



From First PR to Release Manager: A Path into Open Source

Daniel Scherzer



Daniel Scherzer

- Full-stack developer and open source contributor
- <https://scherzer.dev/>
- <https://github.com/DanielEScherzer>
- PHP core contributor
 - ext/reflection maintainer
 - PHP 8.5 rookie release manager; PHP 8.6 veteran release manager



***My* Path**





Terms

- CI: the process of continuously merging in new changes (“**continuous integration**”)
 - In the context of GitHub, generally includes the tests that are run on PRs
- PR: a “**pull request**” on GitHub - a proposal to merge code changes
- RM: a “**release manager**” for one or more software releases

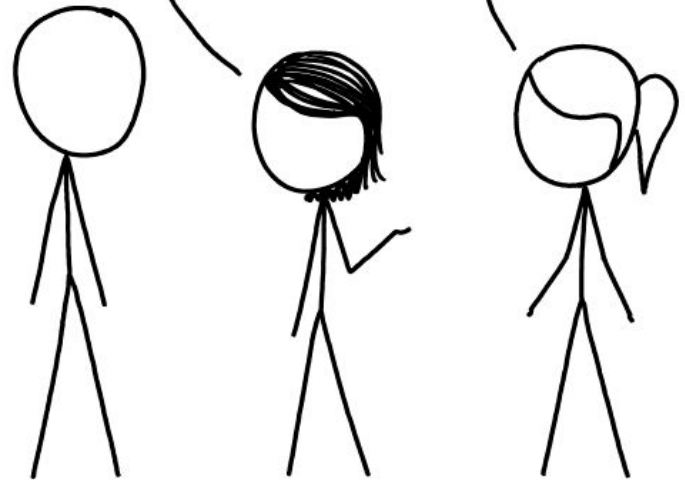
Anyone can contribute

[XKCD #1954](#)

THIS IS DR. ADAMS. SHE'S A SOCIAL PSYCHOLOGIST AND THE WORLD'S TOP EXPERT ON IMPOSTOR SYNDROME.

HAHA, DON'T BE SILLY! THERE ARE LOTS OF SCHOLARS WHO HAVE MADE MORE SIGNIFICANT...

...OH MY GOD.





Timeline

- 2024-08-27: DanielEScherzer GitHub account created
- 2024-08-27: First PR [php/php-src#15609](#) created
- 2024-09-22: First suggestion that I request commit access
- 2025-03-13: Call for volunteers for PHP 8.5 RMs
- 2025-03-21: I requested commit access
- 2025-03-24: Request granted, I become ext/reflection maintainer
- 2025-03-26: I volunteer as a PHP 8.5 RM
- 2025-04-17: PHP 8.5 RM election closes, I am a release manager



My First PR: Typo Fix

```
diff --git a/Zend/tests/assert/expect_015.phpt b/Zend/tests/assert/expect_015.phpt
index 5a4bbebbe914..695c4c166a83 100644
--- a/Zend/tests/assert/expect_015.phpt
+++ b/Zend/tests/assert/expect_015.phpt
@@ -1,5 +1,5 @@
--TEST--
-AST pretty-peinter
+AST pretty-printer
--INI--
zend.assertions=1
--FILE--
```



My First Roadblock: CI



5 workflows awaiting approval

This workflow requires approval from a maintainer. [Learn more about approving workflows.](#)

- | | | |
|---|---|----------|
| ● |  Audit Expected — Waiting for status to be reported | Required |
| ● |  CI Expected — Waiting for status to be reported | Required |
| ● |  Spell Check with Typos Expected — Waiting for status to be reported | Required |



CI Approval - Configuration

Approval for running fork pull request workflows from contributors

Choose which subset of users will require approval before running workflows on their pull requests. Both the pull request author and the actor of the pull request event triggering the workflow will be checked to determine if approval is required. If approval is required, a user with write access to the repository must [approve the pull request workflow to be run](#).

☐ **Require approval for first-time contributors who are new to GitHub**

Only users who are both new on GitHub and who have never had a commit or pull request merged into this repository will require approval to run workflows.

☒ **Require approval for first-time contributors**

Only users who have never had a commit or pull request merged into this repository will require approval to run workflows.

☐ **Require approval for all external contributors**

All users that are not a member or owner of this repository will require approval to run workflows.

Save



CI Approval - Maintainer View



1 workflow awaiting approval

This workflow requires approval from a maintainer. [Learn more about approving workflows.](#)

Approve workflows to run



CI and Feedback Cycles

- Developer availability vs reviewer availability
 - [uriparser/uriparser#328](#) - created 0831, merged 0858 after multiple rounds of back and forth
- Wasting time on broken patches
- Risks of CI on untrusted patches
 - Cost on self-hosted runners
 - Security risks?



Types of Contributions

- Bug fixes: behavior changes - adjust existing functionality
- Features: behavior changes - add new functionality
- Deprecations: ecosystem changes - reduce encouraged functionality
- Removals: behavior changes - reduce available functionality
- Cleanup: internal changes - not visible
 - Refactors, documentation, test improvements, etc.



PHP RFC: Attributes on Constants

- Motivation: deprecating global constants
- Required changes to:
 - Language grammar
 - Compiled opcodes
 - Reflection extension
 - Opcache extension



Finding Reviewers: CODEOWNERS



[author] requested a review from **DanielEScherzer** as a code owner 5 days ago



Finding Reviewers: Rust triagebot



rustbot commented 3 weeks ago Collaborator ...

r? @[reviewer]

rustbot has assigned @[reviewer].
They will have a look at your PR within the next two weeks and either review your PR or reassign to another reviewer.

Use r? to explicitly pick a reviewer

▼ Why was this reviewer chosen?
The reviewer was selected based on:

- Owners of files modified in this PR: libs
- libs expanded to 12 candidates
- Random selection from 6 candidates





Finding Reviewers: Do They Exist?

- No clear maintainer
- No clear timeline for review
- [“Open Source is Not About You”](#)



Unfamiliar reviewers

- PHP ext/zip had no maintainer from 2020 until last month
- Feature addition: [ZipArchive::closeString\(\)](#)
- I volunteered to review
- Documentation for reviewers
- Minimal atomic changes
 - Split out [php/php-src#21659](#)
 - Tests and assertions



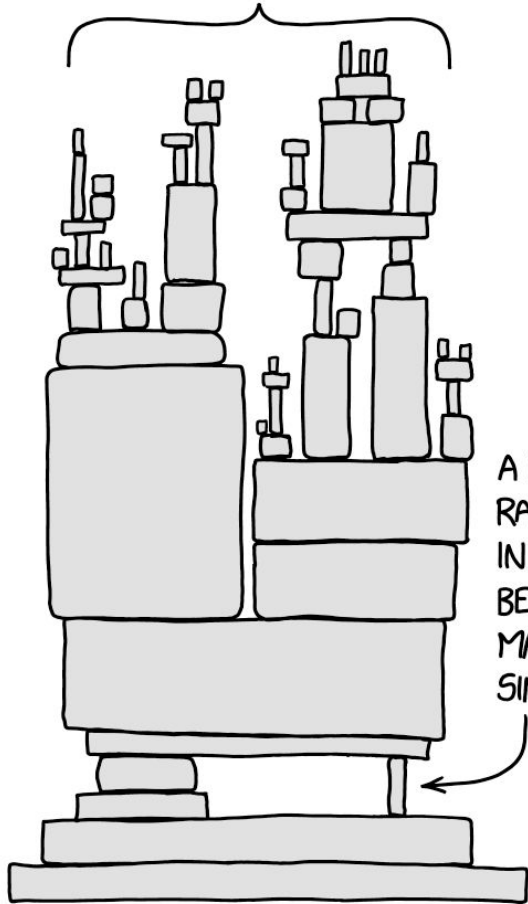
Takeaways (contributors)

- Start small
- Prioritize bug fixes
- Help reviewers
- Be patient

Takeaways (maintainers)

- Provide reviewers
- Document expectations
- “Good first issue” label
- Specify criteria for commit access

ALL MODERN DIGITAL
INFRASTRUCTURE



A PROJECT SOME
RANDOM PERSON
IN NEBRASKA HAS
BEEN THANKLESSLY
MAINTAINING
SINCE 2003

[XKCD #2347](#)



Thank You

Questions?