

Jan 9, 2024 | 📅 SEA Open Discussion on Code Review

Attendees:

- Brian Vanderwende, NCAR/CISL/HPCD, HPC consulting
- Orhan Eroglu, NCAR/CISL/TDD/VAST, Software Engineer
- Negin Sobhani, NCAR/CISL/CSG, HPC Consultant
- Brian Dobbins, NCAR/CGD/CSEG, SE
- Daniel Howard, NCAR/CISL/CSG, HPC Consulting
- Anissa Zacharias, NCAR/CISL/TDD/VAST
- Jenett Tillotson, NCAR/CISL/HPCD/HSG, Systems Engineer
- Janine Aquino, NCAR/EOL/RAF
- Michael Galloy, NCAR/HAO/MLSO
- Katelyn FitzGerald, NCAR/CISL/TDD/VAST, Software Engineer
- Chris Flischer, NCAR/CGD/CSEG
- Will Shanks, NCAR/CISL/HPCD/HSG
- Michael Waxmonsky, NCAR/CGD/CISL, Software Engineer
- Ryan May, UCP/Unidata
- Ana Victoria Espinoza, UCP/NSF Unidata, Software Engineer
- Helen Kershaw, NCAR/CISL
- Jesse Nusbaumer, NCAR/CGD/AMP
- Bob Dattore, NCAR/CISL/ISD
- Bill Sacks, NCAR/CGD/CSEG, Software Engineer
- Cheryl Craig, NCAR/CGD/CSEG/AMP
- Jim Edwards, NCAR/CGD/CSEG
- Julie Prestopnik, NCAR/RAL/JNT
- Sam Rabin, NCAR/CGD/CSEG, software engineer
- George McCabe, NCAR/RAL/JNT
- Paul Prestopnik, NCAR/RAL
- Wayne Chuang, Columbia/CSEG, Integration Engineer
- Gary Granger, EOL, Software Engineer
- John Halley Gotway, NCAR/RAL/JNT
- Erik Kluzek, NCAR/CGD/CSEG/TSS, Software Engineer

Notes (anyone can contribute here!)

- Share pos./neg/ experience with code reviewing
 - Erik: Did not use to do it years ago but like and do it now
 - At CTSM & CESM work, common now
 - Code review feels like someone works with me and we learn from each other
 - Downside: Back/forth that happens, especially since the code review is not #1 priority. Can slow down the process.
 - Would be happy to hear experiences with this
 - Paul: Second Erik on most.

- GitHub made it much easier to code review.
 - When people do not know much about what others do in the code, review gives an opportunity to learn about what is going on in the project
- Bill: Getting very burned out with code reviews generally
 - E.g. Do a review, wait ~2 weeks, can feel really negative sometimes
 - Recently got more negative on it but would love to hear positive experiences about it
- Cheryl: Used to do code reviews in person years ago. Finding bugs and avoiding problems down the line works great. Can't imagine deploying code without reviews. Couldn't maintain the code without reviews.
- Sam: Experience mostly getting my code being reviewed rather than reviewing others'. Need to coordinate with each other to find the time. Trick is that it'd be helpful to walk the reviewer through the code first.
- Julie: The objectives can be communicated well beforehand using a pull request template to reduce the overhead of back & forth and expectations for a due date for the pull request can be set.
 - Here are two examples of pull request templates from two of our METplus repositories to our pull request templates:
 - MET: https://github.com/dtcenter/MET/blob/develop/.github/pull_request_template.md
 - METplus: https://github.com/dtcenter/METplus/blob/develop/.github/pull_request_template.md
- Kevin: Communicating what to look at in the code is really important.
- Negin: A lot of friction points about code review.
 - Ethics around code review is not clear.
 - Code review is a lot of times not equitable, e.g. more pushback for women's code.
 - Here is their summary: <https://developers.googleblog.com/2022/06/Using-research-to-make-code-review-more-equitable.html>
 - Here is their article: <https://cacm.acm.org/magazines/2022/3/258909-the-pushback-effects-of-race-ethnicity-gender-and-age-in-code-review/fulltext>
 - Related article posted by Katelyn: [Systemic Gender Inequities in Who Reviews Code](#)
- Daniel: CISL DEI Committee created this [summary document](#) about a DEI workshop *Vive La Difference* for RSEs. Notable to Negin's point is [this presentation](#) at the workshop by Dr. Kelly Blincoe about code review as a socio-technical activity. Includes relevant data and potential policy implications on code review processes and impact.
- Orhan: Code style actions, automation could be helpful with the code reviewing process to reduce unwanted reviewing (code styling, etc.)

- John: Second automation and prep for code review to focus on the actual objectives. Also second Paul's points about knowing what the others' code does in the code base could help with the process.
- Experiences with getting reviewed / giving review:
 - Gary: Pick the most impactful aspects of the code to comment on, no need to mention everything. Impact can include functionality, quality, maintainability, readability, testability.
 - Brian: Submitting changes without sufficient descriptions is less helpful.
 - Michael: Sometimes reviews have a lot of back & forth, and can get political. Try to keep it very non-personal. The thing being reviewed is not the person but the code that will benefit an entire project/organization.
 - Bill: It's a joint responsibility.
 - Sam: Wording/language can really help:
 - E.g. "Why did you do this?" instead of "You should have done this"
 - Paul: use an automated tool to separate those from your code review
 - Gary: Encourage "the code" and not "your code". We are not our code.
 - Someone may learn more from a suggestion to test the code themselves and verify that it works. Good code needs good tests.
 - Jesse: Make it clear about the asynchronous aspect of the PRs. Also use "why would you do that?" for asking the reasoning (?)
- Any recommendations/thoughts about where the code is coming from (e.g. soft.dev., scientist, external collaborator, etc.)
 - Katelyn: Having been in both scientist and developer perspectives, set expectations and convey what the goals are for each group, collective set of expectations. And, things may differ from person to person, even if they are all one kind (e.g. scientist).
 - Orhan: Set the expectations and process clearly.
 - Julie: Consistency. Type of code you are working on (pure research vs. operational product/deliverable) and how you set expectations is also very important.
 - Cheryl: One of the first things we do in our GH repos is to encourage
- Experience with external reviewers:
 - Orhan: Unfortunately missed the scribe here
- Anything else:
 - Brian: 1:1 code review in person is a bit different than remote.
 - Helen: Onboarding is very useful to get people up to speed.
 - Bill: Do onboarding by working side-by-side rather than a remote pull request review process. Some form of pair programming.
 - Cheryl: When getting someone new to our code contributions, reach out individually with an email that clarifies some important points about the process.
 - Jesse: Any experiences/thoughts with how many approving reviews for a PR to be merged?

- Julie: Largely project dependent, but typically the METplus team has 1 or 2 reviewers (often an engineer, but sometimes two engineers or one engineer and one scientist, etc. depending on the need).
- Paul:
 - [How Microsoft do code reviews](#) mentions the use of emojis to describe things like nitpick, thinking out loud, take it or leave it, etc.

Suggestions / feedback

- Meeting time:
 - This time (Tuesday 2 pm) conflicts with the weekly CESM Software Engineering Working Group meeting
-
-