# Cucumber Community Contributing Guide 1.0

This document describes a very simple process suitable the projects

in the Cucumber ecosystem.

The goal of this document is to create a contribution process that:

\* Encourages new contributions.

\* Encourages contributors to remain involved.

\* Avoids unnecessary processes and bureaucracy whenever possible.

\* Creates a transparent decision making process which makes it clear how

contributors can be involved in decision making.

Also see [guidelines for documentation contributions](https://cucumber.io/docs/community/contributing-to-documentation/)

This document is based on the [Node.js Community Contributing Guide](https://github.com/nodejs/TSC/blob/master/BasePolicies/CONTRIBUTING.md).

## Newcomers

Committers maintain a list of

[newcomer friendly issues](https://github.com/cucumber/cucumber/issues?q=is%3Aissue+is%3Aopen+label%3A%22good+first+issue%22+no%3Aassignee+)

that are suitable for aspiring contributors.

## Vocabulary

\* A \*\*Contributor\*\* is any individual creating or commenting on an issue or pull request.

\* A \*\*Committer\*\* is a subset of contributors who have been given write access to the repository.

\* A \*\*TC (Technical Committee)\*\* is a group of committers representing the required technical

expertise to resolve rare disputes.

# Logging Issues

Log an issue for any question or problem you might have. When in doubt, log an issue,

any additional policies about what to include will be provided in the responses.

Committers may direct you to another repository, ask for additional clarifications, and

add appropriate metadata before the issue is addressed.

Please be courteous, respectful, and every participant is expected to follow the

[Code of Conduct](CODE\_OF\_CONDUCT.md).

GitHub issues should always be logged in the [Cucumber monorepo](https://github.com/cucumber/cucumber/issues).

# Contributions

Any change to resources in this repository must be through pull requests. This applies to all changes

to documentation, code, binary files, etc. Even long term committers and TC members must use

pull requests, except for trivial changes.

GitHub pull requests should usually be submitted against the monorepo here. The main exception to

this is for the main cucumber program; which should be submitted to the subrepos (i.e. `cucumber-ruby`).

Pull requests should be independent so they can be merged/rejected independently of other

pull requests. Every pull request must be made on a separate branch, branched off from the HEAD

of the master branch. No pull requests should depend on other pull requests or be branched off

from non-master branches. Where a Pull request requires changes in multiple languages, assistance

can be sought from others (If you're not comfortable in multiple languages), who can either

raise another pull request for the separate implementation or commit to yours (Whichever is easier).

No pull request can be merged without being reviewed.

For non-trivial contributions, pull requests should sit for at least 36 hours to ensure that

contributors in other timezones have time to review. Consideration should also be given to

weekends and other holiday periods to ensure active committers all have reasonable time to

become involved in the discussion and review process if they wish.

The default for each contribution is that it is accepted once no committer has an objection.

During the review process committers may also request that a specific contributor who is most

versed in a particular area gives a "LGTM" before the PR can be merged. There is no additional

"sign off" process for contributions to land. Once all issues brought by committers are

addressed it can be merged in by any committer.

In the case of an objection being raised in a pull request by another committer, all involved

committers should seek to arrive at a consensus by way of addressing concerns being expressed

by discussion, compromise on the proposed change, or withdrawal of the proposed change.

If a contribution is controversial and committers cannot agree about how to get it to land

or if it should land then it should be escalated to the TC. TC members should regularly

discuss pending contributions in order to find a resolution. It is expected that only a

small minority of issues be brought to the TC for resolution and that discussion and

compromise among committers be the default resolution mechanism.

# Becoming a Committer

All contributors who land a non-trivial contribution will be on-boarded in a timely manner,

and added as a committer, and be given write access to the repository.

Committers are expected to follow this policy and continue to send pull requests, go through

the proper review process, and have other committers merge their pull requests.

# TC Process

The TC uses a "consensus seeking" process for issues that are escalated to the TC.

The group tries to find a resolution that has no open objections among TC members.

If a consensus cannot be reached that has no objections then a majority wins vote

is called. It is also expected that the majority of decisions made by the TC are via

a consensus seeking process and that voting is only used as a last-resort.

Resolution may involve returning the issue to committers with suggestions on how to

move forward towards a consensus. It is not expected that a meeting of the TC

will resolve all issues on its agenda during that meeting and may prefer to continue

the discussion happening among the committers.

Members can be added to the TC at any time. Any committer can nominate another committer

to the TC and the TC uses its standard consensus seeking process to evaluate whether or

not to add this new member. Members who do not participate consistently at the level of

a majority of the other members are expected to resign.