“Everybody in this country should learn to program a computer…because it teaches you how to think.” –Steve Jobs

“Whether you want to uncover the secrets of the universe, or you want to pursue a career in the 21st century, basic computer programming is an essential skill to learn.” – Stephen Hawking

“People think that computer science is the art of geniuses but the actual reality is the opposite, just many people doing things that build on each other, like a wall of mini stones.”  
– [Donald Knuth](http://en.wikipedia.org/wiki/Donald_Knuth) (computer scientist)

There’s been a growing sense around the world that everyone should learn at least some code. And experts from business luminaries like Steve Jobs to top scientists like Stephen Hawking and even computer scientsists talk about programming as if it’s something that anyone can learn.

So that’s the claim – anyone can (and, everyone should) learn to code.

But, if that’s the case, why is it that programming continues to be such a…well….non-diverse field?

Stats Alert!

In Australia, women receive over 50% of the undergraduate degrees (56%) but earn only 15% of the degrees in computer science. Workforce numbers are slightly better with women holding 20% of programming jobs. Numbers in the United States are very similar. (58%, 13%, 20%) - though that number is closer to 12% in the heart of Silicon Valley.

The stats get a bit more depressing when looking at ethnic diversity.

30% Asian, 5% Black, 3% Hispanic (male + female)  
  
When I specifically looked for data on indigenous peoples, I discovered that their employment in tech is so low that a percentage could not be given.

Do you get that? There are entire groups of people for whom careers in tech aren’t even an option.

A lot of people have focused on how to get more diverse candidates into the pipeline. From making sure that women and minorities are fairly evaluated in the hiring process, to recruiting from sources other than comp sci graduates to helping encourage more girls to consider computer science degrees in the first place. Google has actually been focusing on this – even having initiatives to work with Hollywood to make sure that there are positive portrayals of female/minority computer programs in t.v. shows.

And all of that is great….

But it fails to address one major issue:

Women and under-represented minorities leave tech at a faster rate than their peers.

I’ve often heard diversity in tech be compared to a leaky bucket…..you can keep dumping new people into the top, but, if they drop right out of the bottom, you haven’t accomplished much…well, that’s not true….you’ve possibly dashed some dreams and made it even less likely certain people will pursue careers in tech.

Kind of depressing…..

But there’s still hope. There are direct actions that you can take to make sure that your dev team is welcoming to a wide variety of people. I’ll cover 3 of them….because 3 is a good number and I don’t want to keep you here all day.

LANGUAGE

…because everyone’s favorite police are the language police

Last May, a prominent Rubyist tweeted that the best part of being in management was being able to set a culture of non-tolerance for racist or sexist jokes. In her previous jobs, especially as a junior team member, there had usually come a point where someone told a racist or sexist joke (usually one they deemed innocuous) and the team would look to see her reaction. Laugh along and you’re a cool team player. Complain and your time there is short.

The tweet storm got a decent amount of interaction (60 shares / 100 likes per tweet). What I found more interesting were the private conversations it spawned. Friends texted and asked if I had seen it. They shared their stories of someone at their work that made them uncomfortable. Of team meetings where they had to laugh along with jokes they felt were inappropriate or demeaning. The very few who had spoken up at best were told “Yeah….that’s how that guy is. Just ignore him.” And at worst were told they were being overly sensitive. Most just didn’t say anything at all. And that’s the thing, whether justified or not non-traditional developers largely felt that speaking up wouldn’t do them any good. Instead, they were all quietly looking for another job. Likely leaving their employers to wonder why all of the diverse candidates they’d worked so hard to recruit kept leaving in droves.

At my company, we have a rule – let’s call it the Flatiron Rule. It goes like this – If you’re about to say something and that small voice inside your head says to stop – Don’t say it!

That’s a good start and can go a long way toward making a pleasant workplace. The problem is, there’s no magic to that little voice. It’s based on the time, place, and culture you’re a part of. And, unfortunately, there have been many times, places and cultures where it was perfectly acceptable to say and do horrible things to marginalized groups. We can believe that we’re somehow different now and trust that voice or we could come up with a better algorithm.

As a start, I recommend checking for that little voice, then checking to see if you’re about to make a statement about 2 or more people you don’t personally know. If your statement fails either of those checks, strongly reconsider it.

It’s also important to watch how you engage with ideas. To illustrate, I’m going to break one of my cardinal rules and talk politics.

This is the Anti-Cat-Photo Party (the ACPP). They’re a single issue party working tirelessly to remove cat photos from the internet. Imagine for a moment you’re at work getting ready to gather for your Tuesday afternoon meeting. In the small talk beforehand, your manager says, “Have you heard about these idiots who hate cats….are they insane? Who wants fewer cat photos on the internet?”

Now, if you happen to be a member of the ACPP, chances are, you are going to bury that secret deeeeep down inside. I’m not a career coach, but, in general, having your boss think you’re an insane idiot is not going to put you on the fast track to promotion. Even if you aren’t affiliated with that party, the message is still clear – disagreeing with the boss is dangerous.

Instead, imagine that the boss says, “I understand that they’re concerned about running out of server space on the internet, however, I wonder if they’re aware that we can spin up new servers? Obviously we could fill infinite servers with infinite cat photos, but I don’t think that fear is reasonable.”

That language is open – it gives other people a place to go in the conversation. The manager still clearly disagrees, but now there’s an actual argument to engage with and you don’t get the sense that you’ll suffer if your opinion isn’t the ‘right’ opinion. One set of language is open, allows room for differing opinions and discussion. The other gives you the sense that if your thoughts aren’t seen as “right” in the organization, you’ll be ostracized.

That’s incredibly powerful for non-traditional team members. Whether they’re 20 years older than the rest of the team, the only member with a kid, the only person that didn’t graduate from an ivy league school (or at all) or the only minority on your team – they already know that they don’t ‘fit’ quite as naturally as the rest of the staff. Creating a team where ideas can be discussed openly, where people are able to disagree without picking up negative labels (as long as they can argue their position well) will serve you and your team well.

MENTORSHIP

The best kind of ship!

There have been a number of studies that show that show mentoring is associated with a wide range of positive outcomes for the mentees. However, the outcomes depend upon the type of mentorship that’s given.

In a recent Harvard Business Review study, the researchers hypothesized that the reason high performing women were still getting fewer promotions than their male counterparts was that they were less likely to have mentors. In reality, they were slightly more likely to be mentored, but it was a very different type of mentorship.

Women were much more likely to get mentorship that involved helping them understand themselves, finding their preferred styles of operating and pointing out behaviors they would need to change as they moved up the ladder.

By contrast, men were more likely to receive a type of mentorship they labeled as ‘sponsorship’ – they’d have mentors that were much higher in the organization than them aggressively put them up for promotions and new opportunities. Their mentors help them plan their moves and publicly endorse their authority. Having a sponsor strongly correlated with achieving higher positions in the company (and, as a result, higher pay).

One huge problem is that for someone to be able to offer the benefit of sponsorship, they have to be in a higher position in the company than the mentee. Since there are fewer women and minorities in management positions, there are fewer of them available to mentor people. And, since people usually mentor folks that remind them of themselves, this leaves non-traditional developers at a distinct disadvantage.

The good news is you don’t have to look like someone in order to be able to mentor them!

This is Dorab circa 2005. Technically Dr. Dorab Patel Ph.D.

And this is Jessica Rudder…well, technically Jessica Fields B.F.A. (but you don’t see me bragging about it). Dorab and I did not have much in common. We can start with the obvious – he prefers Star Trek while I prefer Star Wars. They’re both excellent franchises…but we have our preferences.

Dorab grew up in Northern India. I grew up in the middle of the United States. Dorab got a Ph.D. in computer science from UCLA. I got a bachelor of fine arts from NYU. He’s the CTO and technical cofounder of an engineering company. I was hired to do keywords and ad copy for their clients.

We did not have much in common, but, when I told Dorab I was interested in learning to code, he took me at my word. When the feed file was broken, he taught me how to use FTP to grab the file so I could prase the XML on my own and find the issue. And he wouldn’t let me use Filezilla no matter how many times I insisted it would be quicker to ‘point and click’. “Learn to use the terminal” he insisted it’ll be faster.”

When I needed data that wasn’t available in our standard reports, Dorab spent hours teaching me how to use SQL to write custom queries to the database. Now that I have experience as a dev, it blows my mind that he let me anywhere near the production database. He just trusted I wouldn’t royally screw things up.

When Stanford announced that they were going to be running their Introduction to AI course for free online, Dorab made sure that I was signed up for it. He also talked me down from a ledge when I panicked about whether or not I could do the material since I barely knew Python and I hadn’t touched statistics in almost 5 years.

When I declared that I absolutely hated recursion and that while I understood it conceptually I just couldn’t get it to work in real life, he just nodded in sympathy. The next day, there was a copy of Gödel, Escher, Bach sitting on my desk. For those of you that aren’t familiar, it is over 700 pages about how everything in life springs from recursion. I read that monster and learned a whole lot about recursion. I also learned not to whine to Dorab about hating something I really just didn’t understand.

After I graduated from Flatiron everyone that knew what I was up to emailed me asking if I’d really learned to code, if I could really get a job – and, I want to take a moment here for a shameless plug – Flatiron School has a 98% placement rate in technical roles by 120 days post graduation…those numbers are independently verified and everything and we’re super excited to be expanding to online education this year with our Learn platform. – I understood people’s curiosity, especially since coding immersives were newer at that point. The questions definitely didn’t hurt my feelings….but, something cool happened. I got an email from Dorab that said, “Saw this and thought of you” with a link to this shirt….a programming inside joke.

If Dorab had only thought to mentor a plucky young guy that reminded him of himself, I’d probably be sitting in a marketing meeting right now still wishing I could become a programmer

If you want to have a diverse team, you need to mentor a diverse array of people. And the mentorship needs to focus on helping them reach their goals whether it’s taking on more complex projects or working toward a position in management. If your mentors are spending more time telling non-traditional developers to be careful about their tone, watch what they wear or to smile than they are helping them to level up their skills, you need to revamp your program. If you don’t have a program in place, get one started – everyone, especially junior developers should have a mentor. There’s a lot of great research out there regarding what types of programs work best – I’m happy to share it with anyone that’s interested. Or, you know….Google.

CULTURE

the things you actually do

If you ask the average person about their company culture, you’ll probably get a mix of mumbled, half memorized company values, a link to the “what it’s like to work here page” or a shrug and an “I dunno”. Culture is hard – hard to define and even harder to put in place. One important thing to remember is that simply saying something is your culture doesn’t make it so. It’s not mission statements, lists of core values or ‘who we are’ pages on your website.

It’s literally, the things you do. Does your team pair a lot? What text editors do you use? Do you eat together? Do you work late? Is there beer and video games? Do you have to love React and hate Angular? Is it all sports all the time?

In an effort to distinguish itself from more boring businesses, a lot of tech companies ended up ‘independently’ developing the same cultures:

Ping pong table!

Kegs! Kegs! Kegs!

Video games!

Work hard, play harder! Woooooooo! High five!

If you’re a 24 year old frat boy, it probably sounds like heaven…but, if you’re trying to build a diverse tech team, it’s more like… hegemony.

Now, hold on, before you think I’m all smart and fancy, I had no idea what that word meant before a couple weeks ago. That’s when my Ancient Greek reading husband told me about this disheveled gentleman, Xenophon, and how he used that Greek root to talk about the political dominance of one particular city state within a league of city-states. They were like, ok, league! We’re all Greek here, but, you know, some of us are more Greek than others. Later, hegemony got extended beyond politics to talk about social norms. It’s not that there are rules or that my boss is banging his fist telling everyone to conform or get out, it’s there’s a dominant culture, and deviating from that culture is marked. We tend to overlook this in tech because our dominant culture is fun, and who doesn’t like fun?

Many times that ‘fun’, ‘open’ culture you think you’re developing is not as universally appealing as you might think. Last December I fielded a call from one of our recent grads. She was starting a new job right before the holidays and was extremely worried about the holiday party. She didn’t want to drink at the party but she thought not drinking would be a black mark…Won’t drink at the party? Not a team player. During the recruitment process they’d made such a big deal about their liberal drinking policy that she’d begun to think it was a ‘must drink’ culture. They had me talk to her because I don’t drink. Actually, as my friend Joe put it, I’m a double weird. I don’t drink and I don’t eat meat. And, I know what you’re thinking….who let a non-drinking vegetarian into the country…It happens.

We got things sorted out and she was able to enjoy the party…but not everyone will have a school full of resources they can rely on to navigate cultures they’re not certain about. This makes it especially difficult for members of your team that are from very different backgrounds. If your team culture is built around certain activiites, chances are, non-traditional developers are going to feel left out.

Now, I’m certainly not saying that you need to make sure that every activity is appealing to every member of your team. If there are more than 2 people on your team, that would be impossible. Instead, you should focus your culture around things that are universal to being a good developer. Things like problem solving, well-structured thinking and consuming junk food….ok….maybe not that last one. The other stuff can just be bonus and since it’s not a ‘core part’ of your team culture, people won’t feel pressured to confirm.

I think it’s important to note that this isn’t a zero-sum game. I’m not a bean counter that only cares about percentages….kick a few white guys out and suddenly the percentages are more warm and fuzzy? No! The fact is that there is room for everyone in tech. At current rates, Australia will have 100,000 more tech jobs than tech workers by 2020. In the U.S., they’re predicting a shortfall of over 1 million.

We need all the people that are currently in tech….and all the people that are currently studying tech and then a bunch more people on top of that just to fill all of the upcoming openings. By structuring your team in a way that supports people in reaching their goals, encourages good thinking and will make your team a place that both traditional and non-traditional programmers love to work.

Then we can focus on the fun stuff….like writing code.

Thanks!