

Incentivizing Code Reviews and PR Testing

This page is dedicated to gathering resources / brainstorming for how to incentivize / encourage community developers to help with Code Reviewing and Pull Request Testing

General Goals / Ideas

1. We need to find a way to encourage more reviewers from our large community of developers. Lots of people doing a small number of tests / reviewers scales very well.
2. Document the incentives for people to do reviews / functional testing.
3. Find a way to make the codebase easier to work with. We've done some of this with Docker. But we should investigate ways to spin up DSpace in a temporary, virtual environment with minimal configuration/steps. This would allow anyone to more easily interact with & test individual PRs
4. Find a way to acknowledge code reviews / functional testing in Release Notes in the same way as development/code is acknowledged.

Resources to make Developers feel welcome

General Goal: Find a way to encourage other developers to get involved & help out in small ways

- [New Developers Hub](#) - Draft docs for new developers started by [Hardy Pottinger](#)
- [Pull Request "Trading"](#) (discussed/approved in [2023-08-24 DSpace Developers Meeting](#))
 - Developers are encouraged to ask other developer(s) to review/test their PR *in exchange for reviewing/testing a PR created by the other developer*. This allows both developers to get more immediate feedback!
 - [Tim Donohue](#) will also take part in this PR "trading" but in a more general fashion.
 1. If you review or test **any two** similar-sized PRs (of your choice), I (Tim) will promise to review **one** of your PRs as soon as possible (Your PR go to the top of my "to do" list.)
 - a. PRs you review or test can be *any PR* on one of our boards ([7.6.1 Board](#) or [8.0 Board](#)), provided they are similar-size to your own PR.
 - b. PRs you review or test *must be* from a developer at a *different institution* from your own.
 - c. You must submit *useful* feedback on the PR you've reviewed/tested (via a comment on the PR or similar). It can be positive or negative feedback (if you test it and it doesn't work for you, that still counts).
 2. If I (Tim) don't notice your two reviews, please message me (privately is fine) via Slack or email & let me know which of your PRs you want me to review as soon as possible.
 3. You can also "trade" to support someone else's PR. Review any two PRs and then ask me (Tim) to review someone else's similar-sized PR.

Resources for making Code Reviews / Testing easier

General Goal: Find a way to make the codebase easier to work with & test PRs with.

- [Testing DSpace 7 Pull Requests](#) - How to use existing Docker scripts to spin up PRs more easily locally, in order to test them or review them.
- Spin up code in virtual environment (quickly) for easier reviews/testing
 - GitHub CodeSpaces: <https://docs.github.com/en/codespaces/overview>
 - [Tim Donohue](#) has played with this, and it's possible to spin up DSpace 7 in CodeSpaces. However, you need to use a 4 Core / 8GB RAM machine type (otherwise, the UI will often fail to build with an "error 137" - meaning it ran out of memory during the build process). Connecting the UI to a running Backend is also not super-easy right now, as codespaces assigns a random URL. We'd need to likely provide a Codespaces configuration for DSpace to make this easier.
 - May be a way to use our Docker scripts **in** CodeSpaces, but it's not easy yet. See <https://notes.alexkehayias.com/running-docker-compose-in-codespaces/> and <https://github.com/orgs/community/discussions/34090>
 - GitPod: <https://www.gitpod.io/for/opensource>
- Automated Code Reviewing resources. Tools/resources exist which can do some automatic checking/verification of code quality in Pull Requests. Some examples include:
 - [Code Scanning in GitHub](#). We already do some of this, but currently we only scan for security-oriented code issues.
 - Free & integrated into GitHub. Interface is a bit clunky at times though.
 - Highly Configurable (e.g. see [query types](#))
 - Could configure this to also check PR code quality against coding best practices (currently we only scan for major bugs / security issues). See [these settings](#)
 - [SonarCloud.io](#) - This is a hosted version of [SonarQube](#).
 - Free for open source projects. Integrates with GitHub & supports both Java and TypeScript. Can run on every new PR.
 - Example projects: <https://sonarcloud.io/explore/projects>
 - Test analysis run by Tim on DSpace backend/frontend: <https://sonarcloud.io/organizations/ttonohue/projects> (Keep in mind, there are definitely false positives listed here. These are just raw reports)
 - Pros: Highly Configurable Used by other major OS projects like Apache. Good documentation/resources on *how to fix* any issues that are found. SonarQube is open source itself.
 - [DeepSource](#)
 - Free for open source projects. Integrates with GitHub & supports both Java and TypeScript. Can run on every new PR.
 - Test analysis run by Tim on DSpace backend/frontend (Keep in mind, there are definitely false positives listed in both. These are just raw reports)
 - Backend: <https://app.deepsource.com/gh/ttonohue/DSpace/>
 - Frontend: <https://app.deepsource.com/gh/ttonohue/dspace-angular/>
 - Not as configurable, but able to turn off individual rules if they are too "noisy" or not useful.
 - Pros: Some "autofix" options. Good documentation/resources on *how to fix* any issues that are found.

Resources for acknowledging code reviewers / testers

General Goal: Find a way to acknowledge / track code reviewers so that we can more easily include them in Release Notes (and include this as a form of contribution for service providers). Ideal is that it is either automated or semi-automated (e.g. a report that can be run regularly per release)

- All Contributors: <https://github.com/all-contributors/all-contributors>
 - Great resource for acknowledging all types of contribution. However, it'd be nice if we could find a way to do this **per release** (i.e. in Release Notes) rather than just in general README.
- "Top Contributors" <https://github.com/tdonohue/top-contributors>
 - Old (unmaintained) project from [Tim Donohue](#) to try to highlight/acknowledge top reviewers/code contributors **per month** using data from GitHub's API.
 - Old demo at <https://tdonohue.github.io/top-contributors/>
 - Maybe look at whether it's possible to adapt this to give similar stats per release?
- Gamification dashboard / leaderboard like this: <https://github.com/PicnicSupermarket/pr-leaderboard>
 - Code is outdated (no updates in 4 years) but the concept is interesting