

Guidelines:WorkingCommunity

(Taken from MoodleDocs)

When you submit a patch (or new module, plugin or filter) for acceptance, it will be reviewed on its technical merits and those alone. So, what should you be expecting?

- Criticism
- Comments
- Requests for change
- Requests for justification
- Silence
- Pointers to another subproject with the same goals

Remember, this is part of getting your patch into one of Openideas projects you have to be able to take criticism and comments about your patches, evaluate them at a technical level and either rework your patches or provide clear and concise reasoning as to why those changes should not be made. If there are no responses to your posting, wait a few days and try again, sometimes things get lost in the huge volume.

What should you not do?

- Expect your patch to be accepted without question
- Become defensive
- Ignore comments
- Resubmit the patch without making any of the requested changes

In a community that is looking for the best technical solution possible, there will always be differing opinions on how beneficial a patch is. You have to be cooperative, and willing to adapt your idea to fit within Openideas. Or at least be willing to prove your idea is worth it. Remember, being wrong is acceptable as long as you are willing to work toward a solution that is right.

Differences between the Openideas community and corporate structures

The Openideas community works differently than most traditional corporate development environments. Here is a list of things that you can try to do to try to avoid problems:

Good things to say regarding your proposed changes:

- "This solves multiple problems."
- "This deletes 2000 lines of code."
- "Here is a patch that explains what I am trying to describe."
- "I tested it on 5 different databases..."
- "Here is a series of small patches that..."
- "This increases performance on typical machines..."

Bad things you should avoid saying:

- “We did it this way in WebCT/Blackboard, so therefore it must be good...”
- “I’ve been doing this for 20 years, so...”
- “It makes this proprietary benchmark go faster”
- “This is required for my company to make money”
- “This is for our Enterprise product line.”
- “Here is my 1000 page design document that describes my idea”
- “I’ve been working on this for 6 months...”
- “Here’s a 5000 line patch that...”
- “I rewrote all of the current mess, and here it is...”
- “I have a deadline, and this patch needs to be applied now.”

Another way the Openideas community is different than most traditional software engineering work environments is the faceless nature of interaction. One benefit of using email and forums as the primary forms of communication is the lack of discrimination based on gender or race. The Openideas work environment is accepting of women and minorities because all you are is an email address. The international aspect also helps to level the playing field because you can’t guess gender based on a person’s name. A man may be named Andrea and a woman may be named Pat.

The language barrier can cause problems for some people who are not comfortable with English. A good grasp of the language can be needed in order to get ideas across properly on forums, so it is recommended that you check your emails to make sure they make sense in English before sending them.

Consider asking or discussing your proposal first

It is surprising how often a forum posting along the lines of “I wish to propose xxxx” can produce interesting results. You may find that somebody has 95% completed the thing you want to do. You may find somebody has done something similar that you can adapt. You may even have issues raised that you never thought of that may result in a change of direction or abandoning it altogether (a good thing as it can save hours of work).

Break your changes up

The Openideas community does not gladly accept large chunks of code dropped on it all at once. The changes need to be properly introduced, discussed, and broken up into tiny, individual portions. This is almost the exact opposite of what companies are used to doing. Your proposal should also be introduced very early in the development process, so that you can receive feedback on what you are doing. It also lets the community feel that you are working with them, and not simply using them as a dumping ground for your feature. However, don’t post 50 times at once in the forums!

The reasons for breaking things up are the following:

- 1. Small patches increase the likelihood that your patches will be applied, since they don’t take much time or effort to verify for correctness. A 5 line patch can be applied by a maintainer with barely a second glance. However, a 500 line patch may take hours to review for correctness (the time it takes is exponentially proportional to the size of the patch, or something). Small patches also make it very easy to debug when something goes wrong. It’s much easier to back out patches one by one than it is to dissect a very large patch after it’s been applied (and

broken something).

- 2. It's important not only to send small patches, but also to rewrite and simplify (or simply re-order) patches before submitting them.

It may be challenging to keep the balance between presenting an elegant solution and working together with the community and discuss your unfinished work. Therefore it is good to get early in the process to get feedback to improve your work, but also keep your changes in small chunks that they may get already accepted, even when your whole task is not ready for inclusion now.

Also realize that it is not acceptable to send patches for inclusion that are unfinished and will be "fixed up later."

Justify your change

Along with breaking up your patches, it is very important for you to let the Openideas community know why they should add this change. New features must be justified as being needed and useful.

From:

<https://pandorafms.com/manual/> - **Pandora FMS Documentation**

Permanent link:

<https://pandorafms.com/manual/guidelines/workingcommunity>

Last update: **2021/11/05 12:05**

