

Navigating The Rust OSS Community

Sep 12, 2019

Navigating The Rust OSS Community

DISCLAIMER

This blog post is the textual version of the talk it was written to prepare for ([video](#)). The content should be essentially the same, though with less stumbling over my own words, less silly hand gestures, and less crying. Please Enjoy.

Preface: Expectations

Are you a new Rustacean trying to take your first steps into Rust's open source community? Are you trying to meet people in the Rust community but don't know where to start? Are you a maintainer interested in attracting new contributors? Then have I got the talk for you! Together we will explore some approaches to find issues to work on, seek out new projects, structure projects to help new contributors and meet other Rust developers to become happy and successful contributors in the Rust open source community.

Okay, Cheesy informercial-shtick aside, why am I up here? I'm here to talk about getting involved in the Rust open source community. My goal is to help new contributors and established maintainers alike. Hopefully by the end of this talk you will have a better idea of how to navigate our community and know how to best help those around you navigate it as well.

One thing this talk is not is a guide on how to learn Rust, the language. I'm not going to talk about how I learned the language or give any tips on how to improve your Rust skills directly. There are many other wonderful resources out there for how to learn rust. I'm here to help you find those resources, not to be one of them. This talk is about the community side of rust, not the technical side.

So A little background about myself. My name is Jane Lusby, I'm a software engineer working on distributed storage at a virtualization company called "Scale Computing". I work professionally on C++ and have for about 6 years now. I was introduced to Rust in the beginning of 2018 at my work's maker night, I fell in love, and quickly decided that I wanted to get involved in the

community at large, give back, learn more, etc. Now, I'd wanted to get involved in open source for a long time, but with other languages it had never really clicked and I'd never managed to get involved.

So what did rust do differently that actually got me involved where other communities had failed? I'm glad you asked...

1. Code of Conduct

One of the first things I saw when I looked into Rust was its Code of Conduct. Now let me give you a little extra background about myself, I'm a woman, I'm gay, and I'm transgender. When I go looking at participating in a community misogyny, transphobia, and homophobia are all things I worry about.

So when I looked at the Rust community's code of conduct and how it banned discrimination on the basis of gender, race, ethnicity, age, disability, and other similar attributes, it was like a huge weight had lifted off my shoulders. It's wonderful to be able to go into a community and know that if you run into a bigot that pretty much everyone has got your back. I didn't feel afraid to get involved because the code of conduct gave me a sense of safety. Thankfully I've not had any problems in the last 2 years, and I'm convinced it's because the CoC as well. The code of conduct acts as preventative medicine and discourages the people its designed to protect against from ever joining your community in the first place. If you want people like me to get involved in your projects just mentioning that you abide by the Rust community's code of conduct goes a huge way to making us feel comfortable enough to engage.

So the code of conduct got me excited about rust, but it's not what got me engaged. What was it that actually got me to do my first open source contribution.

2. This Week In Rust

Incase you haven't heard of it, the Rust community maintains a weekly newsletter of Rustlang news called [This Week In Rust](#). In it you'll find sections on recent blog posts, the crate of the week, call for participation, changes merged to core, upcoming meetups, job postings, and usually a funny little quote at the end.

I came to TWIR because of the blog posts, I was obsessed with rust and wanted to absorb every bit of information about it that I could, but the section that ended up having the greatest impact was the Call for Participation section.

The CFP section of TWIR is a list of good first issues and issues that maintainers need extra help on. As I said earlier, I'd wanted to contribute to open source for years but I had never succeeded in finding an opportunity that actually got me to engage. I think the main issue here

is that my ADHD brain makes it hard to dive into the unknown without a strong motivation, “later” it always says.

Then along comes TWIR, just waving the flag of friendship weekly. “Heres something you can do to help” it would say to me. The weekly list of friendly first issues I could get involved in made it trivial to find issues that were available to work on and approachable for someone new to the code base. I didn’t have to do anything now to find work, it just walked up to me on its own whenever I went to read the news.

So I got involved, I did a minor issue on [rustc](#), just cleaning up some test code, and all of a sudden the problem space wasn’t so scary anymore.

One thing I want to note though is that This Week In Rust depends on community participation to function. It used to be the case that the maintainers would go and hunt down a nice set of good first issues for each week, but they stopped doing that a while ago because they didn’t want to unfairly favor any single part of the rust ecosystem. As a result, if you look at old newsletters vs more recent ones, you’ll see that we consistently have far fewer good first issues now adays. If you are a maintainer of any rust open source projects and you’re tagging good first issues and not submitting those issues to TWIR, you should be, it can make all the difference for new rustaceans who are intimidated by the process of getting involved in open source.

2.1. Friction

So I’m no longer scared of the process, I dive in and started trying to find my own new issues to work on from the issue boards, and unfortunately this is where I ran into my first few bit of friction.

As it turns out, github issue boards can be pretty difficult to navigate, <https://www.rustaceans.org/findwork/starters>, the recommended starting point in TWIR at the time, was particularly hard to work with. It only shows the issue name, summary, and tags. I would go into issue after issue only to find someone else was already working on it. The friction was discouraging, so I switched to only using the github issues lists themselves. There at least I could filter out any issues that had comments in them, which admittedly isnt the best solution, because it’s common for people to open issues on a repo and then for the maintainer to come along and leave a comment with mentorship instructions. I missed all those opportunities due to noise on the issue boards.

The next issue I ran into was maintainer responsiveness. I remember once shyly leaving a comment like “I’d like to take on this feature” and waiting a day or two for a response and getting nothing then quietly deleting my comment and moving on to find a new issue. Now I don’t want to complain here because I think the Rust lang contributors are fantastic people who

do great work for little thanks and no pay and I don't think it's appropriate to expect work from them. But... that doesn't change how it feels when you're trying to get involved and all you hear is crickets.

I ended up settling on a few repos, clippy, rustfix, and cargo, where the maintainers there were particularly responsive and friendly. Shoutout to [@killercup](#), [@ManishEarth](#), [@oli_obk](#), [@phansch](#), [@kennytm](#), [@flip1995](#), [@ehuss](#), and everyone I'm forgetting for making it so easy to get involved and contribute to your projects.

3. Meetups: Making Rust Friends

The next biggest breakthrough for me came with the discovery of Rust meetups. I had no idea these existed, but a coworker of mine told me I should check out the meetup app and we immediately signed up for the bay area Rustaceans group and started going to all of the meetups.

For a while nothing really came out of these meetups OSS wise. I would go to meetups and listen to the talks, eat the free food, maybe talk to a person here or there, but I'm really shy around people I don't know, so I mostly kept to myself or talked with coworkers I'd managed to drag along. I was enjoying myself, and I kept going, but it wasn't until the meetup at Google's office that things Really Started Happening.

What happened at the Google meetup you ask? Manish, our wonderful meetup organizer, walked up to me, unprompted, and asked "Hey, you're Jane right?". I was shocked, how the heck did Manish know who I was? It didn't feel as though I'd done anything worthy of notice, and yet here he was asking for me by name. I replied that I was and he thanked me for my contributions to clippy, and told me if I wanted to do more to let him know, and that he would help mentor me. *(editors note: this is where I start crying)*

This simple act of walking up to me, thanking me for my contributions, and offering to help me do more has without a doubt had a larger impact on my involvement in the Rust community and my life over the last year than any other event in the last two years. I don't think I'd be here doing this talk, or that I'd have gone to RustConf, or that I'd be giving a talk at COGoldRust, or that I'd even have started using Twitter and met so many more people, if it weren't for meeting Manish that day.

Now that wasn't the only thing that happened that day, nor the only person who had a huge impact on me. Eliza Weisman gave a talk on her library `tracing` which was super fascinating and extremely relevant to my personal and professional codebases, and Adam Perry invited me to sit with him when I was shyly sitting by myself in the corner after the meetup hoping someone would come talk to me and introduced me to Isabelle Muerte and Rain. Together all of these people essentially became my first "Rust Friends" and meeting them was the first time it really felt like I was part of the Rust community. *(end crying)*

I ended up tracking Manish down online and sent him a message before the meetup was even over that I was absolutely wanted to get more involved and that I wanted him to be my mentor, and he's still my Rust mentor today.

The big take away here is that meetups are an amazing way to get involved and to feel involved in a community. There are so many amazing rustaceans that go to all these meetups and conferences, so they're a great opportunity to meet and learn from amazing people. If you're not already going to meetups, you should seriously consider starting.

4. Mentor

So with a new mentor backing me up I went home more inspired than ever to work on open source contributions. Initially, I think I had a bit of a misunderstanding of what a mentor is. My hope for mentorship going into it was to have someone who would find me stuff to work on and help me stay on track. Essentially, a manager. Now Manish did his best to do that for me, but in retrospect it was a lot to ask, being a manager is a job in it's own right, and the real value of the mentorship Manish gave me is so much more.

One amazing thing he did was be a constant source of encouragement. I remember talking to Manish about the `tracing` repo and how I wanted to talk to Eliza and get involved, but I was completely intimidated by her and was way too afraid to approach her. Manish on the other hand was like, "No, come on, she's great, talk to her, she's super friendly and will be a blast to work with." He wasn't wrong. I've probably learned more about Rust the language from doing PR work on tracing than I have from everything else in the rust-lang organization combined. I probably wouldn't have talked to Eliza or gotten involved if it weren't for Manish gently encouraging me to take the leap.

Another thing he did was give constant good advice. I would often go to him to complain about features I wanted that didn't exist or talk about the bikeshed of the moment, in this case `await` syntax. He encouraged me to write a little blog post explaining why I was hoping for postfix macro to be the syntax of choice. I felt silly and underqualified doing it at the time but now, having done it, I no longer feel silly or afraid of writing blogs and I am eternally grateful for that.

Speaking of complaining about features I wanted, Manish fundamentally changed how I approach open source work. Initially my strategy was to go to repos with issues open for other people to work on and just do them. Manish on the other hand encouraged me to take those features I wanted, open issues for them, and implement them instead of random issues. Let me tell you, working on features that you personally want rather than random ones other people want is WAY more satisfying. I've found it much easier to be motivated on the features I want rather than ones I ultimately don't care about. It seems like a more sustainable approach to open source and as a result I've contributed to a ton of random repos I use just because I wanted a little feature here or there. It makes me feel warm and fuzzy inside using those features I

implemented myself or thinking about those projects and the maintainers I've had the pleasure to work with, All thanks to Manish's mentorship.

I believe that mentors are fundamentally an amplifying force on growth. If you're a new rustacean trying to get more involved, find yourself a mentor! And if you're an established rustacean and have the bandwidth, you should seriously consider becoming a mentor. It's good for the community.

4.1. Mistakes!

Unfortunately, with all of this new found motivation I got a bit ahead of myself and suddenly ran into a few more snags. I got a little overzealous and opened 3 prs on cargo at once to add "minor" features that I had been wanting for ages. Turns out a few of them weren't so minor! I suddenly had this mountain of work facing me that I felt like I'd committed to and it paralyzed me for a couple of weeks. It didn't seem approachable, and so my ADHD brain went into full avoidance mode. The solution here ended up being to know my limit and to break my tasks up into smaller more approachable tasks. Once I realized, hey I can shelve two of those issues and just work on the one I care most about it was easy to get going again.

Another problem I ran into was organization. I had all of these ideas and features I wanted but I was pretty much just remembering them, talking about them when they came to mind, etc. A fair number of ideas were definitely lost in these early days. Eliza ended up being the one who helped me through this problem. Any time I mentioned a feature I wanted she told me to open an issue, and eventually it became a habit. Bonus points, now people will occasionally pick up issues that I opened for myself and hadn't yet gotten around to doing. Who doesn't love free features!

Now that I had all these issues laying around that I wanted to work on I needed a way to keep track of them all. The solution for this ended up being inspired by David Tolnay's usage of docs.rs as a blog platform. Instead of docs.rs I asked myself if I could use an empty github repo to track my issues. I initially started with a single tracking issue where I would link every other issue, but that notified everyone that I was mentioning their issues so I ended up settling on a project board, which didn't do that. Now I have a nicely organized list of every issue that integrates seamlessly all the different repos I'm working with and I can share this list with people if they're looking for things to work on!

5. Twitter

The next big game changer was learning to love Twitter. Manish and Eliza unintentionally got me into it, I think the main leaping off point was when Manish convinced me to write that await blog post and announce it on Twitter so he could retweet it. I'd never been able to "figure out Twitter"

before this but suddenly it clicked and I realized I could use Twitter for Rust! I started to follow every Rustacean I could find and quickly learned to navigate Twitter effectively.

Now, I've been an avid Redditor for a solid 10 years, and it has been a great way to keep on top of new emerging features, libraries, and blog posts in the Rust community. Twitter is also a great source for this content, in this I think the platform you pick makes no difference. The real difference is how you interact with people on Twitter vs Reddit. Discussion on Reddit is better for impersonal discussion, where as Twitter tends to encourage more personal discourse. It's much easier to make friends and have consistent interactions with people on Twitter than it is on Reddit, which is exactly what you need if you want to participate in a community. Case in point, I've made dozens of rust friends on Twitter, many of whom are in this room today, and I'm pretty sure I've made 0 friends on Reddit. So if you're looking to make some rust friends and get more involved in the community, consider using Twitter and following every rustacean you can find.

6. Direct Encouragement

Now the last thing is something that hasn't fully played itself out, but I expect it will be a major turning point as well. Giving talks! I've never given one other than at work in front of engineers I know personally, and I probably would have continued in that pattern for ages if it wasn't for J Haigh. Manish had been trying to convince me to give a "contribution tour" talk at one of the meetups for a while, and I was interested but I didn't feel ready yet, probably imposter syndrome or some nonsense like that. Then along comes J and they're like, "Hey I see you're active in the Rust OSS community, would you be interested in talking about any of it? COGoldRust currently has a cfp open". Now, I was interested, terrified, but interested. I told them about Manish's idea for a contribution tour and my apprehensions and feelings of being under qualified but they were nothing but encouraging, they helped helped me get over my feelings of imposter syndrome, made the cfp process seem much more approachable, and generally put a friendly face on the COGoldRust conference.

So I said yes, I'm interested, I'll do my best, and I then promptly proceeded to procrastinate for the next two weeks. But that did not stop J! They came back and gently encouraged me to keep working on my submission more and helped me motivate when I probably wouldn't have been able to alone. I expect I would have let the deadline pass and beat myself up over it if they hadn't been so encouraging and helpful. I ended up throwing together a vague outline of the current talk, about how I learned Rust, what I contributed to, and what I learned in the process, which evolved into this current talk about how to effectively participate in the open source community.

I don't really know what will end up happening as a result of being a speaker, but I'll make sure to write a follow up blog in a few months to let yall know! Either way though J's friendliness and encouragement is the main reason I'm here currently. If there are any take aways here it is to be

like J and go out of your way to find qualified speakers and encourage them to take the leap. That could be the thing that helps them finally take the step they've always wanted to take.

Key Takeaways

- CoC gives a sense of safety to underrepresented minorities and repels bad actors
- This Week In Rust is an amazing tool for keeping informed on the goings on within the Rust community
 - TWIR is a great avenue for finding issues to work on or advertising issues you'd like help with
 - Navigating the issue boards by themselves was challenging, particularly hard to figure out which issues are available, which are interesting, and which you can learn the most from, organizing your issue list can reduce friction and attract more contributors
- Meetups and by extension conferences are one of the best ways to make friends in the Rust community and get/feel more connected
- Mentors have a huge amplifying effect on how much work you can do and how approachable the work you want to do seems
 - Working on issues that you care about personally is the most rewarding
 - Don't bite off more than you can chew. Max for me is extra open source 1 issue actively being worked on.
 - Keep track of your active work
 - Always open an issue when you have an idea of something that you want to work on or see done
 - Save a link to the issue somewhere, I'm using a random [github project](#) board for this.
- Use twitter to make friends in the rust community.
- Reach out to people, it makes a huge difference.

Bonus: Shameless Plug

So I want to take this opportunity to share a project that was inspired by the mentorship I've received from Manish, [awesome-rust-mentors](#). If you're familiar with `awesome-rust` or `awesome-vim` or any other such curated awesome list you probably already know what this is. It's a list of rustaceans who are interested in helping new contributors get involved and navigate the community, basically a bunch of Manishes. On it you'll find people's names, contact info, preferred pronouns, spoken languages, and topics of interest. If you're interested in getting a mentor or being a mentor you should definitely go check it out.



I'm also planning on adding a "Call for Contributors" section to [awesome-rust-mentors](#), which is specifically aimed at library specific mentorship for people who are looking to onboard new

people in exchange for mentorship on their project. If you're interested in that please let me know.

Thank You <3

New to Me Things

New to Me Things
dontemailme@yaah.dev

 [yaahc](#)
 [yaahc_](#)

Poorly written blog posts chronicling my attempts to learn new things.