Contributing to Conan

=====================

The following summarizes the process for contributing to the Conan project.

Community

---------

Conan is an Open Source MIT licensed project.

Conan is developed by the Conan maintainers and a great community of contributors.

Dev-flow & Pull Requests

------------------------

Conan follows the ["GitFlow"](https://datasift.github.io/gitflow/IntroducingGitFlow.html) branching model.

Issues are triaged and categorized mainly by type (feature, bug...) complexity (high, medium...) and priority (high, medium...) using GitHub

labels. Then they are moved to a stage called "queue" when they are considered to be ready for implementation.

To contribute follow the next steps:

1. Comment in the corresponding issue that you want to contribute the feature/fix proposed. If there is no open issue, we strongly suggest

to open one to gather feedback.

2. Check that the issue has been staged to "queue" or ask @conan-io/barbarians to do it. This helps in terms of validation and discussion of

possible implementation of the feature/fix.

3. Fork the [Conan main repository](https://github.com/conan-io/conan) and create a `feature/xxx` branch from the `develop` branch and develop

your fix/feature as discussed in previous step.

4. Try to keep your branch updated with the `develop` branch to avoid conflicts.

5. Open a pull request, and select `develop` as the base branch. Never open a pull request to ``release/xxx`` branches.

6. Add the text (besides other comments): "fixes #IssueNumber" in the body of the PR, referring to the issue of step 1.

7. Submit a PR to the Conan documentation about the changes done providing examples if needed.

The ``conan-io`` organization maintainers will review and help with the coding of tests. Finally it will be assigned to milestone.

Each milestone corresponds to a Conan release, a branch `release/xxx` will be opened from the `develop` branch.

When the branch is ready, a tag corresponding to the version will be pushed and it will be merged into `develop` branch.

The release plan is done monthly but dates are not fixed, the ``conan-io`` organization maintainers reserve the right of altering the issues

of each milestone as well as the release dates.

Issues

------

If you think you found a bug in Conan open an issue indicating the following:

- Explain the Conan version, Operating System, compiler and any other tool that could be related to the issue.

- Explain, as detailed as possible, how to reproduce the issue. Use git repos to contain code/recipes to reproduce issues.

- Include the expected behavior as well as what actually happened.

- Provide output captures (as text).

- Feel free to attach a screenshot or video illustrating the issue if you think it will be helpful.

For any suggestion, feature request or question open an issue indicating the following:

- Questions and support requests are always welcome.

- Use the [question] or [suggestion] tags in the title.

- Try to explain the motivation, what are you trying to do, what is the pain it tries to solve.

- What do you expect from Conan.

We use the following tags to control the status of the issues:

- \*\*triaging\*\*: Issue is being considered or under discussion. Maintainers are trying to understand the use case or gathering the necessary

information.

- \*\*queue\*\*: Issue has been categorized and is ready to be done in a following release (not necessarily in the next one).

- \*\*in-progress\*\*: A milestone has previously been assigned to the issue and is now under development.

- \*\*review\*\*: Issue has a PR associated that solves it (remember to use the GitHub keywords "closes #IssueNumber", "fixes #IssueNumber"...

in the description of the PR).

- \*\*closed via PR\*\*: A PR with the fix or new feature has been merged to `develop` and the issue will be fixed in the next monthly release.

Code of conduct

---------------

Try to be polite, Conan maintainers and contributors are really willing to help and we enjoy it.

Please keep in mind these points:

- There are limited resources/time and not all issues/pull requests can be considered as well as we would like.

- ``conan-io`` maintainers can tag/close/modify any opened issue and it should not be interpreted as a rude or disrespectful action. It

\*\*always\*\* responds to organizational purposes. A closed issue can be perfectly reopened or further commented.

- It is very hard to keep the project in good health in terms of technical debt, usability, serviceability, etc. If the proposed feature is

not clearly stated, enough information is not provided or it is not sure that would be a good feature for the future development of the project, it won't be accepted. The community plays a very important role during this discussion so it strongly encouraged to

explain the value of a feature for the community, the needed documentation and its use cases.

- Backwards compatibility and not breaking users' packages is very important and it won't be done unless there are very good reasons.

- You should not get bothered if you feel unattended, Conan is an Open Source project, not a commercial product. Try to explain what you

really need and we will try to help you.

Code style

----------

- In general, follow [pep8](https://www.python.org/dev/peps/pep-0008/)

- Limit all lines to a maximum of 101 characters (`Right margin` setting in PyCharm)

- Start new files with [encoding declaration](https://www.python.org/dev/peps/pep-0263/):

```

# -\*- coding: utf-8 -\*-

```

- Specify imports in the following order: system, `blank line`, 3rd-party, `blank line`, own; all sorted alphabetically:

```

import os

import platform

import shutil

from nose.plugins.attrib import attr

from tqdm import tqdm

from conans.client.tools import which

from conans.errors import ConanException

from conans.model.version import Version

```

- Write unit tests, if possible