Can we improve the experience of first-time LLVM contributors?



Alex Bradbury asb@lowrisc.org @asbradbury @lowRISC

Questions for the audience

Who has used LLVM's Phabricator before?

Questions for the audience

Who has used ever submitted a patch for LLVM?

Questions for the audience

Who has submitted an LLVM patch and found it languishes with no reviewers?

Current contribution process

- Write patch
- Submit to Phabricator
 - Try to identify a CODE_OWNER to review
 - Tag people you might know to help review
 - Look at git blame, and pick on the unlucky soul who last touched the relevant file

Potential problems

- Code owners are often busy
- Newcomers haven't yet gained "review currency" in the LLVM community
- Finding your hard work seemingly ignored can be offputting

Even if feedback is negative, it's valuable to know someone has looked at your code.

What do others do? Case study - Rust

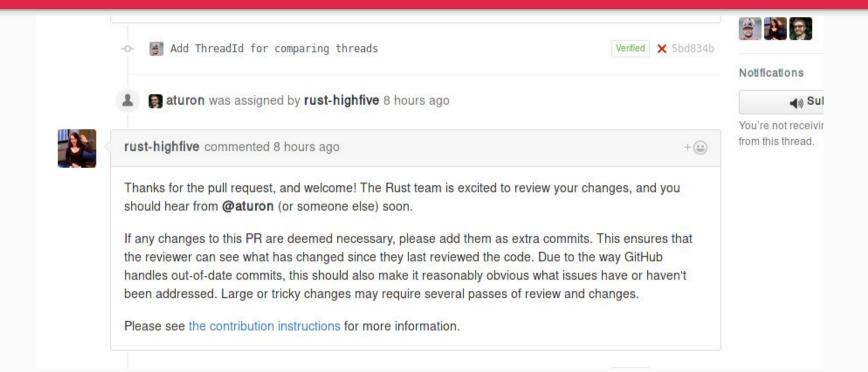
This Week in Rust

06 SEP 2016

New Contributors

- Abhishek Kumar
- Andrea Corradi
- athulappadan
- Eugene R Gonzalez
- Fabian Zaiser
- johnthagen
- Keunhong Lee
- king6cong
- Matt lckstadt
- philipp
- QuietMisdreavus
- Sebastian Ullrich

What do others do? Case study - Rust



Conclusion

- Seems like a good idea let's steal it!
- Need to provide
 - Phabricator bot
 - Community of volunteers to be tagged
- Potential pitfall: no use telling submitters to clean up code style if the fundamental approach will never be accepted by code owner
- I haven't surveyed potential LLVM contributors maybe there isn't a problem that needs to be solved?
- Keen to hear your views let's discuss at the Social