Port Mentor Guidelines

Table of Contents

1. Guideline for Mentor/Mentee Relationships 1

1. Guideline for Mentor/Mentee Relationships

This section is intended to help demystify the mentoring process, as well as a way to openly promote a constructive discussion to adapt and grow the guidelines. In our lives we have too many rules; we are not a government organization that inflicts regulation, but rather a collective of like minded individuals working toward a common goal, maintaining the quality assurance of the product we call the Ports Tree.

1.1. Why Mentor?

- For most of us, we were mentored into the Project, so return the favor by offering to mentor somebody else in.
- You have an irresistible urge to inflict knowledge on others.
- The usual punishment applies because you are sick and tired of committing somebody else's good work!

1.2. Mentor/Co-Mentor

Reasons for a co-mentorship:

- Significant timezone differential. Accessible, interactive mentor(s) available via IM is extremely helpful!
- Potential language barrier. Yes, FreeBSD is very English oriented, as is most software development, however, having a mentor who can speak a native language can be very useful.
- ENOTIME! Until there is a 30 hour day, and an 8 day week, some of us only have so much time to give. Sharing the load with somebody else will make it easier.
- A rookie mentor can benefit from the experience of a senior committer/mentor.
- Two heads are better than one.

Reasons for sole mentorship:

- You do not play nicely with others.

- You prefer to have a one-on-one relationship.
- The reasons for co-mentorship do not apply to you.

1.3. Expectations

We expect mentors to review and test-build all proposed patches, at least for an initial period lasting more than a week or two.

We expect that mentors should take responsibility for the actions of their mentee. A mentor should follow up with all commits the mentee makes, both approved and implicit.

We expect mentors to make sure their mentees read the Porter's Handbook, the PR handling guide, and the Committer's Guide. While it is not necessary to memorize all the details, every committer needs to have an overview of these things to be an effective part of the community (and avoid as many rookie mistakes as possible).

1.4. Selecting a Mentee

There is no defined rule for what makes a candidate ready; it can be a combination of number of PRs they have submitted, the number of ports maintained, frequency of ports updates and/or level of participation in a particular area of interest like GNOME, KDE, Gecko or others.

A candidate should have almost no timeouts, be responsive to requests, and generally helpful in supporting their ports.

There must be a history of commitment, as it is widely understood that training a committer requires time and effort. If somebody has been around longer, and spent the time observing how things are done, there is some anticipation of accumulated knowledge. All too often we have seen a maintainer submit a few PRs, show up in IRC and ask when they will be given a commit bit.

Being subscribed to, and following the mailing lists is very beneficial. There is no real expectation that submitting posts on the lists will make somebody a committer, but it demonstrates a commitment. Some mails offer insights into the knowledge of a candidate as well how they interact with others. Similarly participating in IRC can give somebody a higher profile.

Ask six different committers how many PRs a maintainer should submit prior to being nominated, and you will get six different answers. Ask those same individuals how long somebody should have been participating, same dilemma. How many ports should they have at a minimum? Now we have a bikeshed! Some things are just hard to quantify, a mentor will just have to use their best judgement, and hope that portmgr agrees.

1.5. Mentorship Duration

As the trust level develops and grows, the mentee may be granted "implicit" commit rights. This can include trivial changes to a Makefile, pkg-descr etc. Similarly, it may include `PORTVERSION` updates that do not include `plist` changes. Other circumstances may be formulated at the discretion of the Mentor. However, during the period of mentorship, a port version bump that affects dependent ports should be checked by a mentor.

Just as we are all varied individuals, each mentee has different learning curves, time commitments, and other influencing factors that will contribute to the time required before they can "fly solo". Empirically, a mentee should be observed for at least 3 months. 90-100 commits is another target that a mentor could use before releasing a mentee. Other factors to consider prior releasing a mentee are the number of mistakes they may have made, QATs received etc. If they are still making rookie mistakes, they still require mentor guidance.

1.6. Mentor/Co-Mentor Debate

When a request gets to portmgr, it usually reads as, "I propose 'foo' for a ports commit bit, I will co-mentor with 'bar'". Proposal received, voted, and carried.

The mentor is the primary point of contact or the "first among equals", the co-mentor is the backup.

Some reprobate, whose name shall be withheld, made the first recorded co-mentor commit. Similar co-mentor commits have also been spotted in the src tree. Does this make it right? Does this make it wrong? It seems to be part of the evolution of how things are done.

1.7. Expectations

We expect mentees to be prepared for constructive criticism from the community. There’s still a lot of "lore" that is not written down. Responding well to constructive criticism is what we hope we are selecting for by first reviewing their existing contributions on IRC and mailing lists.

We warn mentees that some of the criticism they receive may be less "constructive" than others, (whether through language communication problems, or excessive nit-picking), and that dealing with this gracefully is just part of being in a large community. In case of specific problems with specific people, or any questions, we hope that they will approach a portmgr member on IRC or by email.