EPISODE 352

[INTRODUCTION]

[0:00:00.4] JM: Engineers have plenty to be skeptical about. We look to datasets to give us something resembling objective truth. Some areas of research have so many variables that it's hard to isolate fact. Kyle Polich hosts the popular data science show, Data Skeptic, where he examines problems and solutions around data and he's one of the guests today in our Roundtable Discussion.

There are many big unanswered questions in our world that might eventually be solved with enough data and the right scientific approach. Some of those areas are nutrition, and drug discovery, and image classification. The hiring process is also like this. How can you predict whether an engineer will make a good hire?

Ammon Bartram of Triplebyte is working on solving the hiring process for engineering organizations and he is the other guest for today's Roundtable Discussion. Kyle, and Ammon, and I talk about a variety of subjects related to hiring and data and skepticism. I really enjoyed talking to both of them.

Software Engineering Daily is looking for sponsors for Q3. If your company has a product or service or if you're hiring, Software Engineering Daily reaches 23,000 developers listening daily. You can send me an email, jeff@softwareengineeringdaily.com if you're interested. Thanks for listening.

[SPONSOR MESSAGE]

[0:01:33.4] JM: Artificial intelligence is dramatically evolving the way that our world works, and to make AI easier and faster, we need new kinds of hardware and software, which is why Intel acquired Nervana Systems and its platform for deep learning.

Intel Nervana is hiring engineers to help develop a full stack for AI from chip design to software frameworks. Go to softwareengineeringdaily.com/intel to apply for an opening on the team. To

learn more about the company, check out the interviews that I've conducted with its engineers. Those are also available at softwareengineeringdaily.com/intel. Come build the future with Intel Nervana. Go to softwareengineeringdaily.com/intel to apply now.

[INTERVIEW]

[0:02:29.5] JM: Ammon Bartram is the cofounder of Triplebyte and Kyle Polich is the host of the Data Skeptic podcast, and today we're going to be talking about a variety of subjects. Guys, welcome to Software Engineering Daily.

[0:02:42.9] KP: Thanks for having me.

[0:02:44.1] AB: Hello.

[0:02:46.0] JM: We'll do some introductory stuff first. Ammon, you and I last spoke about a year and a half ago on the podcast. We did a show about hiring engineers which is what Triplebyte is all about. What have you learned about hiring engineers in the last 1.5 years? Catch me up on the diff between our conversation back then and what you've learned since.

[0:03:09.0] AB: Sure. I think the biggest thing that we've done is just continue to scale up. And so we announced a few months ago that we're now working with Apple and Facebook and so this is really interesting because we get sort of the insight on to the top of the hiring market as well, the biggest companies. I think the most surprising thing I've seen with those companies is that they actually generally run a better process, a pattern that we saw as we work with early stage startups is that often very small startups, the entering founders are heavily involved in hiring day-to-day and interviewing the candidate and selling candidates and making decisions.

As companies move up in size, often recruiters are hired, and this is what you get some of the more problematic, just cured, spinning on resumes and things like that. We're surprised to see that as we move into the bigger companies, they actually get much more engineering-driven. Once again, we see hiring managers making phone calls to help close candidates and really very sort of drastic in process.

[0:04:11.7] JM: The perception is that those companies have a really well-oiled machine. From the outside looking in, I'm just like, "Okay, they've got a very clear workflow for how the do hiring."

Triplebyte, for people who don't know, is a company where I'm sure you have a better way of phrasing this, but if you're a company and you want to get more engineers through your pipeline in your inbound hiring pipeline, you can outsource some of that hiring process to Triplebyte. It's hard for me to imagine how Apple or Facebook would want to outsource Triplebyte unless they really trusted you guys.

[0:04:11.7] AB: I guess they do. That's the positive answer. They're very interested in expanding the pool of who they speak to. We do all our screening background blind people. The goal is to be able to directly assess programing skill rather than rely on credentials and where someone's worked in the past.

Apple and Facebook are both have insatiable appetite for skilled engineers and a pretty high percentage of people on the job market already apply those companies. What they're most interested in is getting access to strong engineers who their own process screens out early.

An interesting thing is that — Facebook, for example, is actually explicitly interested in talking to people through us who have already failed their process. They've seen that we can — Because we directly measuring skills, we can find people who'll get screened out during the early stages of the process who are still actually strong engineers.

[0:05:50.6] JM: They can also use it to refine their own hiring process, because if they say, "Okay, there were some negative flag that we saw with engineer X," so they didn't make it through the Facebook internal hiring process, but they managed to make it through the Triplebyte hiring process. What's the diff there?

[0:06:09.2] AB: Yup. Interestingly, I think they're actually pretty — At smaller companies, I would agree with you. Facebook and Apple, they're pretty well keyed in on things that I think are predictive. I think the key explanation here is that, for example, if you have a rule of thumb that you're not going to go forward and interview people who have not worked as a professional

programmer and are currently doing IT support. That would be a statistically rule of thumb. You with would imbalance — Talk to better people if you skinned out people who have not worked as programmers and were working on IT support.

Of course, some percent of those people are actually very strong programmers. What we do is we don't' really invalidate their rules, we help them find acceptance of those roles and help them hire those people.

[0:06:51.8] JM: Kyle, you worked for 13 years before starting your own company. What was your impression of the hiring process over those 13 years; the hiring process for engineers?

[0:07:04.6] KP: Yeah. I was on both sides of the table there. Obviously, as I got older my career, more so on the hiring side, and I spent a lot of time thinking about it because there were times when it was difficult to find people and it was a slow process. I was always disappointed in most of the hiring processes I went through, so I wanted to what I couldn't change that around.

I came out of the whole thing kind of a game theorist look at it. In data science, one of the key concepts most people know that some engineers don't is that of a confusion matrix. You have something of ground truth; will a person do well in a role or not? The only way to know is to hire them, wait six months and find out. Of course, that's an expensive way to test someone, so you need to make what is essentially a prediction. In a lot of my cases, that's done by an algorithm, but a prediction is what a hiring manager and maybe team who's contribute to that decision is ultimately making.

If you on those up in the grid, when the team says hire, and it's a good hire, that's a plus. When they say don't hire, and it would have been a bat candidate, that's also the correct thing, but it's type 1 errors when you hire someone you shouldn't have that a really really bad, because they can be expensive and bad for morale and all types of things that everyone can kind of imagine. Then there's the type two error where there's a good candidate that you fail to hire.

More often than not, that's actually where people bias towards because the cost of not hiring good person means you just have to keep looking, so it takes you longer to hire. Maybe a little bit more money to hire, but is not as damaging as hiring someone who isn't a good fit. As a

result, we live in this world where think my advice to people on the market are expect to start with a shotgun approach. You've got to go broadly and see where you kind of fit and expect to be disappointed in surprising ways. Jobs you might be perfect might not get offered to you just because the interview process is so biased towards not letting people through that gate.

Ultimately, I think I've learned too that the system overall despite being often run by engineers is not run like a well-oiled machine. Most people are not trained in how to interview properly. They go into a room and seem to ask questions along the lines of, "Does this person know the same things I know?" Whereas they should actually be asking, "Can this person do the job?"

If there's any bit of wisdom I've got, it's to think more that way, that their ability to recall exactly what you studied or to quickly see the problem you've been working on for six months isn't necessarily the best test. There's all types of indicators people have to look at.

[0:09:32.1] JM: This notion of you should bias towards the false negatives rather than the false positives because you want to avoid making a bad hire. You would rather turn down somebody that probably should've worked at the company then hire somebody that should not have worked at the company. I always found this to be somewhat suspicious because what you should actually be evaluating is if the potential upside of a risky hire is massive, then it may be worth hiring somebody who could have some potential downside. Do you think hat that calculus around the biasing towards no. Do you think companies over-index on the no to their detriment?

[0:10:21.5] KP: Absolutely, but I don't know that that's ever going to change, because in my opinion the financials work out best that way. It can be very expensive to hire someone who's not a good fit but it's pretty cheap to fail to hire someone who might've been a great innovator at a company. It also comes down to a question of innovation. If you're hiring a technician like, "We need you to come in and run our exchange server. You've got 10 years' experience running exchange servers. It's pretty cut and dry. We can give you some sort of exam and you show that you're qualified to do it, and you come and basically run a machine that is not necessarily maintain or improve it."

In a lot of cases, I think one of the lucky things about engineers, data scientists, anybody in technology in general is by and large we have the opportunity to bring innovation, and that's

hard to measure. I think you're right in that sense, that sometimes bringing in a few mavericks or new ideas is what's really necessary to help a teen grow, but not every company has the kind of vision to see something like that.

[0:11:19.6] AB: Hey, Jeff. Can I jump in here?

[0:11:21.6] JM: Please.

[0:11:22.1] AB: I think an interesting sort of — Everything Kyle said is I think totally correct. I think an interesting addition to that is that companies are not even aware of what their false-negative rate. Companies play the trick to themselves where they just trust confidently that everyone who failed their interview was bad. It does make sense to optimize for fewer false-positives and more false-negatives, but I think companies still underestimate their actual false-negative rate, and you can look at that by looking at the agreement rate between companies.

Given that candidate did really well and got a five out of five at company A. What's the probability that they get a five out of five at company B? We found that to be startlingly low.

[0:12:10.0] KP: Yeah. I like, especially, Ammon, with companies like Triplebyte that you're at, are doing and that if a the company is screened for 500 people or haven't positioned in 500 people got knows, the idea that all 500 of them weren't qualified is kind of ludicrous unless you're really doing something bad at the front-end recruiting. Having a tool to say, "Let's go revisit our false-negatives and let them do the screener," and a few gems kind of shine through who are really nice auxiliary to that process that can combat this kind of bias towards only type two errors.

[0:12:42.9] JM: I know that I'm thankful that I don't have to go through the meat grinder of getting hired as an engineer at least today anymore. I'm sure that you two are also thankful for that fact. You guys, at this point, are both on the hiring side of the table, if anything. Let's talk some about getting to that place where you have started your own company and you've gotten off the ground. There's a lot of people who listen to the show who are curious about starting their own software company or starting their own consulting business. What have been the

hardest parts for each of you for getting your own organization off the ground? Let's start with you, Kyle.

[0:13:30.5] KP: I guess it's knowing what's a smart thing to work on and what isn't. Pretty much anybody with some engineering talent has no shortage of ideas of cool stuff that could be built, but cool stuff doesn't always pay the bills. Deciding what is going to pay the bills versus what's going to have some returned on the road is important. Kind of assessing where myself or anyone that I'm helping can really get in and make a significant impact I think is a factor beyond the financials that has to be carefully considered.

[0:14:01.5] JM: Ammon, what about you?

[0:14:01.9] AB: Let's see. I'll probably just the last 5% problem. The last 5% of any task ends up taking 50% of the time. Multiply that by times four, five, six concurrent things that a company is focusing on. There's this explosion of stuff that needs attention. I guess this is actually kind of getting back to the same thing that Kyle spoke about. In that sea of things you'll be focusing on, figuring out what the things are actually ball moving is an ongoing struggle and I think one of the most important things that I've gotten better at overtime.

[0:14:36.3] JM: Yeah, it's almost round-robin — Well, not round-robin. It's almost like a priority queue scheduling problem from my point of view at least where you've got different queues of tasks. You've got your inbox. You've got your calendar. You've probably got a to do list somewhere and it's almost like you've got a single CPU and you just switch your personal CPU between your different schedulers and try to schedule tasks against that CPU as efficiently as possible.

[0:15:08.6] AB: Performance goes down.

[0:15:11.2] JM: Performance go down, although your context switching probably gets better with time.

[0:15:16.3] AB: If only that were the case. I'm not sure that's true.

[0:15:22.3] JM: Okay. Is multitasking possible?

[0:15:24.2] AB: It's definitely possible, but there's a pretty big performance hit. I think I'm still biased too far toward doing things that I feel like work but are not important. For me, random example, answering email. I think I should probably never answer email and yet I continue to do so.

[0:15:39.6] KP: Yeah, I have a similar thought. I know that I can get better at it. Another consideration I have is I've gotten older in my career and saying, "Where do I want to be later?" If I get good at everything, that I'm only a little bit good at everything. If I want to be a deep expert at one thing, it often means cutting out all of the — Either cutting or delineating or delegating all the stuff that is in court of what I want to be the best at. Knowing where to strike that balance is sort of multitasking, but I think that's the real key.

[0:16:07.8] AB: I think an interesting part of it — I don't know. I set out 10 years ago at the beginning of my career with the goal of wanting to be deep technical expert, sort of skilled engineer. As I move into starting companies, that's not what it takes to start a company. With it takes to start a company is broad confidence and a bunch of things.

[SPONSOR MESSAGE]

[0:16:34.4] JM: Dice.com will help you accelerate your tech career. Whether you're actively looking for a job or need insights to grow in your current role, Dice has the resources that you need. Dice's mobile app is the fastest and easiest way to get ahead. Search thousands of jobs from top companies. Discover your market value based on your unique skill set. Uncover new opportunities with Dice's new career-pathing tool, which can give you insights about the best types of roles to transition to.

Dice will even suggest the new skills that you'll need to make the move. Manage your tech career and download the Dice Careers App on Android or iOS today. To check out the Dice website and support Software Engineering Daily, go to dice.com/sedaily. You can find information about the Dice Careers App on dice.com/sedaily and you'll support Software Engineering Daily.

Thanks to Dice for being a loyal sponsor of Software Engineering Daily. If you want to find out more about Dice Careers, go to dice.com/sedaily.

[INTERVIEW CONTINUED]

[0:17:54.9] JM: Right, and then you may be build a company that overall has a specific acumen in something that's extremely differentiated, like Triplebyte, which is the acumen is hiring engineers. You're trying to build the company that is best at understanding the engineering hiring process. Actually, I want talk about that a little bit. From the outside looking in, the strategy for Triplebyte was a series of things.

You started by just interviewing engineers, and so this was a service that other companies could outsource to you. Your second step is to pinpoint problems of the hiring process because you have so much engineering flow coming through the door and you're interviewing all these engineers. Since the entire company is focused on it, rather than the typical place like Google where they're focused on engineering and building new things in business strategy, and then also they have to do hiring.

By focusing on it completely, you're probably going to have insights that less focused companies would not have. I think the third goal which maybe you're starting to do or you have already done is you want to build products and services that solve some of those problems around the engineering hiring process. Am I describing the Triplebyte roadmap correctly?

[0:19:13.2] AB: Yes.

[0:19:15.5] JM: Okay. Tell me more about where you are in that roadmap. Do you have ideas for what — As I understand, I just imagine Triplebyte and I imagine every day is there is a huge crowd of engineers that are coming through the door, being interviewed and you're just taking notes and adding those notes to some sort of internal tool. Have you gotten to the point where you're starting to evaluate some products and services that you can build for companies that need help with their hiring yet?

[0:19:43.7] AB: Yeah. I think the only correction I'll make to the process of that, actually engineers apply to us. We've never done interviewing as a service. It's always been engineering applying to us and then we help them get jobs at companies. I think the sort of — Honestly, what you're stating out is a better phrase strategy than I think we had at the beginning. Basically, we just worked as engineers and seen how hard it was to hire talent and we'd also been engineers and done interviews and realized how frustrating it is to be sort of asked what feel like trivial questions. We set out is to focus on it full-time and see if we could build a better process.

I think the core thing we now focus on is a problem I didn't even realized existed when we started, and it's that companies look for pretty measurably different things in interviews. This is partially rational and partially culture. For example, some companies — I don't know. They deal in real money and it's super important that they have very high security standards. Having a great engineering process and doing careful code review and thinking a lot about security is core to their culture.

They, for example, probably want to fail someone who has a really fast, iterative, get stuff done, but don't worry too much about the details to programming. Other companies, they're building a social probably. They want to move as fast as possible and they run experiments and will fail people for being too slow and liberative and not being superfast, move fast and break things.

That makes sense. That's rational, but differences extend to deeper things too, or just more arbitrary things. Company A just has a strong belief that if you don't programmed in compiled language, that's the sign that you're not serious. Company B thinks if you programmed in a compiled language, you're stodgy and old.

I think about 50% of interview failures are due to avoidable fit, avoidable mismatches. A company just — Someone not be having the basic approached engineering the company is looking for. What we found is that if we can measure this stuff, we can get the interview pass rate basically doubled. Over the last two months, about 53% of our interviews have resulted in a job offer at a company, and that's roughly twice the rate at those own companies and we're able to get that rate just by basically measuring these things. Honestly, this is not the model I had in my head when we started two years.

[0:22:03.2] JM: Something I read is that one goal of Triplebyte is to build the engineered genome. You talk about these different attributes like attention to security, or creativity, or proclivity towards social products. You could imagine these being different aspects of an engineered genome and then you could also make the ideal genome for a Facebook engineer and then you can do some sort of similarity calculation between all the different company vectors and the vectors of each engineer and you can probably get really good at the matchmaking process.

[0:22:53.5] AB: That sounds like you've described sort of like a nearest neighbors approach to the matching. That's one model that we have tried. We have also tried [inaudible 0:22:58.3] interestingly. We've found who worked the best actually is to just try to define a few relatively small number, like at this point we're about six axis and then just measure how much each company cares about those axes.

[0:23:16.0] JM: I heard you in another podcast talk about side projects and the fact that side projects are not a strong positive correlate with getting hired. My impression is that side projects are actually strong correlates of who makes for a successful entrepreneur. If you're really good at starting your own side projects, you're probably good at starting your own company. I found it interesting to hear you say, if I remember correctly, that the hiring process does not select for these entrepreneurial people who start side projects. Why is that?

[0:23:54.9] AB: I would state things a little bit differently. Is that what we found is that asking people about side projects was not very correlated with how well they could do engineering interviews. Talking to someone for 30 minutes about their side projects and giving them a score of how competent they seemed was poorly correlated with then giving the programming problem, either a take home problem or in-person interview and seeing how well they would do on that problem. The basic reason was that communication skill ended up dominating talking about side projects.

Ability to sound impressive and organize your thoughts well and generally sell yourself ended up dominating. Some of the best people, for example, best engineers, are very quiet and geeky and not necessary great at self-promotion. Those people often do a bad job when talking about

side project in previous work, but then come in and totally crush if you give them a real assignment.

[0:24:47.9] JM: All right. I want to come back to hiring. I want to talk about skepticism a little bit. Kyle, the title of your podcast's Data Skeptic. It's the most popular data science podcast or one of the most popular ones and I've been enjoying it recently. We live in a time of ever increasing skepticism and you can just see this on Twitter and in the news and it makes sense for people to be increasingly skeptical because we have more data and so we can confirm or refute the hypotheses that people have.

There are a lot of things that it makes sense to be skeptical about. For example, I think about the wide spread use of antibiotics in livestock. Is that really a good idea? Maybe we should revisit it. Do we really have the data to support the current level of use of antibiotics in livestock? I don't know. There are other things we should probably take at face value. For example, the fact that we should eat vegetables. It's pretty obvious from the date of that there's not much downside to eating a lot of vegetables, but I cannot say with extremely strong evidence that eating spinach is going to have this output affect. How do you decide in today's world what to be skeptical about?

[0:26:17.6] KP: At a high-level, first and foremost, everything in moderation. Having the sort of shields up at all points for information coming in I think is a healthy way to be, but not when it gets into tinfoil hat territory of course. The rough heuristic I have for myself is I want to be skeptical of things commensurate with how that information will affect me. If you'll humor me with a tangent, I want to give the definition of something that I think is really important and frames a lot of the ways I look at things, and it's how do we measure the value of information?

For example, let's just say you were thinking about buying a used car and let's also say it's very simple that's it's either a lemon or it's not and you don't want to buy a lemon. You can go look at the car, maybe you don't know much about cars, so you're 50-50, is it a lemon or not, or you can go hire a mechanic who will charge you money, there'll be a cost, and then you'll know with a lot more accuracy whether it's a lemon or not. Maybe not perfect, but let's say 95% accuracy. How much should you be willing to pay for that?

There's a cost in the world in which you buy the used car with no information and you can still make a decision, maybe you get lucky it's a good car, maybe it's not, but there are some expected outcome there. You can also measure the expected outcome of how you would act with the information. It's, “What's my outcome acting with the information, minus what I would've done acting without the information, minus the cost of the information?” and that's the value of the information.

Am I concerned that there are some — I recently heard that people believe that Finland doesn't exist, and that's a conspiracy. Is it really worth my time to investigate and look at unless I want a good laugh, I guess.

[0:27:57.0] JM: Do people really believe that?

[0:27:59.1] KP: You know, it's hard to say for sure, but it seems to be the case that at least some very fringe group may be serious about that. Even if that one is a joke, there are people who genuinely think we did land on the moon despite overwhelming mountains of evidence for that. Like your antibodies case is an interesting one, because that is one I don't have enough to say about, and I can't go either way. It’s easy for me to reject the Finland example because it’s absurd. It’s also easy for me to accept something that comes out in a certain press release because I trust it's really vetted with the scientific process.

When people say — People who protest science, they’re saying the horrible things are happening because of these antibodies. They might not have the right way of conveying that message, but there might be something to that. I eat meat. I don't eat much and I tend not to eat beef, so I'm not too worried about the medical aspects for myself personally, but I do have maybe an ethical concern. Is that a reasonable way to treat livestock? For some people, having livestock at all is ethically no for them. I don't feel that strongly about it.

Then there’s maybe a broad medical concern. Can that practice somehow leak into the drinking water or something like that? Doesn't seem like it because we have an FDA and all these checks in place, but you never know. I also know I'm not qualified to ask a question like that, so I look to expertise. There is one podcast I listen to every week that has a doctor on it who I feel is sort of my source for going out and monitoring those sorts of concerns, and I'll take what they

say with a tiny grain of salt, of course, because they're not an absolute source of truth, but you want to have trustworthy expert places to get information from. I guess be skeptical of everything, but in moderation.

[0:29:39.9] JM: How do you express skepticism when you want to express skepticism around things that are taboo to be skeptical about? Let's say you decided to go deep on vaccines. Let's say you're going to have a kid and you're like, "Well, I hear, let's say 3% of the people who I respect are at least a little skeptical of vaccines or they're scared of vaccines." You decide 3% is enough of your trusted network that you want to go deep yourself and you want to become an authority on vaccines. Let's say you go deep on in and you're like, "Wow! There actually is some evidence here that is a little spooky."

Then you decide you want express some skepticism to those around you. How do you express skepticism around things that are taboo?

[0:30:31.8] KP: There's a couple of kinds of taboos, but I think the case of vaccines are really good example because what makes them taboo I think in part is because parents feel they want to protect their children, and that's a really good sentiment to have. But we also have to ask what we're protecting them from.

Vaccines have almost eliminated polio in the world; measles, mumps, rubella. As a whole, we're a much healthier species because we've been vaccinating. The claim that's out there is that vaccines can cause autism. Because there is the potential for great harm here not only — If that were true, that would be really bad. When people stop taking vaccines, we lose herd immunity. In fact, we've seen outbreaks in certain communities where their infant mortality has skyrocketed because of this information being out there.

Is it worth having a few less cases of autism for triple the death rate in the general population? I don't know. But let's return to the autism claim. We can trace all that back and largely on that topic all of the concerns stem from an initial publication from a man called Andrew Wakefield. That publication has since been discredited, not replicated. It was taken out of the peer reviewed public literature. The scientific community is pretty clear that vaccines are safe for the most part.

There are some cases where certain people can't get vaccines because of pre-existing or other health conditions. I'm not really qualified to talk about that exactly, but I think that brings me to the most important point is you don't want to take your advice on vaccinations from Jenny McCarthy. You want to go to your pediatrician and talk to them and make sure to it's a real pediatrician, not a nutritionist, not a chiropractor, not a foot doctor. Talk to someone who really knows this literature and works on babies all the time, and should have that person as a parent. That's the important person to get your information from, and I think that's true in just about any aspect.

The other end of taboos is people who say, "I believe in some crazy idea," the moon landing hoax people. I used to kind of just say, "I'm going to not engage with them, or ignore it, or whatever. What harm is done? It's just kind of a cookie person running around with a cookie idea."

Actually, I've taken the position that I'm kind of insulted by that on behalf of all the brilliant scientists and engineers who put a freaking man on the moon. That's an incredible accomplishment, and to say that it was all fake, it's like, "Really? What were those people doing working so hard on so many problems?"

Another one is young earth creationists, people who express that their belief is that the world is 6,000 years old. While I may respect your right to whatever religious point of view you want to have, that's just demonstrably false. As I've gotten, I've gotten less willing to kind of go along with that. At least professionally, there is a taboo, a lot of work places you come in with preconceived notions. I guess it hasn't been as controversial as I wanted to be, but I'm fond of saying is that those people who claim something as an art and a science just don't have the science figured out yet.

I've heard that in a lot of workplace environments where people are maybe protecting an old process or trying to maintain power in an organization and they're, yeah, you have to be very diplomatic about figuring out, "Well, we've innovated in some way. How is that going to affect the organization? How do we make sure everyone is valuable, making a contribution here?" Those

taboos are the ones I actually struggle with the most and seeing how we can get everyone on the same page to move an organization forward.

[SPONSOR MESSAGE]

[0:34:09.9] JM: Ready to build your own stunning website? Go to wix.com and start for free! With Wix you can choose from hundreds of beautiful, designer-made templates. Simply drag and drop to customize anything and everything. Add your text, images, videos and more. Wix makes it easy to get your stunning website looking exactly the way you want. Plus, your site is mobile optimized, so you'll look amazing on any device. Whatever you need a website for, Wix has you covered.

So, showcase your talents. Start that dev blog, detailing your latest projects. Grow your network with Wix apps made to work seamlessly with your site. Or, simply explore and share new ideas. You decide. Over one-hundred-million people choose Wix to create their website – what are you waiting for? Make yours happen today. It's easy and free. And when you're ready to upgrade, use the promo code SEDaily for a special SE Daily listener discount. Terms and conditions apply.

For more details, go to www.wWix.com/wix-lp/SEdaily. Create your stunning website today with Wix.com, that's W-I-X-DOT-com.

[INTERVIEW CONTINUED]

[0:35:38.1] JM: I tend to agree with what you said about the art versus science thing, that people who will say that there's an art typically are people who have not acquainted themselves with the science so much, or that's often the case at least. One thing that I have trouble with sometimes is when I am talking to somebody about one of these really hot topics, like vaccines or climate change, is there a phrase that people pull out of their hat which is, well, absence of evidence is not — Or what is it? Absence of evidence is not evidence of absence. They'll say these sort of thing where it's basically implying that this system is so complex that you actually cannot make a strong judgment on this system with your dataset, with the size of your dataset.

On some these arguments, that's actually kind of a compelling argument. Is the response thereto just — I don't know. Like defer to the conversation until you have more data? What do you do?

[0:36:42.8] KP: I think you have to refocus at some point. A topic that's come up a lot in my professional life is deep learning has grown and how much it's impacting industry and work in different places and people point at deep learning and say, “Oh, it’s a black box. No one knows how it works.”

I actually disagree very strongly with that statement, but that's another show entirely about how do we interpret deep learning. Let's just say let's humor that idea and say that technology is 100% black box and humans will never understand what's going on inside of it. That blackbox still has inputs and outputs. If a vendor shows up and they say, “We have a tool to help you in your hiring process. You give us all these inputs and we’re going to run them through our deep learning algorithm and out will come a recommendation hire or don't hire.”

You can put in test applicants. You can put in a deliberately bad resume to make sure it flags it. You can put in resumes that you, in advance, have some idea about these were good ones are bad ones and see how it scores them. You can run empirical tests and say, “Whatever I'm interacting with has some claim that something is coming out of it.” Whatever the claim is should be testable in some way even if you can't totally understand the mechanism.

[0:37:53.3] JM: Ammon, we’re talking about skepticism and we’re talking about hiring earlier, what are the common beliefs around hiring engineers that when you started Triplebyte people would just say these things, like sacred cows, and you would say, “Well, I do wonder if that's actually true. How can we test that?”

What are the common beliefs around hiring engineers you’ve become really skeptical of?

[0:38:20.9] AB: Top my list there would just be very specific ideas that companies and engineers have about the right way to identify skill. This gets back to the idea that sort of most people doing interviews are not — They’re not carefully trained in what they’re asking. They just sort of have some questions that maybe they were asked earlier in their career that they then

sort of copied. That often how it works. Then they ask those same questions going forward to everyone they interview.

What happens there is people think there's this one thing that is just the magic solution. Ask the candidate about concurrency, and if they talk about concurrency, then they're a struggling engineer. Or ask the candidate to code on a whiteboard, and if they can do that, they're good. Or never ask them to code on whiteboard if they — Whiteboards are terrible.

I think what we've found just looking at inputs and outputs. Looking at people in the door and trying to predict who's going to do well at a company. What we found is just there are many many different ways to be good. Someone can be a great programmer and if you ask them at working on a whiteboard, their actual arguments for why a whiteboard is a good way to run a programming interview. This is something that a lot of people have very strong belief that whiteboards are terrible and there is a slight argument.

The argument is giving an engineer a whiteboard in an interview, you are removing the constraint that their code actually compile and run and so they don't get these lose time worrying about the exact syntax of an API call. They can just focus on the core logic.

I think ultimately that falls down — If that's what you care about, you should probably just have the programmer write code on a computer and then tell them that doesn't need to run. But there is that argument, and so the more I've done this, the more I've just sort of come to challenge all of the beliefs that everyone has. The answer is that it's some weighted combination. The true signal is just some weighted combination of all the different parts that everyone has in their head and anyone part can they do a stray.

[0:40:20.5] JM: What are the subjects around hiring that are taboo to discuss?

[0:40:25.9] AB: Okay. I'll give the easy first. The easy answer is actually all these stuff. If you look at any discussion on Hacker News about a blog post about hiring, you'll see deep passion over these exact issues, over whether whiteboards are fair, over whether algorithms, over whether any specific interview question is fair.

You see person A, engineer A is taboo, colliding with engineer's B strong opinion and the result is kind of a flame war. I think that's one answer. The second answer of course is hiring and the hiring process is sort of on the frontier of diversity in tech and age discrimination. Those are two very relevant topics that because of their importance, have a certain amount of taboo.

[0:41:14.2] JM: Are you making efforts to normalize against those or are you just starting to study them?

[0:41:20.0] AB: Yes. The main thing that we do is try to — Our goal is to directly assess skill, and so we hope that by ignoring backgrounds, we are leveling the playing field. We've modeled our data to make sure that we don't see any problematic signs of bias. Beyond that, our primary focus is on how do we directly assess the skill.

[0:41:45.9] JM: Triplebyte is a Y Combinator company and I've seen some great posts from the Y Combinator founders**:** Jessica Livingston and Paul Graham where they talk about taboos. Jessica Livingston had a post several months ago called the Sound of Silence, where she was talking about the most useful things that she's learned at startups, but she was talking at the fact that she actually can't talk about them because if she were to discuss them publicly, they would be too controversial.

Paul Graham had a piece about this called What You Can't Say, that was really popular. I think that's like five or 10 years old. If you're a startup founder and you need to know these things, these useful things that are taboo or controversial, how do you get access to that information? How do you get people to tell you these secrets?

[0:42:40.9] AB: Well, let me push back a little bit on the idea at least in the scope of hiring. The topics that are most taboo in hiring are things that I think the proper response is to ignore them. The best response is — There you go. I'm wading in deep here. The goal of hiring process should be to be — When directly assessing the skill, to be blind the background. The goal is that female programmers and programmers of different religions and backgrounds are treated identically in the technical evaluation. I think it's definitely we're sort of reading up on unconscious bias and the things that can be done to improve that

Beyond that, I don't think there are any sort of huge dark secret that I could say that would help a company hire better. It's a very charged topic. It's very very important and everyone's really interested in it. As the founder of a company working in this space, I need to be careful about what I say because there are a lot of controversy I can weigh it into. I don't think that's ultimately resulting an important information being suppressed and not communicated to companies.

[0:43:50.3] JM: Yeah. Maybe I should be more clear, because what I'm referring to is not necessarily the taboos of hiring, but specifically — I guess I should say more generally, Jessica is talking about the fact that there are these things that people can't talk about publicly because they're too controversial. Maybe it's something like if you're a founder — I don't actually believe this, but maybe in some secret, like if you're founder, you should have a bunch of ways where you can screw your later employees out of their equity. You should have these little clauses in their contracts where you can actually screw them and pull the rug out from under them. That would be an example if it were true of something that you would not be able to talk to publicly because they were controversial, but maybe it's positive expected value to actually do that as a founder.

My question is more about if that kind of thing were true, something that people actually could not write about without getting in trouble with the community, how would you find out about it? Is this just something you have to be kind of in the in group and you have to have the right — You have to be sitting down at the right dinners where people will talk about this stuff?

[0:45:01.0] AB: Yeah. Okay, that makes perfect sense. I think friends, mentors — I've been to Y Combinator myself, so that's sort of great place to get that kind of information. Let me actually address your specific example though because I find it interesting.

An example of that is the vesting schedule for employee equity. I don't know if you're familiar with this issue, so that incentive stock options by law have to be exercised within 90 days when a candidate leaves the company. That's the standard that most companies go with and this ends up putting employees under a lot of pressure when they leave. Often, they're able to pony up a large amount of money to exercise options or forgo the equity.

Yeah, I've been in rooms where it's debated. That policy of 90-day comes from tax law, but it's possible to change how you issue the options plus that they convert into FMISO's to NSOs at 90 days, and there are a lot of debate internally [inaudible 0:45:52.9] about whether that's good. It's friendly to the employees because it gives them equity they've already vested, allows them to hold on to it. But on some level, it's makes it easier for people to leave your company.

There are founders who argue that you're being too bleeding heart if you go with the log or exercise. I think that startups are actually pretty biased against employees, and so I'd actually would rather startup founders maybe don't have access to that kind of information. Maybe it's actually good for the world if ideas like that don't spread and people just do the thing that's friendly to the employee.

[0:46:27.6] JM: What I thought was interesting about this debate, I remember reading about this whenever it was in the news a year or two ago and I think there was a point counterpoint between Scott Cooper of Andreessen Horowitz, which is the venture capital firm that brands itself as being the most founder-friendly, and Adam D'angelo of Quora who wrote about the 10 year vesting schedule that they have a Quora and writing about why that's important and why it is more in accordance with long-term thinking. Do you remember the framing of each side of that argument?

[0:47:05.4] AB: Yeah. Yeah, I read that. I believe so.

[0:47:10.1] JM: Yeah, if you remember — It's kind of random off topic, but I'm actually curious about that. We haven't talked about that on Software Engineering Daily.

[0:47:17.1] AB: Okay. First, I'm going to just go over the issue so that listeners are familiar with it. Employees are given an equity grant usually in the form of options and those options vest schedule often one quarter, after one there, and then the remaining three quarters over the next three years. By law, there's after almost always ISOs, which means they're given a certain tax preference when exercised.

By law, ISOs have to be exercised within 90 days of when an employee leaves a company. That means if someone is an employee at a very successful startup, maybe their equity has

appreciated a huge amount and so they need to — In order to exercise those options, they need to pay a tax that difference. Difference between the market value and the amount they're paying, and so that can be tens of thousands of dollars, maybe hundreds of thousands if it's a very successfully company. That sometimes prevent — That's basically a bet. You're placing a bet on the future success of the company, and so that places a lot of pressure on employees to make this hard decision when they're leaving a company.

The argument against it, the argument in favor of the 10-year exercise window is basically that we're talking about compensation that the employee has already earned, the employee — You sign a contract, the employee was X% of the company and options over four years. They stayed for those four years. They've earned their equity and it's just like totally unfair to sort of via some tax law and the fact that they don't have access to capital. You're going to claw that back from them. I mean it's just that being fair to your employees and giving them what they've earned is a better foundation for a company. That's the position that I take.

The counter-argument is just that startups are brutal and you have to be efficient, and the current — Basically, taking caps of the business, taking ownership out of people who are actively in the company is harmful. The argument is that a company that have the higher probably of succeeding, if as high percentage of the equity ownership as possible is held by people who are currently on the ground at the company you're working trying to make it succeed. You want to minimize that involve people who are no longer involved in the company's day-to-day owning equity. By making it easy for people to quit and take their equity with them, you're sort of diluting the incentive to succeed.

[0:49:34.9] JM: Yeah. The pressure with a 90-day option exercise period is you have the spend cash on an illiquid asset in the hopes that that asset is going to be worth some money in the event that the company has a liquidity event. It's a such a gamble to pay for those options in the 90-day period. You should have the option to execute that purchase over a larger period of time. That is my opinion, especially because silicon valley's a small place, and if you hear about, "Oh, Triplebyte is the place where if you go and you leave your job, you're going to get screwed on your options. You're going to have to fork over a bunch of money on this big gamble if you want to actually take advantage of your vested options." That's actually really harmful for hiring process.

Not to name names, but you look at Uber, and like these things add up. If you have cultural issues, and I would classify this as a cultural issue because it essentially comes down to, “Are you going to have a cutthroat option schedule or not?” That’s going to have cultural implications, and the lesson of Uber is that these things add up. I think this is a good segue into self-driving which is the last topic I wanted to talk about. When you get to a place where the market is so competitive, the market for self-driving engineers has got to be one of the most competitive engineering markets that’s ever existed. It's basically like, “Who was going to go work—” There are people who will still go at Uber. I know they are still hiring some great top self-driving engineers, but the reputation stuff has really harmed then.

Anyway, the self-driving market. I self-driving such a domain specific expertise of engineering that Triplebyte is sort of staying out of that area, or do you feel like you could still service a company that is looking to hire self-driving car engineers?

[0:51:49.2] AB: We’re placed a bunch of engineers at Cruise particularly by GM last year. I think it varies a lot by — Perhaps Kyle could speak to this as well. It varies a lot by company what — We’re not working as much with people who are sort of deeply experienced self-driving specialists, but of course there’s a range of experience with an engineering team. Cruise, for example — They’re certainly hiring small number of deeply experienced self-driving and machine learning specialists. They’re also hiring people to sort of just work on building the actual production system on specialized operating system and people to work on the infrastructure around it and people to work on the sort of the machine learning testing infrastructure.

There’s sort of wide industry around of sort of the — Calling the support roles is actually too majority, right, because they’re actually intellectual —

[0:52:43.7] JM: Data engineering.

[0:52:45.9] AB: Any team is going to have a small number of deep domain experts and then a larger number of sort of people doing the actual rank and file work course engineering work. I think those jobs are actually pretty approachable.

[0:53:00.2] JM: Right. Okay, Kyle. Have you thought much about how data engineering and data science at a self-driving car company is different, because it seems like the volume of data that's coming in at these companies is an order of magnitude larger, or it threatens to be. Maybe it's not today, but it threatens to be. Have you thought much about what the data engineering pipeline for a self-driving company looks like?

[0:53:27.4] KP: To a certain a degree. I don't know as much about what's necessarily in all those cars because I'm not actively in that industry. I follow it. I'm hugely excited by it. I can't wait. I, in fact, have a prediction that my young niece will never be allowed to drive a car, because I think by the time she turns 16, it's going to be illegal for humans to drive, or at least as the world I hope I get to live in.

As far as — Yeah, there's massive amounts of data that's probably generating a gig a second, from the light R and the cameras and everything, but all that information is very transient. Amazon knows every purchase I've made over the last two decades. My self-driving car is going to not have to remember how bad of a left hand turn I made six years ago. You need very quick decisive information. The data storage isn't as big of a problem. I think it's going to be more around throughput and things like that.

On the other side, we know that the main technology that's really making these cars happen is deep learning. Obviously, I'm sure there's been innovations in the laser space and all types of stuff like that, but at least the most front and center thing is deep learning has made a lot of these image and video processing possible, and those models are incredibly difficult to train offline, but once you've trained them, they're relatively easy to run in production. Then it's just a throughput of how to we make sure the cameras are synced up to this thing and it's kicking out all the signals at once. Depending on how you've organized that, if what your first wave of infrastructure does it just pull out geometry from the scene, if it does it really well and there's not a lot of errors in it, then the problem at that point becomes a little bit simpler from there forward.

I don't know so much about the architecture, but kind of as a mindset, it's not just support roles. It's clear self-driving cars are going to place some big role in our economy, in our society. Yes, maybe the coolest obvious thing to work on is the video kind of work of it, but even how will

highways change when the majority of cars are automated, and urban planning and stuff like that. There's going to be a lot of supporting really interesting engineering problems surrounding it. Self-driving cars are these blossoming area of different cool problems to be working on.

[0:55:34.6] JM: Ammon, do you have an any perspective for how the self-driving car growth is going to affect hiring both in terms of the amount of roles in self-driving and the salaries?

[0:55:48.7] AB: Salaries we see I think are pretty in much in line with the rest of the industry. One comment I'll make which is interesting is there's actually a pretty big surplus of interest in machine learning roles in general and also in self-driving car roles right now. If you're an engineer, four or five years of experience and you want to get involved in machine learning and self-driving, it's actually pretty hard. There're huge amount of people with actual demonstrable industry experience building an actual system. Those folks are very highly in demand that they get bid up, salaries go up.

People who are trying to move from [inaudible 0:56:25.5] to machine learning have may be sort of done down the Coursera courses and done a few projects at their job and are applying for a full-time machine learning role. Those people we see actually struggle a bit. Often, taking sort of a slightly tangential path end. They take — We see those folks moving into data engineering roles and working doing those for a few years and then moving on to actual machine learning role.

[0:56:46.7] JM: All right, guys. Thank you so much for coming on Software Engineering Daily and talking about a variety of topics. Everyone listening should go check out Data Skeptic, and if you're looking for a job, you should check out Triplebyte, or if you're looking to hire engineers at a better, faster pace, check out Triplebyte. Thanks again, guys.

[0:57:06.5] KP: Yeah, great to be here.

[0:57:07.1] AB: Thank you.

[END OF INTERVIEW]

[0:57:14.4] JM: You have a full time engineering job. You work on back-end systems of front-end web development, but the device that you interact with the most is your smartphone and you want to know how to program it. You could wade through online resources and create your own curriculum from the tutorials and the code snippets that you find online, but there is a more efficient option than teaching yourself.

If you want to learn mobile development from great instructors for free, check out CodePath. CodePath is an 8-week iOS and android development class for professional engineers who are looking to build a new skill. CodePath has free evening classes for dedicated experienced engineers and designers. I could personally vouch for the effectiveness of the CodePath program because I just hired someone full-time from CodePath to work on my company Adforprize. He was a talented engineer before he joined CodePath, but the free classes that CodePath offered him allowed him to develop a new skill, which was mobile development.

With that in mind, if you're looking for talented mobile developers for your company, CodePath is also something you should check out. Whether you're an engineer who's looking to retrain as a mobile developer or if you're looking to hire mobile engineers, go to codepath.com to learn more. You could also listen to my interview with Nathan Esquenazi of CodePath to learn more, and thanks to the team at CodePath for sponsoring Software Engineering Daily and for providing a platform that is useful to the software community.

[END]