Drunk Post: Things I've learned as a Sr Engineer

I'm drunk and I'll probably regret this, but here's a drunken rank of things I've learned as an engineer for the past 10 years.

- The best way I've advanced my career is by changing companies.

- Technology stacks don't really matter because there are like 15 basic patterns of software engineering in my field that apply. I work in data so it's not going to be the same as webdev or embedded. But all fields have about 10-20 core principles and the tech stack is just trying to make those things easier, so don't fret overit.

- There's a reason why people recommend job hunting. If I'm unsatisfied at a job, it's probably time to move on.

- I've made some good, lifelong friends at companies I've worked with. I don't need to make that a requirement of every place I work. I've been perfectly happy working at places where I didn't form friendships with my coworkers and I've been unhappy at places where I made some great friends.

- I've learned to be honest with my manager. Not too honest, but honest enough where I can be authentic at work. What's the worse that can happen? He fire me? I'll just pick up a new job in 2 weeks.

- If I'm awaken at 2am from being on-call for more than once per quarter, then something is seriously wrong and I will either fix it or quit.

- pour another glass

- Qualities of a good manager share a lot of qualities of a good engineer.

- When I first started, I was enamored with technology and programming and computer science. I'm over it.

- Good code is code that can be understood by a junior engineer. Great code can be understood by a first year CS freshman. The best code is no code at all.

- The most underrated skill to learn as an engineer is how to document. Fuck, someone please teach me how to write good documentation. Seriously, if there's any recommendations, I'd seriously pay for a course (like probably a lot of money, maybe 1k for a course if it guaranteed that I could write good docs.)

- Related to above, writing good proposals for changes is a great skill.

- Almost every holy war out there (vim vs emacs, mac vs linux, whatever) doesn't matter... except one. See below.

- The older I get, the more I appreciate dynamic languages. Fuck, I said it. Fight me.

- If I ever find myself thinking I'm the smartest person in the room, it's time to leave.

- I don't know why full stack webdevs are paid so poorly. No really, they should be paid like half a mil a year just base salary. Fuck they have to understand both front end AND back end AND how different browsers work AND networking AND databases AND caching AND differences between web and mobile AND omg what the fuck there's another framework out there that companies want to use? Seriously, why are webdevs paid so little.

- We should hire more interns, they're awesome. Those energetic little fucks with their ideas. Even better when they can question or criticize something. I love interns.

- *sip*

- Don't meet your heroes. I paid 5k to take a course by one of my heroes. He's a brilliant man, but at the end of it I realized that he's making it up as he goes along like the rest of us.

- Tech stack matters. OK I just said tech stack doesn't matter, but hear me out. If you hear Python dev vs C++ dev, you think very different things, right? That's because certain tools are really good at certain jobs. If you're not sure what you want to do, just do Java. It's a shitty programming language that's good at almost everything.

- The greatest programming language ever is lisp. I should learn lisp.

- For beginners, the most lucrative programming language to learn is SQL. Fuck all other languages. If you know SQL and nothing else, you can make bank. Payroll specialtist? Maybe 50k. Payroll specialist who knows SQL? 90k. Average joe with organizational skills at big corp? $40k. Average joe with organizations skills AND sql? Call yourself a PM and earn 150k.

- Tests are important but TDD is a damn cult.

- Cushy government jobs are not what they are cracked up to be, at least for early to mid-career engineers. Sure, $120k + bennies + pension sound great, but you'll be selling your soul to work on esoteric proprietary technology. Much respect to government workers but seriously there's a reason why the median age for engineers at those places is 50+. Advice does not apply to government contractors.

- Third party recruiters are leeches. However, if you find a good one, seriously develop a good relationship with them. They can help bootstrap your career. How do you know if you have a good one? If they've been a third party recruiter for more than 3 years, they're probably bad. The good ones typically become recruiters are large companies.

- Options are worthless or can make you a millionaire. They're probably worthless unless the headcount of engineering is more than 100. Then *maybe* they are worth something within this decade.

- Work from home is the tits. But lack of whiteboarding sucks.

- I've never worked at FAANG so I don't know what I'm missing. But I've hired (and not hired) engineers from FAANGs and they don't know what they're doing either.

- My self worth is not a function of or correlated with my total compensation. Capitalism is a poor way to determine self-worth.

- Managers have less power than you think. Way less power. If you ever thing, why doesn't Manager XYZ fire somebody, it's because they can't.

- Titles mostly don't matter. Principal Distinguished Staff Lead Engineer from Whatever Company, whatever. What did you do and what did you accomplish. That's all people care about.

- Speaking of titles: early in your career, title changes up are nice. Junior to Mid. Mid to Senior. Senior to Lead. Later in your career, title changes *down* are nice. That way, you can get the same compensation but then get an increase when you're promoted. In other words, early in your career (<10 years), title changes UP are good because it lets you grow your skills and responsibilities. Later, title changes down are nice because it lets you grow your salary.

- Max out our 401ks.

- Be kind to everyone. Not because it'll help your career (it will), but because being kind is rewarding by itself.

- If I didn't learn something from the junior engineer or intern this past month, I wasn't paying attention.

- Oops I'm out of wine.

- Paying for classes, books, conferences is worth it. I've done a few conferences, a few 1.5k courses, many books, and a subscription. Worth it. This way, I can better pretend what I'm doing.

- Seriously, why aren't webdevs paid more? They know everything!!!

- Carpal tunnel and back problems are no joke. Spend the 1k now on good equipment.

- The smartest man I've every worked for was a Math PhD. I've learned so much from that guy. I hope he's doing well.

- Once, in high school, there was thing girl who was a great friend of mine. I mean we talked and hung out and shared a lot of personal stuff over a few years. Then there was a rumor that I liked her or that we were going out or whatever. She didn't take that too well so she started to ignore me. That didn't feel too good. I guess this would be the modern equivalent to "ghosting". I don't wish her any ill will though, and I hope she's doing great. I'm sorry I didn't handle that better.

- I had a girlfriend in 8th grade that I didn't want to break up with even though I didn't like her anymore so I just started to ignore her. That was so fucked up. I'm sorry, Lena.

- You know what the best part of being a software engineer is? You can meet and talk to people who think like you. Not necessarily the same interests like sports and TV shows and stuff. But they think about problems the same way you think of them. That's pretty cool.

- There's not enough women in technology. What a fucked up industry. That needs to change. I've been trying to be more encouraging and helpful to the women engineers in our org, but I don't know what else to do.

- Same with black engineers. What the hell?

- I've never really started hating a language or technology until I started becoming intimately familiar with it. Also, I think a piece of tech is good if I hate it but I simultaneously would recommend it to a client. Fuck Jenkins but man I don't think I would be commuting software malpractice by recommending it to a new client.

- That being said, git is awful and I have choice but to use it. Also, GUI git tools can go to hell, give me the command line any day. There's like 7 command lines to memorize, everything else can be googled.

- Since I work in data, I'm going to give a data-specific lessons learned. Fuck pandas.

- My job is easier because I have semi-technical analysts on my team. Semi-technical because they know programming but not software engineering. This is a blessing because if something doesn't make sense to them, it means that it was probably badly designed. I love the analysts on the team; they've helped me grow so much more than the most brilliant engineers.

- Dark mode is great until you're forced to use light mode (webpage or an unsupported app). That's why I use light mode.

- I know enough about security to know that I don't know shit about security.

- Crap I'm out of wine.

- Being a good engineer means knowing best practices. Being a senior engineer means knowing when to break best practices.

- If people are trying to assign blame to a bug or outage, it's time to move on.

- A lot of progressive companies, especially startups, talk about bringing your "authentic self". Well what if your authentic self is all about watching porn? Yeah, it's healthy to keep a barrier between your work and personal life.

- I love drinking with my co-workers during happy hour. I'd rather spend time with kids, family, or friends.

- The best demonstration of great leadership is when my leader took the fall for a mistake that was 100% my fault. You better believe I would've walked over fire for her.

- On the same token, the best leaders I've been privileged to work under did their best to both advocate for my opinions and also explain to me other opinions 'that conflict with mine. I'm working hard to be like them.

- Fuck side projects. If you love doing them, great! Even if I had the time to do side-projects, I'm too damn busy writing drunken posts on reddit

- Algorithms and data strictures are important--to a point. I don't see pharmacist interviews test trivia about organic chemistry. There's something fucked with our industry's interview process.

- Damn, those devops guys and gals are f'ing smart. At least those mofos get paid though.

- It's not important to do what I like. It's more important to do what I don't hate.

- The closer I am to the product, the closer I am to driving revnue, the more I feel valued regardless of how technical my work is. This has been true for even the most progressive companies.

- Linux is important even when I was working in all Windows. Why? Because I eventually worked in Linux. So happy for those weekend where I screwed around installing Arch.

- I've learned to be wary for ambiguous buzz words like big data. WTF is "big" data? I've dealt with 10k rows streaming every 10 minutes in Spark and Kafka and dealt with 1B rows batched up hourly in Python and MySQL. Those labels can go fuck themselves.

- Not all great jobs are in Silicon Valley. But a lot are.

Finally, if you really want to hurt me, don't downvote I don't care about that. Just ignore this post. Nothing makes me sadder than when I wrote a long post and then nobody responds. So if you hate this post, just ignore.