Muddling Through The Middle Bits:

What comes after junior but before senior

@northofnormal * Detroit Labs * she/her

Who Am I?

👩‍💻I'm Anne

📱I'm an iOS developer

🐣I'm a former junior developer

@northofnormal * Detroit Labs * she/her

I'm Anne! I work for a company called Detroit Labs, and in fact I'm a graduate of their first apprenticeship program in 2014. I'm not a junior developer anymore, and, in fact, I'd describe myself as very much in this muddling middle bit of my career. And that's why I'm giving this talk, because it's a talk I needed to hear.

Who Are You?

✋Junior Developers?

@northofnormal * Detroit Labs * she/her

So who are you? Who in this room would consider themselves or be described as a junior developer. Awesome. Welcome!

Who Are You?

👍Junior Developers?
🖐Senior Developers?

@northofnormal * Detroit Labs * she/her

Who here could be described as a senior developer?

Who Are You?

👍Junior Developers?
👍 Senior Developers?
✋Mid-career Developers?

@northofnormal * Detroit Labs * she/her

And who here is stuck in the middle with me?

Sharing Circle

@northofnormal * Detroit Labs * she/her

Let's have a sharing circle moment. Those of you who aren't junior developers anymore, when did that happen to you, officially? When did you get the title change or the big raise? Next question: when did you stop FEELING like a junior dev? When did you believe it? Those of you who are in a position to evaluate others, what are the signs you look for that a person is no longer "junior"

How I Stopped Being a Junior Developer:

? ? ? ?

@northofnormal * Detroit Labs * she/her

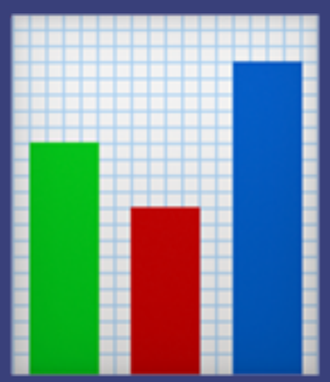
For me, the answer is kind of...I dunno. About a year, year and a half after my apprenticeship ended, the company announced that they were going to evaluate a bunch of the apprentices to see if we were junior devs or had advanced beyond that. So I had a couple of meetings with a senior dev that I hadn't worked with before--maybe three meetings, total. We mostly talked about baking, not coding. And maybe he got feedback from my project team? I'm unclear. And then one day an email went out to the whole company that Anne was no longer a junior developer! Great news! But...I had no idea what criteria were used, if I had passed top of the class or just barely, and I still felt like I was one wrong move away from setting the entire code base on fire... I didn't feel like I had crossed any kind of threshold, and it was probably a solid year of me basically denying that I wasn't a junior dev anymore, to myself and to others.

Why Are We Here?

@northofnormal * Detroit Labs * she/her

Like I said, this was a talk I needed to hear, a talk with advice I've had a hard time finding. I've been to great talks for junior developers, great talks about the challenges of being a senior developer, but not many bridging the gap between the two. This is my attempt to do that.

The Dip



@northofnormal * Detroit Labs * she/her

We do a quarterly engagement survey at Labs, asking a bunch of questions of people across the company, and we've started to notice something interesting.

 I'm generally happy at my job

@northofnormal * Detroit Labs * she/her

At about 1-2 years of time at Labs, general happiness decreases

📉 I'm generally happy at my job

📉Someone at Labs cares about my success

@northofnormal * Detroit Labs * she/her

And the feeling that someone else at Labs cares about your success drops off as well. This one was even bigger than general happiness.

📉 I'm generally happy at my job

📉Someone at Labs cares about my success

📉I'm appropriately recognized and rewarded

@northofnormal * Detroit Labs * she/her

And things really crater when it comes to recognition and reward.

But Why?

@northofnormal * Detroit Labs * she/her

What's this about, though? We found that these dips correlate pretty closely with mid-career devs, particularly those who have just passed out of junior-dev-hood. And it's a U-shape, not a cliff. These measures all start to climb again as developers get further down the career road.

A Clue!

@northofnormal * Detroit Labs * she/her

And that's a clue, that we see this fall in early-mid-career devs and then see it start to rise again. Let me explain.

What does a career projection look like? It's kind of a linear thing, right? You start out, you hit some milestones, and eventually you retire to Fiji, right?

@northofnormal * Detroit Labs * she/her

So you start as a baby dev, and if you are in a good environment, you are getting a lot of encouragement and positive feedback. People are generous with their time and encourage your questions. Incidentally, if there are any juniors in the room who aren't in this kind of environment, Detroit Labs is hiring! But if you are, this is all very encouraging and empowering. Granted, most of the time you spend as a junior dev is terrifying but ideally, you are in a state of constant learning and constant, recognized, rewarded growth.

But sooner or later, you outgrow that, right? Someone says "Congrats, you're promoted!" and they shake your hand and you get a raise and then...what happens? Well, for me, people stopped patting me on the head and telling me I was doing great! They got less high-five-y about me closing cards or accomplishing a thing. I was just expected to be competent.

From 

To 

@northofnormal * Detroit Labs * she/her

Not only that, I was SUPER GREAT at being a junior developer. I asked all the questions, I paired on all the things, I didn't let anything scare me, I left my ego at the door, took all the PR comments and asked for more. By the time I was having those meetings with that senior dev, I was on the mountaintop of junior-developer-dom. I had climbed the cliffs of not knowing anything to the pinnacle of being not entirely terrible! And that moment of triumph, that moment when someone somehow decided I wasn't a junior dev anymore, just meant that I had been sent down to the bottom of the next mountain. I was kinda not entirely terrible...in the company of peers who were expected to be competent. Pretty disheartening.

🤔 This Is Interesting

➡️ Junior Developers

➡️ Senior Developers

➡️ ...other?

@northofnormal * Detroit Labs * she/her

So I like to pay attention to words, the words we choose, the words we use, the implications and connotations of various words. Why some people are called confident and others are called pushy, that kind of thing. And something I noticed early on is that we have catchy and descriptive names for people starting out in their careers, who are in learning mode and just launching their careers. And we have a great name for well-established people, the ones who are expected to have a deep well of insight and experience to draw upon. But we don't really have a great term for the space in between, do we? For those of us who aren't junior but aren't senior, either. We're defined by negative space in a way, aren't we? In terms of what we are NOT instead of what we ARE.

Survey Time

@northofnormal * Detroit Labs * she/her

What do you call these people? Does your company have a good title, or do you use some great term yourself?

Does !Junior == Senior ?

@northofnormal * Detroit Labs * she/her

I don't think anyone really believes this, that you jump right from junior to senior. But there's no catchy, universally understood for term for this mid-career space. And that leads to a subtle perception gap. People who still feel very new and unsure about what they are doing are thrust out of this safe, learning, junior space. I know that I spent about a year after the official announcement of my promotion out of junior status denying and rejecting the idea that I was anything but a junior dev. I kept that junior mindset of asking questions, doubting myself, double-checking everything, and observing more senior devs as they made architecture and other decisions.

 "Speak up more"

 "Don't be afraid to argue"

 "We didn't get your input"

 "Be more comfortable making and challenging decisions"

@northofnormal * Detroit Labs * she/her

This is the feedback I got from my colleagues on the very next project I worked on. Notice a pattern?

Junior devs find power in questions

Mid-career devs find power in answers

@northofnormal * Detroit Labs * she/her

My ability to ask good questions and the lean into ignorance as a path to knowledge is what made me a great junior dev. But almost immediately upon entering the mid-career space, these same strategies started to hold me back. It was time for me to stop observing those conversations and start participating in them.

Time to Release the Suck

@northofnormal * Detroit Labs * she/her

My friend Ellen Mey has a great talk about being a junior dev that she calls "Embracing the Suck." It's about the idea that to succeed as a junior dev you just need to lean into how hard things are and how stupid you feel. I tell junior devs all the time, "No one expects you to be any good at this." Being a junior dev is explicitly about GETTING BETTER not about really being any good right now. How many times have you seen--or been--a jr dev screw themselves and the project by hiding their ignorance, not asking for help soon enough? But mid-career devs can screw themselves and the project by not letting go of the suck. By continuing to doubt themselves and not putting the bit of experience that they do have to work.

Let your questions change

@northofnormal * Detroit Labs * she/her

Which isn't to say stop asking question--never stop asking questions. That's life advice, not career advice. But I have found that the questions I need to ask have changed.

@northofnormal * Detroit Labs * she/her

Let me explain what I mean. I've found that the questions I asked a junior developer had answers. How do I change the background color when the user is logged in? How do I parse this data? How do I handle that kind of error? For the most part, you can google the answers to junior dev questions. But the questions I face now...don't have answers. What's the best way to translate this messy API into neat data models? What's the architecture pattern that will fend off the most tech debt while the project scales?

In general, this is the pattern I see. The junior devs want to know how to do the thing, the midcareer devs want to know what's the smartest way to do the thing, and the senior devs wonder why we are doing the thing in the first place. Junior devs are looking at each individual tree in the forrest, and they should be limiting their field of view a this point. Midcareer devs need to look at the entire forrest, at how each tree interacts with all the others and also with the streams and the squirrels. Senior devs need to step even further back and look at the entire ecosystem and why we need trees in the first place. I think this progression is important. I think the HOW questions I asked as a junior are informing the WHAT questions I'm asking now. I'm remembering the things that seemed confusing or complicated to me as a junior and now I'm questioning the value of that complication. Were we working against a constraint that required tricky workarounds, or was it a case of someone being clever in code? And I expect that these WHAT questions will inform the WHY questions I start asking as I become a senior developer.

The Bank of Experience

@northofnormal * Detroit Labs * she/her

There's an aspect to the mid-career space that's about building and becoming comfortable spending from your bank of experience. Think about just-promoted me on that new project. I had just spent almost two years asking constant questions and getting constant answers. I made a bunch of mistakes, and learned a bunch of lessons. I had already observed and interrogated senior devs around me as they made decisions. That's my bank of experience. That new project needed my voice but I held it back.

Time

@northofnormal * Detroit Labs * she/her

And this is where just the passage of time comes in. At the beginning, your bank of experience is going to be kind of poor. You'll only have in it what you learned as a junior, a lot of how but not a lot of what. But the experience bank is kind of opposite world banking. The more you spend from it, the richer it gets. This is kind of frustrating but...you will need to make mistakes. You'll need to make the wrong call sometimes, and this is key, you'll need to learn from it. And while this process takes time, you aren't sitting there watching the calendar go by. You are taking control of situations and shaking lessons and greater understanding out of them. You will need a certain amount of temporal space to have these experiences. Time-based gatekeeping isn't helpful here, you need to be using this time wisely.

You Need To Screw Up

🅰️ worked before, so it will probably work here

@northofnormal * Detroit Labs * she/her

How many times have you run through this cycle with a framework or pattern or particular tool?

You Need To Screw Up

~~🅰️ worked before, so it will probably work here~~

🅰️ was a disaster last time we need 🅱️ instead

@northofnormal * Detroit Labs * she/her

You Need To Screw Up

~~🅰️ worked before, so it will probably work here~~

~~🅰️ was a disaster last time we need 🅱️ instead~~

Nevermind, 🅱️ was terrible, we need... 🆎

@northofnormal * Detroit Labs * she/her

You Need To Screw Up

~~🅰️ worked before, so it will probably work here~~

~~🅰️ was a disaster last time we need 🅱️ instead~~

~~Nevermind, 🅱️ was terrible, we need 🆎~~

...Okay sometimes 🅰️, sometimes 🅱️, it depends

@northofnormal * Detroit Labs * she/her

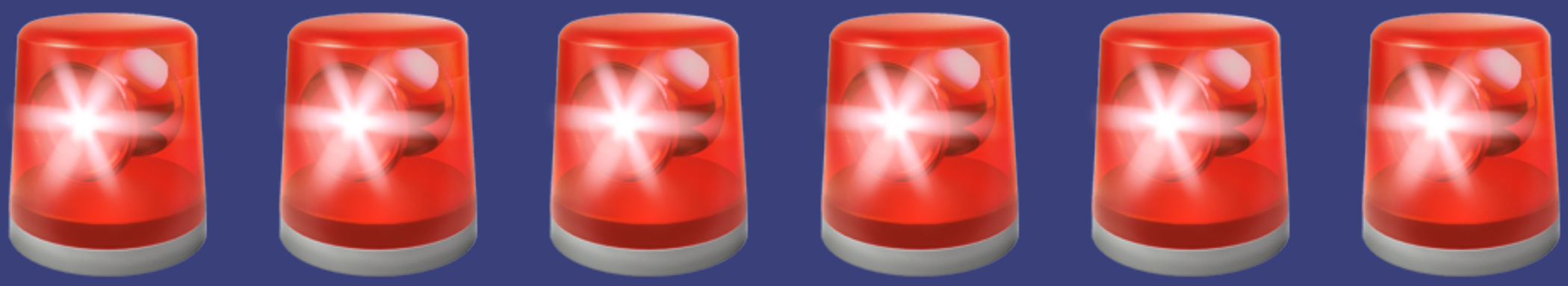
The Bank of Experience

@northofnormal * Detroit Labs * she/her

That's all cash in the bank of experience. Hopefully, if you are being thoughtful, if you are paying attention to each of the cases where these tools didn't work, you'll realize the situations where A is going to work better than B and vice versa. And the next time someone asks which to use, you can say, "Well, it depends, tell me more about..." And this kind of thing can get super frustrating because, well, who here has driven through Ohio?

@northofnormal * Detroit Labs * she/her

It doesn't matter if you driving north-south or east-west in Ohio, you are going through flat, boring farm country. Driving through Ohio feels just like not moving at all. It's been three hours, and you keep seeing the same tractor, the same corn fields, the same three cows. It's like being stuck in an endless, boring, loop. You are making progress, your odometer shows that, but woah does it not feel like it. And this process of screwing up, learning a thing; getting it right, learning a thing, this constant stream of mistakes and successes and insights that you are piling into our bank of experience can feel very much like not moving, too.



@northofnormal * Detroit Labs * she/her

And that leads to one of the particular dangers of the mid-career space.

This is when we lose

women

and

people of color

@northofnormal * Detroit Labs * she/her

Because for all the talk about the leaky tech pipeline, this is where that pipeline really bursts.

32% of women in tech feel stalled in their career and plan on leaving within the next year.[1]

[1] Hewlett, Sylvia Anne and Lauren Sherbin. Athena Factor 2.0: Accelerating Female Talent in Science, Engineering, and Technology. 2014, talentinnovation,org

@northofnormal * Detroit Labs * she/her

48% of black women feel stalled in their career.[1]

[1] Hewlett, Sylvia Anne and Lauren Sherbin. Athena Factor 2.0: Accelerating Female Talent in Science, Engineering, and Technology. 2014, talentinnovation,org

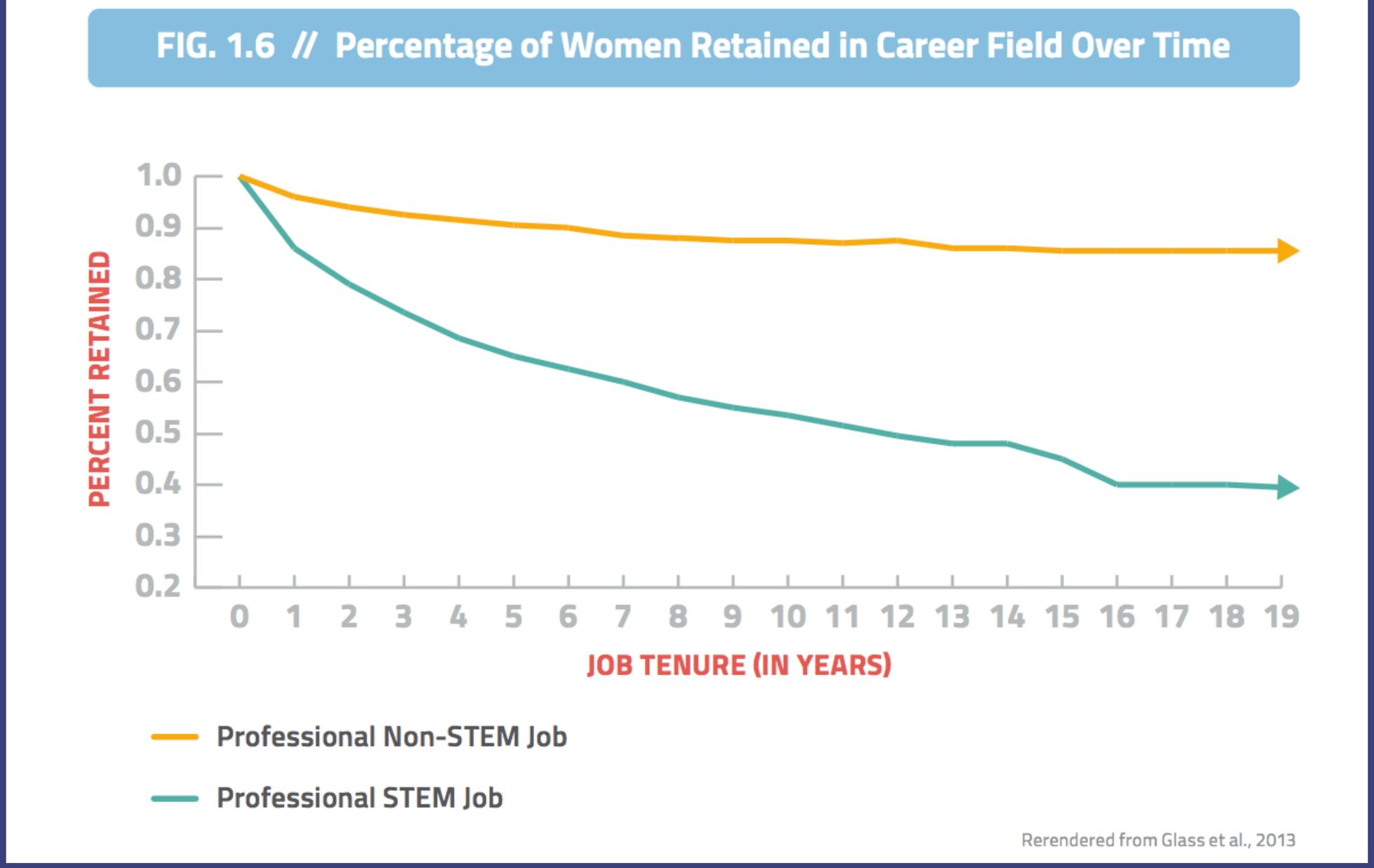
Think about that. Nearly half of all black women feel that their careers aren't moving forward. That's so far beyond statistically significant it almost has to be deliberate. Nearly a third of all women, and half of black women, feel like their careers are not moving forward. That they have reached a certain point and they are stalled or blocked from moving forward.

56% of women will leave their tech career during the middle career stage[1]

[1] Hewlett, Sylvia Anne and Lauren Sherbin. Athena Factor 2.0: Accelerating Female Talent in Science, Engineering, and Technology. 2014, talentinnovation,org

@northofnormal * Detroit Labs * she/her

Given the previous stats, this one can't be a surprise. Over HALF the women in tech never make it out of this middle bit and bail on tech entirely. And before we entertain any nonsense about how women somehow in large numbers are just not interested in working in tech, this same study found that 80% of women surveyed "love their work." And yet over half of them will leave.



Take a look at this chart. This is the percentage of women retained in their career fields over time. There's a slight dip in non-STEM careers, but for the most part it remains over 90% over two decades. Percentage retention of women in STEM careers drops below 90% in the first YEAR and just keeps dropping. One year in and 10% of women are out. But 80% of them love the work that they do.

57% of all professional occupations

25% of all computing occupations[1]

[1] Hewlett, Sylvia Anne and Lauren Sherbin. Athena Factor 2.0: Accelerating Female Talent in Science, Engineering, and Technology. 2014, talentinnovation,org

@northofnormal * Detroit Labs * she/her

And we can't afford to loose women in the industry. Overall, women hold just over half of all professional occupations, but only a quarter of all computing-related occupations.

👩🏽‍💻 3% Latina women

👩🏿‍💻 1% African American women[1]

[1] Hewlett, Sylvia Anne and Lauren Sherbin. Athena Factor 2.0: Accelerating Female Talent in Science, Engineering, and Technology. 2014, talentinnovation,org

@northofnormal * Detroit Labs * she/her

As always, it's worse when we look at women of color. 3% of computing occupations are held by Latina women. 1% are held by Black women ONE PERCENT. And of that 1%, 80% feel that their career is stalled and will not go forward. And you are all great people, so you are thinking HOLY CATS we have to fix this! Maybe you are thinking of the great women and people of color you work with--and if you don't work with great women and people of career, Detroit Labs is still hiring--and how much you want to keep them in the industry. Maybe you are thinking about mentorship programs, pairing them with more senior colleagues who can guide them through...

@northofnormal * Detroit Labs * she/her

Bad news for you

Mentorship Doesn't Work

@northofnormal * Detroit Labs * she/her

@northofnormal * Detroit Labs * she/her

A recent study found that when women receive mentorship, they get advice on how they need to CHANGE, whereas men who receive mentorship get public endorsement of their authority--amplification--and concrete steps to make career moves. Women get told to BE better, but men get told how to DO better. And this has longer-term consequences, in that men who are in mentorship pairings are more likely to be promoted than women in mentorship pairings are.

Stereotype Threat

@northofnormal * Detroit Labs * she/her

Who here has heard the term "Stereotype Threat" before? It works like this: you bring in a group of women and give them, say, a math test, but you tell them, "Women tend to do less well on this test, but we've chosen you specifically because we expect you to beat the odds" they will do less well on the test compared to a control group that is not told that. Just introducing the idea that a group will perform a certain way can make it so. There's a great book by Claude Steele called "Whistling Vivaldi" about this effect, and how it shows up across many different axes (sprinters, Asians vs. Women at math)

The Call Is Coming From Inside The House

@northofnormal * Detroit Labs * she/her

This is all so scary and complicated, because what it says to me is that all this talking about how hard it is for women and people of color to succeed in tech...is reinforcing the idea that it's hard for women and people of color to succeed in tech. I'm a big fan of naming a problem and saying it out loud. We can't address the things we can't talk about, but apparently there's a danger in talking about some things too much?

@northofnormal * Detroit Labs * she/her

I don't know how to fix this. I don't know what to do. It's a complicated problem made worse by a small group of noisy assholes who don't see it as a problem. I'm not here to debate them, but they don't want a debate anyway. All I know is that there is something about the post-junior-developer phase of a career that is particularly perilous to people who are already under-represented in tech.

@northofnormal * Detroit Labs * she/her

One thing I've started doing is trying to keep track of my accomplishments, especially the small ones. Learning when you are a junior happens in leaps and bounds. There's literally things you know how to do today that you didn't know yesterday. Hitting the mid-career, it's more about gaining subtler degrees of understanding. If you keep looking for the big wins, for the thing you couldn't do yesterday, you are going to be disappointed. The BGSU football stadium and that weird touchdown Jesus aren't super exciting landmarks, but they are signs that you've gone some distance. I've started a slack channel called weekly-wins for myself and some people I'm mentoring, and every Friday the slackbot reminds us to post one thing we've done this week that we are proud of. I've heard of other people setting aside time each week to write up their accomplishments, so they have a running log of small wins and incremental, subtle progress.

And don't neglect your "team skills." Some people call them soft skills, but that's nonsense. Dealing with other people is way harder than just code. Are you a good leader? How are you working with the juniors on your team? How do you interact with the client or product owner? How well do you understand the business side? How are you at giving feedback? These things are skills that take as much development and conscious practice as any architecture pattern or new syntax. At Detroit Labs, our job tiers specifically call out the "Peer Supporter/ Influencer" and "Company Contribution" skills and experiences we want senior developers to demonstrate.

vs.

@northofnormal * Detroit Labs * she/her

There's generally a clear map for getting out of junior-dev-hood, right? It's not universal, there's probably a couple of them, but you know what? I remember reading lots of great "10 things every junior iOS needs to know" articles. And it was all solid, concrete stuff: Core Location. Protocols. Testing. It was a lot of google-able HOW stuff, complete with tutorials. Now I'm in a forrest of infinite possibilities that I have to find my own way through. There are no tutorials for a judgement call. That's scary, and it's hard. And I know I can't do it without a lot of support and I know that a lot of people out there, great developers who love the work they do, who are going to get lost in this forrest, or trapped in it, and leave entirely.

From 

To 

@northofnormal * Detroit Labs * she/her

And that's why I think another key part of this middle career path is to start sharing knowledge. Like the bank of experience that grows as you spend from it, I'm finding my way forward by guiding others. Now that I know a little bit, it's time to start cementing that knowledge by sharing it with others. There's a bunch of ways to do this: give a conference talk, speak at a local meetup. You can start writing--if you have a company blog, I promise you that your marketing person wants your contribution. You can start mentoring, making yourself available to guide others. There's open source contributions, building the tools we all use.

@northofnormal * Detroit Labs * she/her

I mentioned that weekly-wins slack channel earlier, and I think that gets at one of the ways we can help each other through this forest. We can make sure we're amplifying each other's accomplishments. We undervalue our own stuff, that's a human thing, but we're always pretty cool at noticing our friends successes. Make sure that gets passed around. Call out the cool thing you saw a coworker do at your next retro. Tweet about their conference talks or open source releases. If you've noticed that a junior colleague is starting to ask more complicated questions--that shift I talked about earlier--let them know, and let whoever keeps track of their progress know. Remember what I said earlier about the differences in the kinds of mentorship people get. Amplify everyone and you dodge this trap and help make the entire industry better. I am a world champ at low self esteem, let me tell you. But I helped organize self.conference this weekend and even I had to admit that I did okay after listening to people talk all weekend about what a great job we did.

Junior Devs need to build the trust of others

Mid-Career Devs need to build trust in themselves

@northofnormal * Detroit Labs * she/her

When you are a junior developer, you need to build the trust other people have in you. You need to make sure they can trust you to accomplish the tasks you set out for yourself, to ask questions when you need to, to generally be relied upon to do the job. Once you pass into your middle career, though, you need to start trusting yourself. And I mention this here because I think sharing knowledge is a great way to build this confidence. Earlier this year, I gave a workshop on building a small augmented reality app in Swift. After the workshop, someone referred to me as an "expert" in ARKit and I actually laughed out loud. I told a friend about the exchange, and she reminded me that an expert is one that has expertise. And I may not know everything there is to know about augmented reality in iOS frameworks, but I know enough to walk about 40 beginners through a small AR experience. Just because its not everything doesn't mean it's nothing. And sometimes you have to do the thing before you can believe you can do the thing. And that not-believing can hold you back. You won't speak up. You won't learn the lessons you need to from your experiences. You won't help anyone else grow. You might get frustrated and leave the industry.

@northofnormal * Detroit Labs * she/her

When I was a wee baby developer back in bootcamp, one of our instructors told us that there was two ways to 10 years of experience. One is by learning and growing, tackling new challenges, being thoughtful about your mistakes and successes, and the other is to keep doing the same thing over and over again, maybe a little faster or slightly better. And I didn't get that for a long time, because at that point it felt like I was learning something new and exciting every day.

@northofnormal * Detroit Labs * she/her

But this is the root of those paradoxes of enriching the bank of experience by spending from it and finding our way by guiding others. What I was learning everyday as a junior was SIMPLE stuff--not easy, by any means, but all those HOW questions that have answers. The things we need to learn to keep moving through the middle bit are DIFFICULT things--which isn't to say that they are hard, or even easy sometimes. But they are complex, complicated things. Things like making a judgment call and standing by it. Owning your mistakes and taking the time to find a lesson in them. Speaking up and speaking with confidence. Putting your knowledge on display. Teaching others the things they'll need to succeed. Helping build the tools our entire community uses.

How do?

@northofnormal * Detroit Labs * she/her

Sow, how do we do it? How do we muddle through the middle bit and make in unscathed to the ranks of senior developers?

Release The Suck

Change Your Questions

Build Your Bank Of Experience

Drive Through Ohio

Guide Others

@northofnormal * Detroit Labs * she/her

Release the Suck: get out of learning mode and into doing and deciding mode. Change and expand the questions you are asking, and get comfortable with ambiguous or contradictory answers. Build our bank of experience by making decisions and learning from them, even the "right" ones. Drive through Ohio and learn to see the signs that you are making progress. Make sure your work through this forrest of the middle career is helping others move forward as well.

@northofnormal * Detroit Labs * she/her

But what do I know? I'm in the middle part of my career myself. Most of this is just reflection, some research, some conversation with senior developers who have made this journey themselves. It's hard to see the path clearly when you are still on it. I'm interested in what the view looks like from your spot on this journey.

northofnormal

@northofnormal * Detroit Labs * she/her