Re: What is a Contributor?

Brett Gilio

<2020-08-16 Sun 13:16>

Contents

1 Before I can contribute. . . 1

2 Focus on development skills, and making your patches. 2

3 Teaching others. 3

1 Before I can contribute. . .

Drew DeVault outlines many great points regarding the intrinsic value of contributing to a project. A contributor is usually one that is able to identify problems on their own, or by knowing where to look. From there they can apply their knowledge or skill-set to the problem of their choice. This is the way in which virtually all free software projects harness their community base. However, what is neglected in Drew's article (perhaps on purpose, as it is outside of the scope of this FAQ-style entry) is how to identify when you are ready to tackle a problem in a particular code-base.

Often times when I am conversing with people over IRC, email, or a particular forge for source hosting, there are people who have desires in making a contribution to a favored project. Usually their aspirations are quite large and unrealistic for an untrained programmer, often wanting to contribute the next driver to Linux, or bring the Hurd to completion. What they neglect to realize is the capacity of such a contribution. Since they do not see the kind of rigor, etiquette, and style expectations for making contributions to such projects, the very moment they are faced with starting to make their contribution they have already put themself in a lose-lose situation.

A person who wishes to make a contribution to any project, not just large code-bases, should begin by trying to understand what their particular

strengths and weaknesses are when it comes to programming. For example, if you have exemplary style and skill with the Python programming language, and you wish to contribute to a non-Python code-base you must first ask yourself if your contribution is going to be beneficial to the overarching goal of the project. What you know of Python may not translate well to a project in C or another language. In such cases, are you prepared to learn another language to make a contribution, or should you focus your insights on a request for somebody else to perhaps tackle?

Recognizing one's limitations is paramount to the health and vitality of free software projects. We often accept contributors to projects under the guise that they will usually be one-off changes. However, if you approach a software community and establish yourself as a staple of that community you must be ready to make clear your limitations. It is important that you do not place yourself in a situation where you will feel harassed for not contributing for a week, or making a change that not everybody will agree with.

2 Focus on development skills, and making your patches.

When you feel it is appropriate to begin making contributions to a project it is best, as Drew also says, to get in the habit of following up all requests with your own patch implementations of whatever change you feel is necessary. Submitting a patch not only shows great intent for keeping the vitality of a project moving, it additionally offers you a pass into mentorship. When you submit a patch, you are not only making yourself necessarily vulnerable to critique, additionally you will find that there are different vantages with which you can work.

A patch, however, must come with respect and good etiquette. Acceptance of all criticism must take priority before marring over disagreement. We are all bastions of free software, and wish to see it progress. Naturally we will find disagreements between each other and our proposed implementations. Keeping a level mentality, and striving to understand each other as mutuals is key to approaching free software as a person of clarity and not imposition.

One thing you must additionally find yourself becoming accustomed to is working within the style guidelines that the project dictates. This includes, but is not limited to: Git commit messages, indentation style, bracket placement, type and naming schemas, comment style, and testing your changes. Failure to do these steps is often accepted on first contributions with main-

tainers and committers taking care to help you along. However, repeated failures to make accord with the guidelines of contributing to a project will often find your patches neglected because it simply becomes a burden for the project maintainers.

3 Teaching others.

If you are a regular contributor to a project, and likely already understand the expectations of working within an existing project it is equally important that you find a way to be reasonably pedagogical. Simple fact is, teaching is not for everybody. Often, no mentorship is preferred to bad mentorship. It is highly damanging to push somebody away from contributing to a project (and free software in general) because of a bad experience somebody had. Naturally, not all bad experiences can be avoided, but awareness of what you have to offer somebody who is new is key.

If you are not somebody gifted in socialization and teaching others what you know, it is great to have sources to offer the person in question. That may be contacts to other people who are more open to mentorship, links to books/documentation/resources in question, and general encouragement while making it clear (in a polite fashion) that you are not currently available to take somebody as a mentee.

Great teaching is pivotal to the vitality of the community you are working in. If you are able to, work with people of all skill-levels, from the person who doesn't know what imperative programming even means yet feels they want to write the next low-level graphics API, to the person who is making regular contributions and may need assistance in viewing a particular problem from a different perspective.

We all have an essence to contribute to a project that goes beyond simply making a change to a code-base. Approach any project and project maintaines with dignity, and offer them only what you are comfortable offering. Free software has a future when we treat each other with respect.

Have a response?

Responses and discussion pertaining to any of the blog entries on my website are welcome! Start a discussion on the mailing list by sending an email to ~brettgilio/blog-discussion@lists.sr.ht.

Errata:

- *<2020-08-17 Mon 12:47>* Make changes to spelling, and general grammar.
- *<2020-08-23 Sun 21:30>* Replace `org-blogfeed` with `org-webring`.