ON GETTING OLD(ER) IN TECH

DEC 6, 2016 · Written by Don Denoncourt

After years of scoffing at talk of prejudice in the information technology field – as a white male with good hair –, I'm starting to call prejudice against my being old(er). It's true: age discrimination is a real thing.

Since 2008, the number of age discrimination complaints has grown to around 25,000 a year. Some may argue that everywhere we turn these days, someone is complaining about something being unfair. Alright. Let's not just take complaints into account. But rather, let's look at the average age of IT workers at well–established companies. Facebook: 28. LinkedIn: 29. Google: 30. To put that into perspective, the average age of all U.S. workers is 42. Well above the average age at these companies. Even Mark Zuckerberg once publicly said, at an event held at Stanford: "I want to stress the importance of being young and technical. Young people are just smarter."

By my clock, it's 3 till 60, so, yeah, I'm starting to get touchy about this topic. Having the insight that white – mustache – hair brings, I'm in a good position to debunk the younger is smarter philosophy. Allow me to point out what I've done throughout my decades of work – yes, decades plural – to stay employable, and share with those of you who are getting older (which is everyone), what you can do to avoid being put out to pasture once you pass 30.

NO COUNTRY FOR OLD MEN?

"Younger people are smarter." Balderdash. So that means companies shouldn't hire Sandi Metz, Kent Beck, or Uncle Bob Martin? All 30–year veterans. "They're just anomalies." Not in my experience.

Six years ago, with the help of John Staler, I developed kettleruса.com. I did the Groovy and Grails front–end, and John did the RPG back–end. John is one of the best programmers I've ever worked with. Yet, John has been around for… quite a bit. Let me put John's age into perspective with a quick story: I'd regularly test checkout by buying kettleruса.com's cheapest product: ping pong balls. I'd often forget to cancel my orders, which resulted in the delivery of numerous packets of ping pong balls to my house. Anyway, Captain Kangaroo was a popular children's TV series from 1955 to the mid–70s. The Captain regularly had ping pong balls dropped on him as a prank by his moose friend. One day, I made a Captain Kangaroo reference to my growing stock of ping pong balls. I assumed John had watched the show. But, when I saw that he was confused about the reference, I described the sight gag to him. John responded: "Don, I grew up without a TV. It was the early '50s." That's right, not only was John pre–Node.js, pre–Ruby, pre–Java, and pre–Internet, he was pre–television. Yet, you aren't going to find a better programmer – RPG or otherwise. Not even including the self–proclaimed "smarter" 30–year–olds and younger.

Yeah. OK. So he's RPG. Want an example where the programmer codes in something other than "antiquated" RPG? How about HTML5, JavaScript, and C#? Then, read my blog post on Jim Stanicki, the guy that got me into this field back in the early '80s.

20 YEARS OF EXPERIENCE VERSUS 1 YEAR 20 TIMES

When I hear someone say they have 20 years of experience, I wonder if that's really true or if they merely had 1 year of experience 20 times. I've known too many developers that used the same techniques they learned in their first year of employment for the entire span of their career. I found this to be true in the IBM AS/400 RPG marketplace with 40– and 50–year–olds, but I've also found it to be true with 30–something Java developers. In the early 2000s, I traveled the country teaching Java seminars to RPG developers. I had expected that those RPG developers would be familiar with modern modular RPG programming techniques, but what I found was that most of them were still using old–school RPG. They had stopped enhancing their core skill set – much less acquiring new ones. Then, between 2008 and 2010, I was training Java developers at Circuit City on Groovy and Grails. These folk were mostly late 20s and early 30s, and they were just fine sticking with good–old, write–everything–yourself, don't–bother–with–frameworks, Java.

My point is certainly not that these younger developers were smarter. It's that many programmers let themselves grow stale. And the bigger problem is, after doing the same year's worth of experience ten times, many programmers forget how to learn. Not only can it be extremely hard to catch up with ten years of technology, it can be next to impossible if you've forgotten how to learn.

If you plan on being in the IT field for more than 10 years, you need to be a lifelong learner. I've always been a lifelong learner. I've learned and developed with numerous programming languages, frameworks, and strategies. As a result, I've honed my learning skills. Case in point: I've been told that you can't learn a second language when you're in your 50s. I'm here to tell you: that is false. I started teaching myself Italian when I was 52. Now, I read and listen to novels written or spoken in Italian on a daily basis, and I'm way past "conversational" Italian. And so, for me, learning another programming language or framework is a piece of cake.

ONLY AS GOOD AS YOUR LAST TWO YEARS OF ACCOMPLISHMENTS

Kent Beck has suggested that, with consistent use of pair programming, the capabilities of programmers don't differ much after two years of experience. Understand that this is in an environment where techniques and skills are seamlessly shared. An environment where the secrets of veterans become common knowledge. My take–away, when I first heard this decades ago, was 1) pair and otherwise compare myself to other developers as often as possible, and 2) don't think that my decades of experience guarantees marketability or warrants higher salary.

I often say that I'm only as good as what I've accomplished in the last two years. I could tell you about all my accomplishments over three decades, such as replacing the use of a System/3 punch card system with the AS/400, writing a Cobol debugger, or… Ah, I'm boring you. What you do care about are things I did in the last two years. Things like: learning and developing with ElasticSearch, configuring multiple applications on AWS OpsWorks, setting up Docker for multiple client applications, converting Rails 2.x applications to Rails 4.2, upgrading Ruby 1.9 to 2.2, and making more effective use of Git and, more specifically, GitHub.

Forget my age. I'd be willing to bet that my list of accomplishments in the last two years rivals any 20–to–30–year–old's. The perennial question then is: what am I going to say are my accomplishments two years from now?

HOW DEVELOPERS CAN MAKE THOSE TWO YEARS COUNT

LEARN, ADAPT, AND LEARN SOME MORE

Treat this year as if it were your first year as a developer and assimilate everything you can. Reclaim the energy you had in your first year of coding. Regain the drive you had to prove to yourself and to your employers that you were "all that" for this IT field. Resume reading about tech, playing with new techniques, and persuading others to teach you. Reacquire the excitement of collaborating on newfound knowledge with other developers. Be a lifelong learner and investigate all forms of learning, including:

- Podcasts
  - I like Greater Than Code and The Bike Shed.
- Webcasts
  - Ruby coders continue to love RailsCasts and, I've lately been enjoying Ruby Tapas.
- Magazines and newsletters
  - I subscribe to: InfoQ, Ruby Weekly, DZone Daily Digest.
  - I also subscribe to ThoughtWorks Radar to stay on top of what technologies are hot.
- Online interactive courses
  - I was a Rails instructor for CareerFoundry, but there are hundreds of options available.
- Conferences and seminars
  - Try to get to one conference a year. Understand that you often learn more in conference hallways than you do in the sessions.
  - If you can't get to conferences, take advantage of the fact that many sessions are available online. I recommend confreaks.tv/conferences as it lists sessions from dozens of conferences.
- Blogs
  - Follow a few quality blogs – like the Corgibytes blog and Giant Robots Smashing into Other Giant Robots – where you're notified of new posts.
  - Have your own blog and post regularly. Everyone comes up with solutions to problems or has an opinion that others would benefit from. Plus, writing a blog post solidifies your knowledge.

Also take the time to discover and leverage what kind of learner you are. Do you learn best from books (that's me) or do you need a classroom environment? Are you an audio learner? Whatever your learning style, try to learn something new every day.

NO WORK IS AN OPPORTUNITY TO LEARN

Being between jobs is no excuse for not learning new technologies. You don't have to be employed to get experience. Do your own internet startup. Platform As A Service (PAAS) hosts, like Heroku and AWS, can be free or almost free. Come up with some goofy idea and flesh it out. Put it on GitHub as a public repository so others will see it. Do the whole cycle from backend database and maybe some NoSQL and a simple front–end. Later move that front–end to a single–page app like Ember or React or Angular. Add checkout with credit card processing.

Here's a bad idea for a startup, but one that would give you loads of real–world experience: Virtual Lemonade. Create a site where someone can use their phone's current location to see what lemonade stands are available. That means you'll need Global Information System (GIS) enablement and a database of stands. You'll need server–side code and then client–side. And then, you'll need to provide the ability for kids to register their stand with your service. Maybe give them notifications of a traveller looking for a stand so they can say they are open. You may not become the next Zuckerberg, but you certainly will pick up a full gamut of marketable skills.

BE AND LOOK FIT, BUT DON'T WORRY ABOUT LOOKING YOUNGER

I don't think, as some career counselors recommend, that you should try to look younger with hair dye or plastic surgery. What you should do is exude energy. To remain relevant in an industry that seems fixated on youth, it's crucial to be spirited. And an overweight 50–something wheezing smoker does not suggest vitality.

A year or so ago, I was attending a two–week training session with about a dozen 20– or 30–something developers. The training was on the 22nd floor and, every day after we came back from group lunches, I'd always take the stairs. The first day or two, one or two of the kids would join me, but I got no repeats. It's pretty hard to be considered a has–been when they can't keep up with you.

Don't go overboard just to prove a point. Be yourself. Only, your best self. Fitness can be as simple as a daily brisk walk. Maybe stuff in your earbuds and listen to a tech podcast on that walk. I watch webcasts while on the crosstrainer at the YMCA, and I listen to podcasts while unicycling. I prefer lunch workouts because it clears my head in the middle of the day, and I'm refreshed when I get back at it in the afternoon.

I don't believe you should go to extremes to look younger. I could look younger simply by shaving my white mustache off (the hair on my head is full with only a few flecks of grey). But I earned those white hairs and wrinkles; they're badges of experience. I once had someone say to me: "The eighties called, they want their mustache back." "Well," I quipped, "This mustache is from the eighties, so I'm keeping it."

BE INTERESTING

I don't care what age you are, if you're a couch potato, I guarantee that you will elicit boredom in an interview. Be interesting… to yourself. Everyone benefits from hobbies. It doesn't matter if others think your hobby or passion is odd or a bit off. For example: I keep bees and ride a unicycle. I also know civil war reenactors – which you'd have to pay me to do –, but I find these people fascinating. I'm sure, if you don't already have a hobby, you have interests that you just haven't turned into hobbies. Having a hobby is essentially a fun form of lifelong learning.

BE UPFRONT ABOUT YOUR AGE

I feel that you should provide clear indication to potential employers what your general age is so you can weed out age–discriminatory employers. Do you want to work in an environment where they look at you as dead weight or one that values your energy and experience? Here are the first two sentences from my cover letter to Corgibytes: "Your team is looking for someone with 7+ years software development experience' and a 'Polyglot programmer with skills in 5+ programming languages and 2+ frameworks.' How about 7+ years of C/C++, 7+ years of Java, 2+ years of PHP, and then 3+ years of Ruby (not to mention 7+ years of RPG and Cobol, otherwise, you could do the math and guess my age)." So I gave a tongue–in–cheek account of my age in the first paragraph.

TAKE A CUT IN SALARY FOR NEW OPPORTUNITIES

I have taken big cuts in salary three or four times in my career. I'm talking 10–20 thousand dollars a year. And not because I was out of work. I left existing jobs because I did not see myself growing technically in those positions. I've also turned down big salary jobs because I felt they'd stifle my technical growth. Some of my job changes ended up, arguably, being bad choices as they ended up being dead ends, but I make sure that I always walk away from a project with accumulated and marketable knowledge.

I've seen too many people lock themselves into technologies (like Lotus Notes and Domino) and find themselves unmarketable after spending 10 years in that industry. Even if you have a high–paying salary, don't let the tech world pass you by. Be sure to be up on the latest technologies. And, if you can't do that at your current position, maybe it's time to move on.

BE FOREVER YOUNG

As I mentioned earlier, yes, age discrimination is here and real. Our bodies get old and some – like Zuckerberg – will use that against us. But the biggest mistake would be to compound that by letting our minds and spirit get old as well. That is where we can stay "young."

Bob Dylan said it best:

"May you build a ladder to the stars
And climb on every rung
May you stay forever young"

"May you grow up to be righteous
May you grow up to be true
May you always know the truth
And see the lights surrounding you
May you always be courageous
Stand upright and be strong
May you stay forever young"

"Forever young, forever young
May you stay forever young"

Note that when Bob Dylan released his 33rd album, Together Through Life (which climbed to number one in Britain), he was **68** years old.