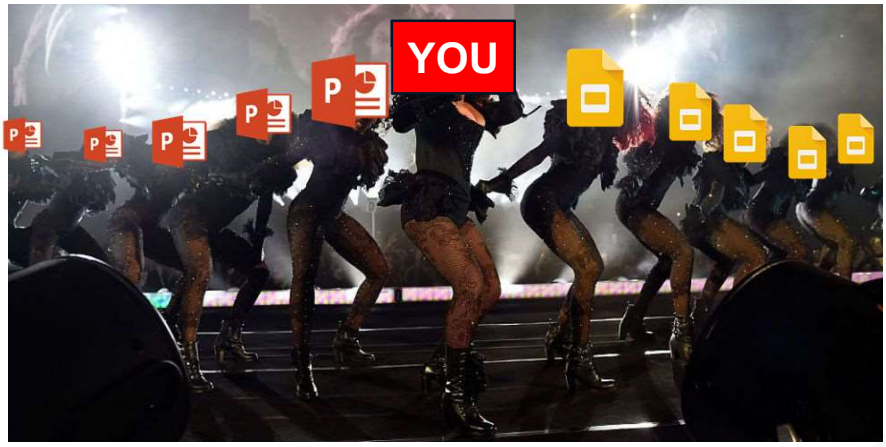
First, familiarize yourself with the tools of the trade.

In all the jobs I've occupied in industry and academia, a working knowledge of command-line Linux means you can "talk the talk" of any team you join. That makes learning your team's culture much easier.

If you can navigate files and folders and use a text editor, that puts you ahead of most new graduates coming out of school. Bonus points if you can use git version control and ssh remote access in the command line too.

Speaking of git, knowing how to use it to collaborate with others is huge. Simple code clone, pull, commit, and push operations are all you need day-to-day. Especially impressive if you know how to use branches, submit merge requests, and review code with your team---these help teammates stay aware of what everyone else is doing.

## #2 Hone your presentation skills




**What questions** do you want to get?

15

Second, communicate well. I know Allie's been going over presentation skills in this class.

In my team, I always try to find opportunities for my junior teammates to speak in front of an audience, whether internally to the company, or to external collaborators and customers.

Those are great for building up their confidence and establishing their reputation as experts.

I give them two tips when preparing.

You're Beyonce. Your slides are backup dancers. Backup dancers never upstage or distract from Beyonce. Your slides should similarly support and draw attention to you.

Ask: "What questions do you want to get after your presentation?" This keeps them focused on speaking to the audience's priorities and holding their attention.

# #3 Ask and answer questions

As a **student/intern**:

- **Ask questions**
- Get comfortable **interacting with experienced people** (e.g., professors, senior colleagues, management)

As a **new hire**:

- Get **immersed in the team's culture**
- **Continue asking questions** and learning

As a **junior team member** (i.e., months of experience)

- Become the "**go-to guy/gal**" for something
- Offer answers and **guidance to interns and new hires**

16

And dovetailing on that last point on communication, don't be afraid to ask and answer questions.

If you're a student or intern, your #1 job is to learn. You should be asking all sorts of questions about everything from your team's technologies and processes, to where to get a good cheap lunch near the office.

As a new hire, you should continue learning and connecting with your team. Really pick up on the culture and make it work for you and your growth.

Before you know it, a new batch of interns is starting, and they'll be asking about how merge requests work or where to get a good sandwich. And you'll be in a position to continue and contribute to your team's culture.

## Learn more

My public research: irwinreyes.com

**Ask me anything!**

- Technologies used
- Hiring/interviewing
- Remote work
- Work/life balance
- Usable security/privacy
- Missiles→mobile pivot
- Senior vs. junior tasking
- Non-management paths

17

And with that, I'll wrap up this talk and take questions.

I tried to avoid getting mired up in nitpicky specifics, so feel free to ask me for more details about anything I mentioned.

If you'd like to learn more about some of the stuff I've been involved with over the years, feel free to visit these links. Thanks so much for your attention.