Information organisation

Lix has a lot of information as a project, and we want to make it accessible in a way that it can be found later if necessary.

There are various tools for keeping information in the project, and they have different purposes

Chat

Chat is good for:

- Information that will be meaningless in hours
- Ephemeral discussions, in general

The chat is expected to move way too fast to follow. As such:

- Don't write things in chat that you expect to be found later
- If discussions of design happen, write them down, at least by copy pasting into a pad and adding the pad to the index
- If tips and tricks are discussed, please write them down
- Please do reviews on Gerrit so they are archived, rather than in chat
- Write things down in the log if they are expected to be found

Wiki (https://wiki.lix.systems)

The wiki is good for:

- Development process information, like you would find on https://rustc-dev-guide.rust-lang.org/ for the Rust compiler, for instance.
- Design documents

The wiki is not good for:

- User facing documentation
- Documentation that deserves to be reviewed
- Actively writing a document in real time with others

Markdown files in the Lix repo

Markdown files in the Lix repo are good for:

- Maintaining things that are tied directly to the code

- Documentation that needs to be reviewed
- User facing documentation

Markdown files in the Lix repo are bad for:

- Quickly iterating on things
- Design documents

Forgejo issues (https://git.lix.systems)

Our primary issue tracker is Forgejo issues.

We are currently attempting to use the Forgejo project boards feature to communicate what people are working on; it may be replaced with better Kanban software in the future. When making project boards on Forgejo, make them on the `lix-project` organisation unless they are strictly contained within one project.

The issue tracker is good for:

- Actionable work
- Bugs

The issue tracker is not good for:

- Dreams or otherwise not actionable information that is a long term goal
- Private information
- Information that needs to be found later, design documentation

Where to put an issue

- lix-project/lix, if it is contained within Lix (but is not more appropriate to put in the installer e.g.)
  - If it is an upstream bug, tag its equivalent `lix-import` on https://git.lix.systems/nixos/nix, and get someone with the bot token to run the issue import script in `maintainers/issue_import.py`. (FIXME: someone ought to put that on a cron job)
  - Please never file issues on our Nix mirror.
- lix-project/installer, if it is the installer
- lix-project/web-services, if it is infrastructure related
- lix-project/meta, if it does not fit anywhere obvious and you just need it to put it on a board
- lix-project/nixos-module, if it is a packaging bug in that specifically

Gerrit (https://gerrit.lix.systems)

Gerrit is good for:

- Reviewing code
- Maintaining a record of code reviews

Gerrit is not good for:

- Persisting information in a discoverable way to anyone in the future
- Documentation

Pad (https://pad.lix.systems [private])

We anticipate that the pad service will be semi-permanently private by default, since it doesn't support ACLs.

The pad is good for:

- Sketching out drafts of documents that aren't ready yet
- Planning private things
- Generally getting people on the same page about things in active design, making what might be meeting notes, or similar.

The pad is not good for:

- Information that should be available to users (unless it is planned to move)
- Information that is not actively changing

It is very important that pads remain temporary in nature, and are quickly moved to their final destination, e.g. the Wiki.

N.B. For users who aren't in the Lix core team, the service returns 500 when you attempt to login. This is a known issue that can't be fixed.

Revision #4
Created 2024-03-24 14:07:55 UTC by jade
Updated 2025-06-27 21:54:23 UTC by piegames