Proc1: OpenXT Project Process and Workflow

> DRAFT

This portion of the governance documentation deals with day to day processes and workflows on the OpenXT project. The processes listed here are less formal and have evolved as work has progressed on the project. These processes are also subject to change if it is needed or if the community desires it.

Contributing

Guidelines for contributing have been in place for some time now. Please refer to:

How to contribute

Coding pages

Source Code

All source code (and certain other material like documents and the BVT test harness) resides in public git repositories on GitHub. There are currently two organizations:

OpenXT Organization

OpenXT-Extras Organization

The first is the primary organization where all the currently active projects reside. The latter is primarily for two purposes. The first is to collect related projects that are not central to OpenXT (like build machine setup scripts e.g.). Secondly it is used as a place to move repositories when they become obsolete. The allows the project to leverage GitHub's redirects functionality. See the section "Redirects and Git Remotes" on the Transferring a repository page (as well as the Renaming a repository page).

GitHub Pull Requests

The primary method for contributing on the OpenXT project is using the GitHub Pull Request process. See the GitHub documentation on Pull Requests for a full description on how it works:

Using Pull Requests

OpenXT Pull Request Process

The following process has evolved as the standard way in which the vast majority of Pull Request (PR) submissions are handled.

- Monitor all new and existing PRs. Some PRs may be open but pending further evaluation because someone brought up a question or identified an issue that has not been addressed.
- Some PRs can get rejected right up front. These are ones that have obvious issues before any in depth review or testing is done.
  - The PR is a large feature change or addition and should have had an RFC done first.
  - The PR does not have a ticket. Aside from very minor PRs e.g. say one to un-break the build, all PRs should have a ticket.
  - Someone makes a specific request for something like documentation, diagrams or other supporting material before the PR is considered.
- Try to determine who might best be able to review and test the PRs. In a lot of cases this is reasonably obvious and one of the repository maintainers will just take it on, but there are challenging cases:
  - When the repository maintainers do not have the knowledge or familiarity with the language(s) the PR contains (e.g. JavaScript or Haskell). The maintainers will attempt to locate someone with those skills.
  - When the repository maintainers cannot reproduce the issue because of requirements for an appropriate environment or hardware. The maintainers will attempt to find someone who does.

        - When the repository maintainers know there are others who would be much better suited to evaluate a PR for a variety of reasons.
- Code review and static analysis of the PRs. The depth and nature of this is very specific to each PR and the language(s) it is written in. Maintainers take a best effort approach to this given the resources available.
- Determine what the best approach to test the PR is. This is very subjective and specific to each PR. The basic approaches are:
    - A full build on the build server with the PR specific changes in it is done then tested against instructions the author provides. This is done for large PRs (like new features) or PRs that not easily tested in isolation.
    - A partial build of the modules/components is done by one of us and tested in an existing build against instructions the author provides. This is done for smaller scope PRs where the changes are more isolated.
    - A full build on the build server with the PR specific changes where the success of the build is the only acceptance criterion. This is rare and is usually related to changes that only effect the build environment.
    - Testing is not done and we rely solely on inspection and review. This is exceedingly rare and only happens when the change is trivial or there are extenuating circumstances. The reasons for using this approach would be cited in the PR and/or the ticket.
- As a rule, people with commit access do not review, test and merge their own work. They must have someone else do this for them and, importantly, have some one else with commit access do the merge. The exception to the rule is really only cases where the build is broken and needs to be un-broken ASAP.
- A PR will hopefully reach the state where it can be merged. This could follow numerous paths as indicated above. When reviewers and testers have signed off, usually with an approval comment, the committer will then note that the PR is ready for merging by stating something like "will merge soon". This gives a final window for anyone to raise any issue or objection.
- Occasionally there will be PRs that have issues or concerns that the author has not addressed or commented on. In these cases we ping the author over a period of weeks/months and ask them to address the issues/concerns or close the PR out (and try again later hopefully). If after several attempts and a reasonably long period nothing happens, a dormant PR will be closed with an indication of the reason.
- Finally there is the (rare) case where the PR cannot be considered further. In general, all community concerns and questions must be addressed before the PR can proceed to a merge-able state. If there is conflict in the community over the PR the final recourse is escalation of the issue to the Project Governance Board.

GitHub Teams and Roles

To date the definition of GitHub teams and project roles has been ill defined. There are a number of teams centered around functional areas in OpenXT (see https://github.com/orgs/OpenXT/teams). The process for adding a member to a team is basically to nominate the person on the mailing list and vote with +/-1. The Owners and Admin teams have remained static since the formation of the OpenXT project. The process of defining and managing teams and roles needs to be more well defined going forward. It will also need to accommodate different roles and procedures on different layers in OpenXT.

Changes to this Governance Document

Changes to the processes and workflows described in this document (and related process documents that are linked from this) are generally decided by community discussion. Some of the processes have just evolved with no objections from the community. If any conflict or issue arose with this mode of changing these processes or workflows it would result in an escalation to the Project Governance Board.

License of this Governance Document

Revision History of this Governance Document