Influence: Lessons from a Staff Engineer and ex-Twitter 8-year Tech Lead

Lessons on influence, managing up, and working with difficult coworkers from 8-year tech lead & Staff Engineer, Ron DeVera

JORDAN CUTLER
SEP 24, 2023

92 26 Share

Hey everyone, Jordan here 👋. Before jumping in, I want to thank you for the growth of the newsletter. In just 1 week, we went from 10k subscribers to 13k subscribers. It makes me so happy to know you are finding so much value from the newsletter.

👋 Intro to Ron

This week's issue is incredibly special. We take an inside look at how a seasoned Staff Engineer works and thinks.

I interviewed Ron DeVera, who worked as a Staff Engineer at Twitter, co-created the Feather design system, and served as its tech lead for 8 years.

Ron is also now my tech lead at Qualified and is a Staff Engineer here.

What makes Ron special?

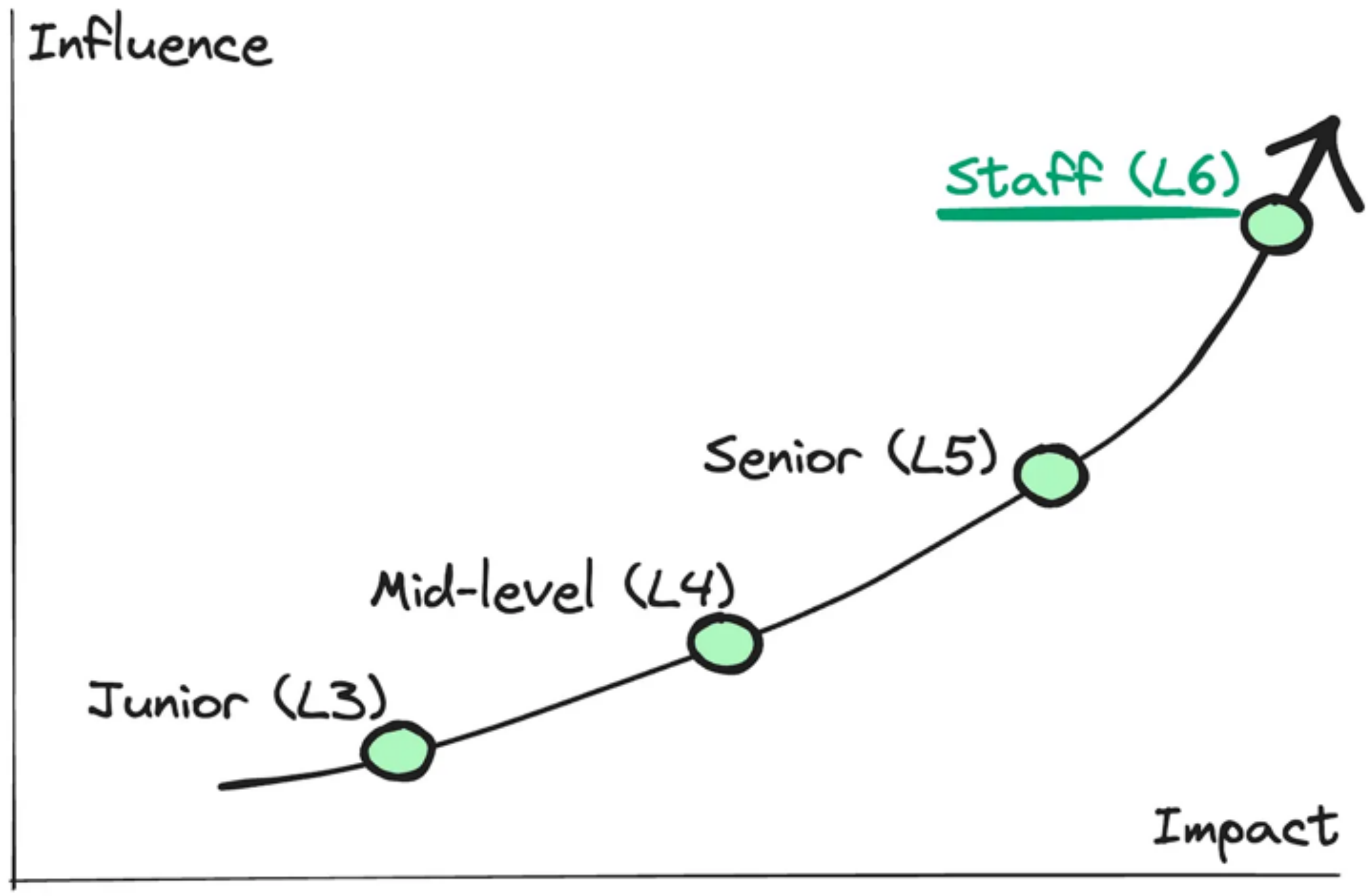
In just 3 months since joining Qualified, Ron has:

- Gained a deep amount of trust from management and all team members
- Influenced in and outside of the team to move important initiatives forward

- Shipped high-impact changes that impact the whole organization

His communication and core skills stood out to me as the best I've ever seen.

Quick terminology: *In big tech, a Staff Engineer translates to L6, which is 1 level above Senior.*



As you level up, more of your impact comes through influence rather than code.

⚠️ ***Disclaimer:*** *For brevity of the article, I've paraphrased the discussion. The wording retains the tone and essence of what was said.*

Also, while I have heavily condensed this down, it's in a more interview-style format which results in a longer than usual but more personal format.

If you'd like to jump straight to the TL;DR and takeaways at any point, see the bottom.

⭐ Main takeaways

- How to influence and think like a Staff Engineer
- How to “manage your manager” effectively and handle disagreements
- What differentiates a Staff Engineer from a Senior Engineer

🤝 Influence

Joining a new team

Jordan: In just 3 months since onboarding, you gained a lot of trust, influenced direction, and made high-impact changes. How do you think about onboarding at a new company, gaining influence and building relationships quickly, and getting up to speed on the way things are done?

Ron: When I join a new team or company, I go straight into curiosity mode. About the tech, the people, and the history of the company. I think back to my experiences that put me in the shoes of the people and the company I’m at so I can relate to their problems as deeply as possible. When I chat with everyone, particularly executives, I try to imagine and ask what are the things that are really stressing them out right now.

I also like to join social channels like virtual coffee, cats, dogs, cooking, and try to find out what people are interested in. That and 1:1s with people as I join to build up relationships and have a working rapport.

Finally, **I constantly remind myself I’m here with fresh eyes.** I ask a ton of questions about things that don’t fully make sense to me, but also recognize the struggles that it might have taken to even get it working in the first place.

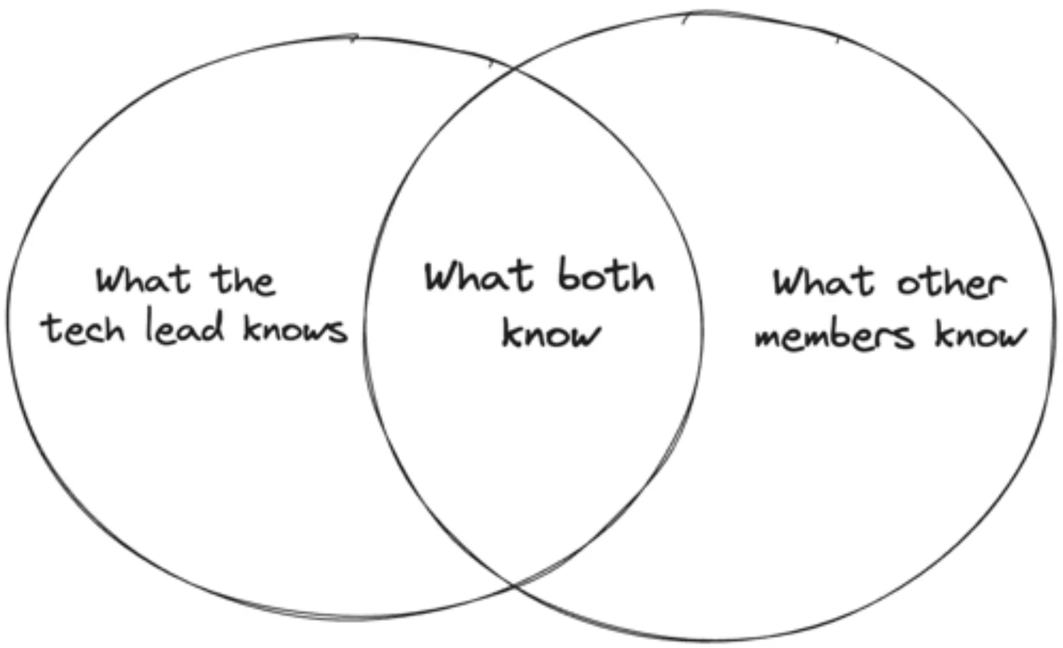
Making suggestions and influencing outcomes

Jordan: Getting into that more, how exactly do you approach asking your questions and making suggestions?

Ron: Maintain that curiosity. I don't like to just put my suggestions and ideas onto other people. I usually ask, "What do you think about { my suggestion }." When I ask, "What do you think?" I really do want to hear what the other person wants to hear about the idea. Especially in a situation of being new on a team, there's often a lot of missing context. Hearing the other person's perspective on it helps uncover reasons for why things were done a certain way and what the best path is moving forward.

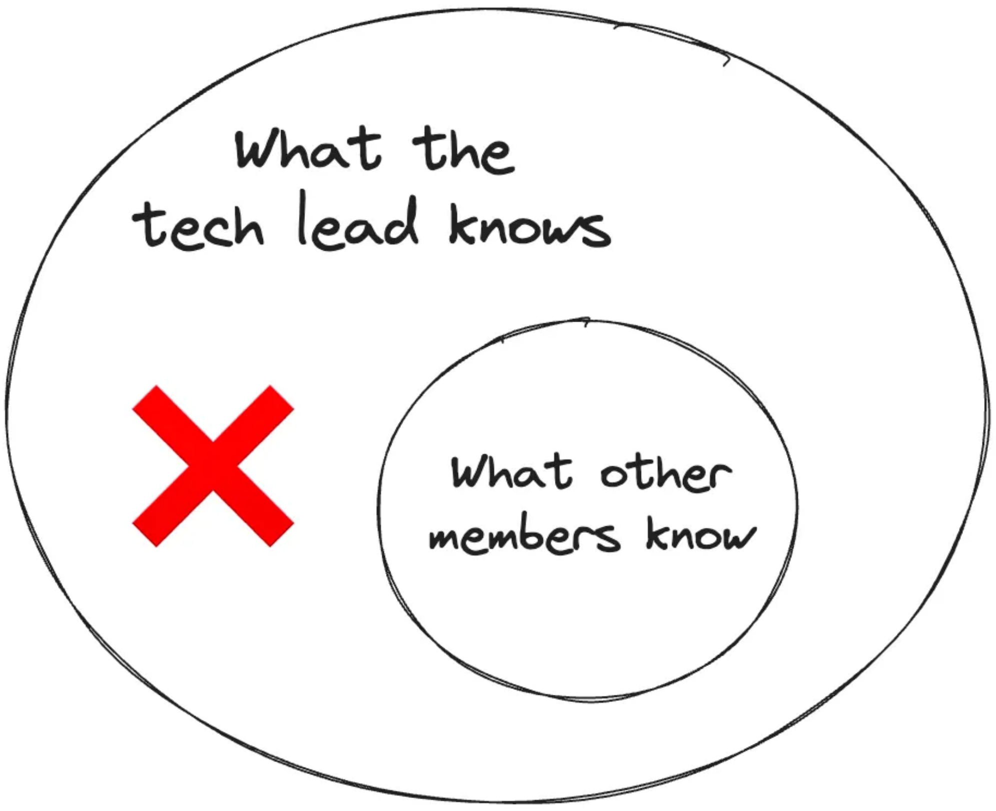
What's important to me as a tech lead is not falling into the trap of thinking that I'm completely right. There are only a few areas where I feel the need to put my foot down—like legal or security reasons.

There's generally a Venn diagram of knowledge, with things I don't know that I need to figure out before I make my suggestion.



The mindset of how you should approach suggestions you bring. Understand there's missing knowledge you need to uncover.

As a tech lead, you shouldn't believe your knowledge is a superset of others. It can lead to toxic environments.



What you shouldn't assume as a tech lead ❌

Jordan: How does the way you make suggestions change depending on your relationship with the person?

Ron: If I'm making a suggestion to someone I haven't interacted with before, it's completely different. I really make sure that I've done my research and that it's clear to the other person.

1. **Assume the best intent:** I assume the best intent and knowledge in the other team and person.
2. **Do research:** I do research under that assumption to see why something may be a certain way before reaching out.
    a. Many times the cause might be a tool being used, rather than the people.
        i. **For example:** You may see accessibility problems on your company's marketing site, and it could be because the tool used to generate that site does

a poor job of helping with accessibility, rather than the people blatantly ignoring that.

   b. The cause might also be a constraint at the time.

      i. **For example:** You notice some low-performance code that was added because of a feature that needed to be shipped on a tight deadline.

3. **Determine if it's worth it:** Depending on how much you care, how important the issue is, and how important you think the issue may be to them, you can decide if it's worth bringing up to them. Review your priorities constantly depending on timing.
4. **Find what you're missing:** Gather information from them on if it's on their radar, something like: "Hey team, { introduce yourself }, I noticed this issue here. I dug into X but hit a roadblock. Do you happen to know more about this?"
   a. In this, you may realize the team has a massive relaunch they're working on and just don't have the bandwidth to take on your suggestion or that it's all going to change in the coming weeks anyway.
5. **Frame it to their goals:** For example, if you want to suggest accessibility fixes to a marketing website team, you could frame the accessibility fixes as a way to improve SEO through their lighthouse score.
6. **Make it easy:** Based on their answer, make it easy for them to do your suggestion. You could provide a code example or links to documentation or resources you found. Or offer to help with it yourself.

Jordan: In this approach, you end up taking on a lot of upfront work yourself before even reaching out to the other team. How do you balance that with all your normal responsibilities?

Ron: You definitely could go directly to Slack with a question. At least for me though, I always appreciate when someone comes to me with a question that is very clearly well-researched. When someone does this, I'm way more inclined to help continue their effort.

Many times I will see something that can be improved, but I'll save it in my to-do app

because it's not the right time, or I want to do more research on it first.

😬 Working with difficult coworkers

Jordan: How would you handle working with a difficult coworker? Like someone that takes very long to respond or review your PRs for example?

Ron: One thing that really helps with this is building a rapport through recurring 1:1s in the first place. Otherwise, you end up needing to schedule one-offs that are a bit more awkward.

I like to use the **Situation-Behavior-Impact** framework. I tell them what's happening and the **impact it has on me, our teammates, or other people**.

For example: "Sometimes it takes a few days for me to hear back from you. The impact it has on me is it slows down my project. I'm not able to get feedback quickly enough and when I do, after making the change, it takes a little while longer to hear back from you, and I can't tell whether you've actually read it or if you have some other issue that's going on where you don't have time. And if you don't have time, I'd certainly love to know that too, so I could adjust my schedule or have someone else review it."

Again, assuming the best intent helps here because it informs your tone as you share the feedback.

Usually, it's an abbreviated version of situation-behavior-impact, but if it's repeated or becomes a larger issue, then I follow that format more strictly.

Jordan: How do you approach entering the conversation so it isn't super awkward?

Ron: It depends on the type of relationship you have. If it's a recurring 1:1, I always try to start it off with a casual catch-up before getting into the core of the 1:1. If I sense the other person just wants to get straight into it though, then I'll adjust and share the feedback right away.

If it is a one-off 1:1, I usually tell them ahead of time what I'd like to talk with them about. **Surprising someone with feedback can lead to them feeling ambushed.**

Handling repeated issues

Jordan: What if they still don't change after the feedback?

Ron: It varies based on the person. If the person was pleasant the first time around, I try to let them know a second time it's still happening and the impact it has on me or others.

If the person doesn't seem open to feedback, I usually go to my manager. **However, I try to only go to my manager as a last resort. Having your manager give the feedback could lead to a damaged relationship.**

Jordan: How do you approach the conversation with your manager?

Ron: Funny enough, I use **situation-behavior-impact** for that too. I tell them the situation of what's been happening, the behavior of the person, and the impact it's had on me or others. I also tell them the steps I took to try to resolve it on my own. Finally, I ask if they could try to be sensitive to the relationship as they look into this, just to help avoid a damaged relationship between myself and the other person.

☝️Managing up

Jordan: What is the mindset you have when you think about discussing topics with your manager—whether it's a problem in the team, a process that should be changed, discussing promotion, or anything else?

Ron: One thing I try to avoid is asking for too many small things and things that I could probably solve myself if I put a bit of time into it. Because of that, when I do ask for things, I usually find my manager is more receptive since I'm only asking when something is truly important.

Another thing is just recognizing that they have so much going on. Before the conversation, I think about the number of people they're trying to manage, the different things they might have going on—like security audits, hiring, funding, infrastructure costs, etc.

Jordan: Ah, so getting into that mindset helps you come with a more understanding and empathetic tone? When you do ask for something, how do you approach it?

Ron: Getting in that mindset first helps a lot. When I do ask, I might say something like, "Here's something that's going on. Do you have ideas on how to fix this?" Then I might list 1 thing I've done to try solving it on my own first.

What you shouldn't do is say, "Here's a problem. These are 10 things I've tried. Now what do I do next?" This just adds more work for them to process everything. Let them make suggestions too since they might suggest something you haven't thought about yet.

One thing to keep in mind is that your manager is just another person with another role. Treat them like a person, not someone you just want something from. Build a relationship, ask about their hobbies and interests, and get to know them.

🤔 Staff vs. Senior Engineer

Jordan: What do you think is a common quality of all Staff engineers that differentiates them from Senior engineers?

Ron: Staff engineers are very impactful. The impact can come in different ways like:

- Changing the way the team works
- Improving some core infrastructure
- Delivering over and over again really fantastic products

It has to be repeated. It can't just be a one-time thing.

It also means you need to be able to prioritize constantly.

Ron and I discussed that Senior engineers show these traits sometimes. The main difference between the Senior and Staff level is how consistently you can show these and how impactful the work is that you're doing.

📖 TL;DR

- **Influence**
  - Avoid directly pushing suggestions too much. Generally, come to your suggestions through upfront research and empathy toward what has been done before. Only then, ask "What do you think about this approach?"
- **Working with difficult coworkers**
  - Use Situation-Behavior-Impact to deliver feedback
  - Give the feedback in the right way and at the right time depending on the relationship.
- **Managing up**
  - Avoid asking for too many small things. When you do ask for something, it will be more impactful.
  - Before asking for something, recognize all the other stressors they may have going on. It will change how you approach the conversation.
  - Your manager is a person too. Just with different responsibilities. Get to know them and their interests.
- **Staff vs. Senior engineer**
  - Staff engineers **repeatedly** do one or more of:
    - Change the way the team works
    - Improving core infrastructure or patterns
    - Deliver fantastic products

  - They are also masters of prioritization, balancing quick fixes and improvements with longer-term initiatives.

💭 Closing thought

There were **two** major threads that kept coming up in almost every way Ron approaches situations.

1. **Understand the situation from the other person's perspective**
    a. In the onboarding topic, he talked about focusing on everyone's challenges and identifying how he can help the most.
    b. When making suggestions, he focuses on the timing and priorities of the other person, as well as why they made the decisions that they did; instead of pushing his suggestion on the other person.
    c. When managing up, he takes into account his manager has a ton going on, so he focuses on solving problems for them. If he does need to go to them, he tries to make it as easy as possible for them.
2. **Assume good intent and that you are missing some context.**
    a. When making suggestions, he tries to find out what he doesn't know that the other person does and assumes they had a good reason for doing it.
    b. When dealing with difficult situations with coworkers, he also goes into it assuming the best intent. The person may not be responding quickly because they have 20+ other requests they are managing. Going into it with that mindset helps maintain a positive tone rather than a stern one.

💡Ron's recommended resources

Ron attributes a portion of his skills to learning about management.

- He recommended the Rands in Repose blog as a resource that was useful to him.

- He also recommended the Soft Skills Engineering podcast (I briefly checked it out and it seems amazing).
- For books, I'd recommend Making of a Manager or The Manager's Path.
- For newsletters, I'd recommend Leading Developers or Hybrid Hacker.

🙏 Thank you to Ron

A big thank you 🙏 to Ron for sharing his insights and being an overall great human. It's obvious he truly cares about everyone he works with, building great relationships, and deeply considering their opinions before his own.

If you'd like to find him, this is his website.

📣 Shoutouts of the week

Two YouTube videos went out this week that featured me. So I want to give a shout-out to the creators for having me on and taking the time to put the video together.

1. Huzaifa Asif - Personal and Career Development Podcast
2. Typo team and Kshitij Mohan - Metrics strategies and burnout prevention

Finally, one more shout-out to this article:

- Josh Comeau graciously gave me access to his new Joy of React course (not sponsored) in exchange for an honest review. I'm not all the way through it, but from what I've gone through, I can tell this is the course I wish I had when I first learned React. If you have an L&D budget, I'd recommend checking it out: https://www.joyofreact.com/. Also, his CSS for JS developers course I took in full and it was the best investment I made into my career.
    - He also gave me a 10% coupon code for all of you. Use "CUTLER" at checkout.

As always, thank you for reading and for the growth to 13k+ subscribers.

- Jordan

P.S. If you're interested, I'm accepting the following:

1. *New coaching clients: See [Mentorcruise for rates]*
2. *Newsletter sponsorships: Feel free to reply to this email to reach me.*

Did you find this issue valuable? If so, there are two ways you can help:

You can also hit the like ❤️ button at the bottom of this email to help support me. It really helps!

92 Likes · 6 Restacks

26 Comments

Write a comment...

Anton Zaides Writes Leading Developers Sep 24 ❤ Liked by Jordan Cutler

So many gems!!

I loved that part: "At least for me though, I always appreciate when someone comes to me with a question that is very clearly well-researched. When someone does this, I'm way more inclined to help continue their effort."

As you mentioned - it takes a lot of preparation on your side, but is definitely worth it!

And I appreciated this one a lot, as a manager: "Before the conversation, I think about the number of people they're trying to manage, the different things they might have going on". When people come to me and SEE me, acknowledging my difficulties, it makes me much more inclined to go out of my way to help them.

And thanks for the shoutout! 🙏

P.S. Ron, please come work for us! 😂

LIKE (2) REPLY SHARE

1 reply by Jordan Cutler

Tiger Abrodi Writes Saiyan Growth Letter Sep 24 ❤ Liked by Jordan Cutler

Assuming good intent is so important.

Even if the other person I'm actuality doesn't have good intents, it could change them as well!

LIKE (2) REPLY SHARE

1 reply by Jordan Cutler

24 more comments...