

Weihang Lo's achievements

Data collected: 2023-09-08 to 2023-05-31

Have been working for \$WORK since then, for the first two months I shipped features and fixes to awslabs/smithy-rs. In the other months, I have been working on Cargo, the official open source package manager for the Rust programming language. As a member of the official Cargo team, I am expected to participate activities listed in [Cargo Team: Membership expectations](#). I've been doing beyond the expectations, becoming one of the most significant contributor and reviewer for Cargo. Some numbers:

- Triaged 420+ issue with informative responses and labels
- Reviewed 200+ pull requests for rust-lang/cargo
- Submitted 120+ pull requests to rust-lang/cargo
- Submitted 50+ pull requests to rust-lang/rust (including submodule updates)
- Mentored 11+ people from issue to pull request merged

Improved the contributor experience of Cargo

Achievement: By improving the contributor experience, Cargo is in a healthier maintenance status, getting more contributors. Maintainers now accepting a number of large feature requests, unblocking potential features \$WORK and the community wanting. See [Inside Rust Blog: Welcome Arlo and Scott to the Cargo Team](#).

I have been focused on lowering the contribution barrier by improving contributor documentation and reorganizing modules in Cargo. Notably

- Making Cargo itself a Cargo workspace — This is a stepping stone toward modularization of Cargo project, which can help Cargo plugin developers reuse the internal logic of Cargo. Internal tooling at \$WORK will gain huge benefits from this. It also gives a clearer view of how Cargo codebase is organized, letting contributors become easier to join and hack on it.
 - <https://github.com/rust-lang/cargo/pull/11851>
 - <https://github.com/rust-lang/rust/pull/109133>
- Improved documentations of Cargo internals — This is very vital. Cargo is prominent as an tool, but it is somehow hard to use as a library. I've tried to document more private items to enhance the usability. Notably,
 - Published API doc to the same place with rustc API doc: <https://doc.rust-lang.org/nightly/nightly-rustc/cargo/>
 - Other docs improvement PRs
 - <https://github.com/rust-lang/cargo/pull/12159>
 - <https://github.com/rust-lang/cargo/pull/12153>
 - <https://github.com/rust-lang/cargo/pull/12133>
 - <https://github.com/rust-lang/cargo/pull/11758>
 - <https://github.com/rust-lang/cargo/pull/11711>
 - <https://github.com/rust-lang/cargo/pull/11703>
 - <https://github.com/rust-lang/cargo/pull/11669>
 - <https://github.com/rust-lang/cargo/pull/11241>
 - <https://github.com/rust-lang/cargo/pull/11219>
 - <https://github.com/rust-lang/cargo/pull/11207>
 - <https://github.com/rust-lang/cargo/pull/11185>

- <https://github.com/rust-lang/cargo/pull/11157>
- Improved the issue label system — I started a discussion to improve the current issue tracker. The goal is reduce the burden for both maintainers and contributors to figure out what's going on in an issue. We now have a solid label system and it's pretty clear what is the next step for each issue.
 - The discussion <https://github.com/rust-lang/cargo/issues/11788>
 - Make PR auto-labeled so we can find relevant PR quicker <https://github.com/rust-lang/cargo/pull/11679>

Shepherded large features for Cargo

There are several large impactful features needs maintainers to dive deep and guide the contributors toward success. I've shepherded a couple of them. Namely

- Discussed and reviewed the feature “[lints] table” — This enables Cargo users to specify a [lints] table in `Cargo.toml`, so that they can have a central place to share lint configurations, and take recompile detection into account. Most important, users don't need to hack around `RUSTFLAGS` anymore, and may prevent link rule breakage to some extent.
 - <https://github.com/rust-lang/cargo/pulls?q=is%3Apr+label%3AZ-lints-table+involves%3Aweihanglo>
- Revived the feature “scrape-examples” — This enables Cargo to automatically find example usages from the project itself and display in API doc. It is also a great case demonstrating how to bring people back to the contributor line-up. Not to say this feature will affect everyone relying on docs.rs.
 - <https://github.com/search?q=repo%3Arust-lang%2Fcargo+author%3Awillcrichon+involves%3Aweihanglo+scrape&type=pullrequests>
- Reviewed and discussed “gitoxide integration in Cargo” — This could bring git shallow-clone into Cargo, speeding up fetching git dependencies and registry index, and code checkout.
 - <https://github.com/search?q=repo%3Arust-lang%2Fcargo+author%3Abyron+involves%3Aweihanglo+gitoxide&type=pullrequests>
- Analyzed issues around “artifact dependencies” — The feature itself was merged, but the original author has been working on other features. This accomplishment shows how to take over a feature for the author and continue collaborating with others for it to get stabilized. I've either fixed or mentored them towards resolved. This is also a feature that some teams at \$WORK want for their peculiar build workflow.
 - <https://github.com/rust-lang/cargo/issues?q=is%3Aissue+label%3AZ-bindeps+involves%3Aweihanglo>

Other highlights

- Created a gratitude thread in the Reddit Rust community to encourage people show their appreciations to open source contributors — <https://www.reddit.com/r/rust/comments/13ug42p>. I believe that helps the community in general and gave everyone faith in Rust.
- Cargo postmortem analysis has been published in the official “Inside Rust” blog. This COE was written at work and we shared the great COE template in public on behalf of Cargo team.
 - <https://blog.rust-lang.org/inside-rust/2023/05/01/cargo-postmortem.html>.
- Interviewed by Rust Magazine as an open source developer — <https://rustmagazine.org/issue-1/weihanglo/>
- Made contribution to 3 CVE and was mentioned in official CVE advisory and blogpost twice for the contributions of fixes. Also helping some public statements to clarify the situation and how to mitigate.
 - <https://blog.rust-lang.org/2023/01/10/cve-2022-46176.html>
 - <https://blog.rust-lang.org/2022/09/14/cargo-cves.html>
 - In addition, I lifted an limitation brought by an CVE fix to help developers at work unblocked.
 - <https://github.com/rust-lang/cargo/pull/11337>

- Fixed Cargo configuration loading order issue, which introduced by a colleague at work, and praised by the same colleague in their live coding stream. <https://github.com/rust-lang/cargo/pull/11077>
- Mentored 10+ issues towards pull requests merged.
 - <https://github.com/rust-lang/cargo/issues/10837#issuecomment-1339365374>
 - <https://github.com/rust-lang/cargo/issues/11248#issuecomment-1280494254>
 - <https://github.com/rust-lang/cargo/issues/3736#issuecomment-1288472646>
 - <https://github.com/rust-lang/cargo/issues/11315#issuecomment-1306071330>
 - <https://github.com/rust-lang/cargo/issues/11251>
 - <https://github.com/rust-lang/cargo/issues/11311#issuecomment-1295718944>
 - <https://github.com/rust-lang/cargo/issues/11708>
 - <https://github.com/rust-lang/cargo/issues/10834#issuecomment-1401006740>
 - <https://github.com/rust-lang/cargo/issues/11513#issuecomment-1385289734>
 - <https://github.com/rust-lang/cargo/issues/11574#issuecomment-1381987309>
 - <https://github.com/rust-lang/cargo/issues/11555#issuecomment-1379359837>
- Have been in charge of Changelog since Rust 1.65 to 1.69 to today's nightly version.
- Involved in several large feature/RFC discussion, including but not limited to
 - Cargo target features <https://github.com/rust-lang/rfcs/pull/3374>
 - `cargo script` <https://github.com/rust-lang/rfcs/pull/3424>
 - General mechanism for RUSTFLAGS <https://github.com/rust-lang/rfcs/pull/3310>
 - SemVer v2 <https://github.com/rust-lang/rfcs/pull/3266>
 - Backward compatible default features <https://github.com/rust-lang/rfcs/pull/3283>
 - Running binaries from dev/build deps <https://github.com/rust-lang/rfcs/pull/3168>
 - Path sanitization <https://github.com/rust-lang/rfcs/pull/3127>

Appendix

- [Making Cargo a Sustainable Dependency](#) — This was the reason I have been working on Cargo
- [Weihang's Roadmap 2023](#) — This highlights what I can help for Rust builders at \$WORK
- [Number of contributions to rust-lang/cargo from 2022-09-08 to 2023-05-31](#)