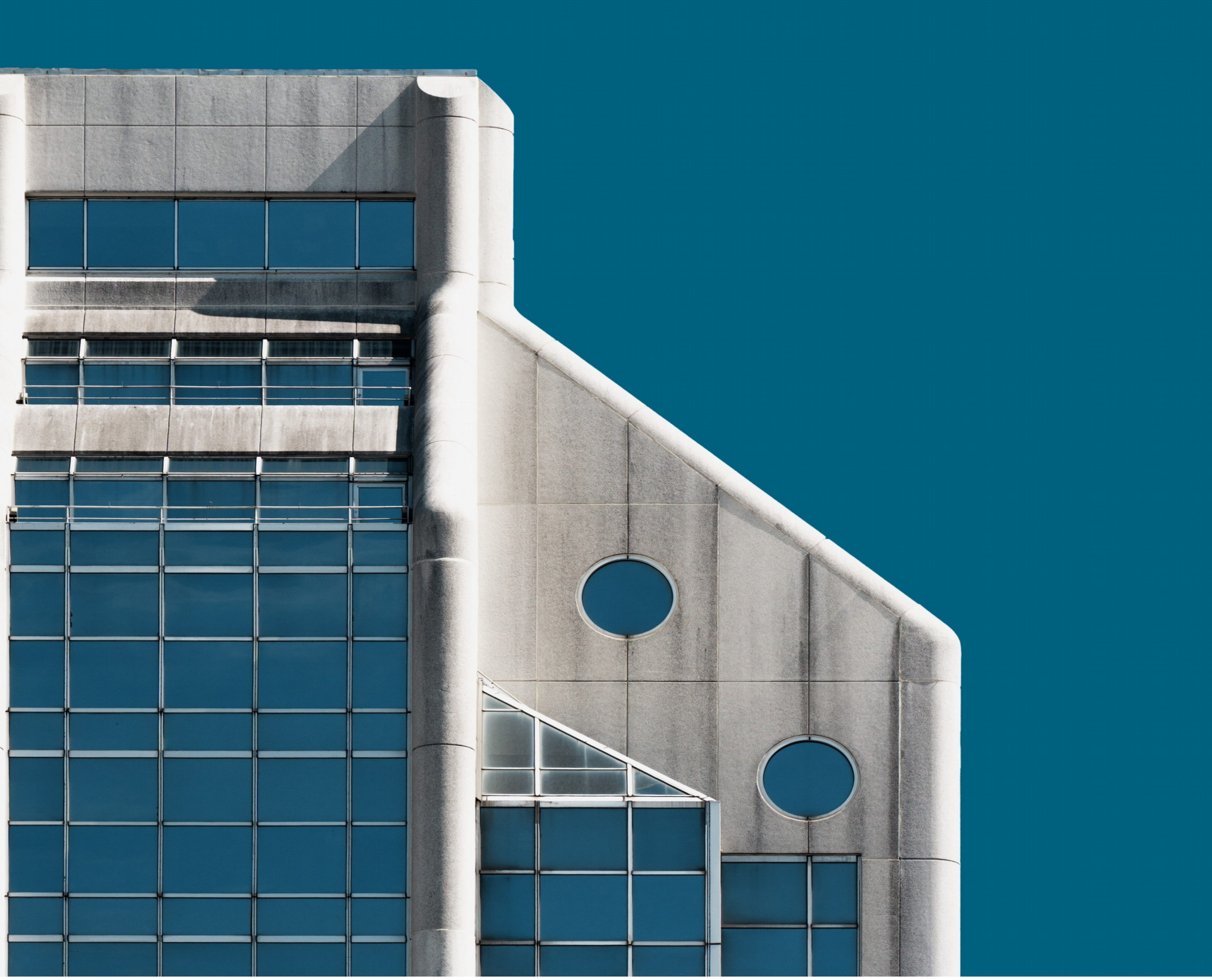

NGI Zero PET and Discovery Legal To-Dos

Guidelines for External Contribution Policy

Published by the Free Software Foundation Europe (FSFE)
Berlin, September 2019
EU Transparency register ID: 33882407107-76
www.fsfe.org

Responsible according to European press law:
Matthias Kirschner / FSFE e.V.
Schönhauser Allee 6/7
10119 Berlin
Germany

Contributions by: Gabriel Ku Wei Bin and Lucas Lasota.

Photo credit:
Pierre Châtel-Innocenti on Unsplash.

Version: 2.0 – 05.2020

Table of Contents

GUIDELINES FOR AN EXTERNAL CONTRIBUTION POLICY

Why do I need a external contribution policy for my project?

An external contribution policy is a set of rules that govern the management of third party FOSS within a project. More precisely, it states the terms on which external contributions are handled and how they are audited and distributed.

Therefore, if you want to build a community around the project, and accept contributions from developers outside your team, a simple and clear external contribution policy can help your project to maximize the impact and benefit of using FOSS, at the same time ensuring the mitigation of technical and legal risks.

External contribution policy and legal documentation

Since an external contribution policy is not a strictly legal document, it makes sense to have a legal instrument that you can point contributors to in order to protect your project from code contributions that may infringe third party copyright or patents. Depending on the size and compexity of the contributors network, there are different legal instruments you can use to manage external contributions.

In order to determine which of those you should use for your project you need firstly to decide how your project will handle third party copyright. Will it require the transfer of the copyright of all contributions? Will contributors be able to retain their rights? You should consider the needs of your specific software project and the community that it serves, and decide accordingly.

Below are some pointers about each one of them. For more information about each one of these mechanisms, you can refer to our **FAQs and Background Information. I**f you have questions, please also do not hesitate to contact us.

Developer Certificate of Origin (DCO)

DCOs set the incoming license of the external contribution to be the same as the outgoing license of your software project. DCOs reduce the barriers for external developers to contribute, while still requiring them to certify that they are submitting their own work. The simplicity of a DCO makes it perfect for smaller projects that want to have a community of contribution around the project.

DCOs are in essence an affirmation that the contributor intends to contribute the contributed code under the project's license, and that the contributor has the right to do so. In using a DCO, external developers certify that they adhere to these requirements by signing off on their commits.

Contributor License Agreement (CLA)

On the other hand, using a CLA reserves the right for you to decide whether to relicense the entire software project (and accordingly all past contributions to it) under a different license in the future. You can find templates for contributor-friendly, multi-purpose contributor agreements on the following website: https://contributoragreements.org. Please note that CLAs can be quite complex. We recommend you to contact us if you are planning to have a CLA implemented in your project.

Copyright Assignment Agreements (CAAs)

CAAs are a more heavy-handed manner of ensuring that the software project always has the rights to use the code that has been contributed. CAAs are complex legal documents, and can create complex legal questions in many jurisdictions. Accordingly, we do not recommend that you adopt them for your software project.

Fiduciary License Agreements (FLAs)

FLAs are similar to CLA but do not involve copyright assigment, allowing the contributors to retain their rights. Instead, the holder of rights (trustee) grants a broad license back to the contributor. You can find templates for a FLA on the following website: https://contributoragreements.org. Please note that FLAs can be quite complex. We recommend you to contact us if you are planning to have a FLA implemented in your project.

Implementing an External Contribution Policy

A simple, clear and lightweight policy can be easily implemented by creating a file where you indicate the rules of contribution and another one with the legal mechanism to enforce this rules. Given the the DCO is a simple and an easy-to-implement measure for smaller projects, we recommend the DCO throughout these guidelines. You can implement an External Contribution Policy on your project by following these steps:

1. Create a `CONTRIBUTING.md` file in the root of your project's repository

The `CONTRIBUTING.md` file should make it easy for contributors start working with it. It must state very clearly what the steps a contributor needs to follow are, in order that their contribution may be accepted. Keep the guide as concise as possible.

Minimally, the `CONTRIBUTING.md` file should include:

- A brief description of the project and a link to its documentation;
- How potential contributors can contact the project team;

- Information about the canonical source code repositories;
- Describe the legal aspects of the contribution and how contributors can sign the DCO;
- Procedures of how to actually contribute (pull requests accepted, etc.);
- Very specific instructions on how contributions are accepted;
- Any project specific rules as code formatting and code quality;
- The process for code review.

2. Add the DCO text to a file in the root of your project’s repository

```
Developer Certificate of Origin
Version 1.1

Copyright (C) 2004, 2006 The Linux Foundation and its contributors.

1 Letterman Drive
Suite D4700
San Francisco, CA, 94129

Everyone is permitted to copy and distribute verbatim copies of this
license document, but changing it is not allowed.

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I have the
right to submit it under the open source license indicated in the file; or

(b) The contribution is based upon previous work that, to the best of my
knowledge, is covered under an appropriate open source license and I have
the right under that license to submit that work with modifications,
whether created in whole or in part    by me, under the same open source
license (unless I am permitted to submit under a different license), as
indicated in the file; or

(c) The contribution was provided directly to me by some other person who
certified (a), (b) or (c) and I have not modified it.

(d) I understand and agree that this project and the contribution are
public and that a record of the contribution (including all personal
information I submit with it, including my sign-off) is maintained
indefinitely and may be redistributed consistent with this project or the
open source license(s) involved.
```

3. Ask contributors to sign the DCO

You can choose the way you ask contributors to sign a DCO electronically. If you are using Git, contributors can *sign-off* that they adhere to these requirements by adding a `Signed-off-by` line to commit messages. To indicate that the contributor agrees to the terms of the DCO, they “sign off” the contribution by adding a line with their name and e-mail address to every Git commit message. If the contributor sets their `user.name` and `user.email` as part of the Git configuration, they can sign the commit automatically with `git commit -s`:

```
$ git commit -s -m 'This is my commit message'
This is my commit message
```

```
Signed-off-by: Random Developer
<random@developer.example.org>
```