

GitHub Issues Management

Since the ToIP Foundation, like most other Linux Foundation projects, uses GitHub as our information repository "of record", it is recommended for Working Groups and Task Forces to use [GitHub issues](#) during the development of any ToIP deliverable (specification, template, white paper, etc.) This page documents the recommended process (originally developed by the [Technology Architecture Task Force](#)) for managing and resolving GitHub issues.



This process is recommended for issue management *even if the deliverable itself is not in GitHub*. For example, if the deliverable is being developed as a Google doc, this process can still be used to discuss and resolve any issue not easily handled by a Google doc comment thread.

Process

1. **Setup.** First, the Working Group or Task Force ("group") should contact the ToIP Foundation Program Manager (currently [Michelle Janata](#)) to request setup of a GitHub repo for the deliverable. The group should also request that [Github Discussions](#) be turned on for the repo so the Discussions feature is available to the group for Q&A or other discussions that are not necessarily issues (but can be quickly converted into a formal issue when needed).
2. **Editors.** The Working Group or Task Force ("group") should appoint an Editors team who will take on the job of reviewing issues, assigning group members to an issue, and proposing when an issue is ready for closure.
 - a. As a general rule, any one member of the Editors team can perform an action permitted under this role — it does not require consensus among all the Editors. However the Editors are trusted to use their judgement about when they should consult the other editors first or seek the consensus of the whole group.
 - b. The members of the Editor team should be published on the group's home page and acknowledged in the final deliverable for their extra contribution.
3. **Maintainers.** The Editors should in turn appoint a set of Maintainers who have the Github skills (and the necessary permissions) to accept PRs and publish versions of the deliverable. Typically a subset of the Editors serve as Maintainers, but all the Editors can serve in this role, or it can be assigned to others in the group.
4. **Contributors.** This includes anyone else in the group who actively contributes to either discussions, issues, or PRs.
5. **Labels.** The Editors should agree on a set of labels to categorize and prioritize issues for resolution. See the recommended starting set below. While any group member should be able to apply labels to issues, it is the Editors job to ensure labels are applied consistently, fairly, and timely.
6. **Assignments.** Any group member should be able to assign an issue to another group member. It is the Editors job to try to make sure issues have assignees, and that issues are assigned consistently, fairly, and timely. If an assignee is not progressing with an issue, the Editors can re-assign it as necessary.
7. **Subgroups.** If an issue appears to require in-depth discussion and analysis, the Editors should assign a subgroup to tackle the issue and come back to the group with a proposed resolution. This subgroup should:
 - a. Keep as much of their discussion as possible within [GitHub Issues](#) — and, if necessary, [Github Discussions](#). If any substantive discussions take place in other channels (e.g., Slack) or proposals are drafted outside of GitHub (e.g., in a Google doc), *they must be copied into GitHub to create a permanent public audit trail*.
 - b. Hold special calls/meetings if needed, but record those meetings and document key discussion points and decisions *and copy those to GitHub*.
 - c. Develop a proposed resolution to the issue (along the lines of an ISO "Technical Report").
 - d. Return with a proposal (text and diagrams) for resolution of the issue (along the lines of an ISO "Technical Spec"). Ideally this proposal is in a form that can be easily: a) turned into a PR (for a GitHub document), or b) copy-and-pasted as a revision to a Google doc or other format.
8. **Closure.** If the Editors believe an issue has been resolved via one or more PRs that have been accepted and merged, then one of the Editors should apply the label `status: PR-merged` and close the issue. If the Editors believe consensus has been achieved about some other resolution of the issue — and that resolution is fully documented in the issue — then one of the Editors should apply the label `status: last-call`.
 - a. Once that label has been applied, a group member **MUST** object to closure by making a comment on the issue **within 5 calendar days** to reopen discussion of the issue.
 - b. If there is no objection within 5 calendar days, the proposed resolution shall be applied to the deliverable by one of the Maintainers and one of the Editors shall close the issue with no further discussion.
 - c. If there is an objection, the Editors will take it to a group meeting to reach final consensus on closure.

Recommended Labels

We recommend the following set of labels as a starter set. They quickly communicate the priority, type, and status of an issue. **As a general rule, at any one point in time, an issue should only have one label from each of these three categories.** Your group can augment these with additional deliverable-specific labels.

Label	Usage
<ul style="list-style-type: none">• priority: critical• priority: high• priority: medium• priority: low	<ul style="list-style-type: none">• Progress on this issue is critical to the group's forward progress.• It is important for the group to resolve this issue soon.• This issue is important to resolve before the next release.• This issue is "nice to have" for the next release, but could be deferred if time runs out.

<ul style="list-style-type: none"> • type: content • type: editorial • type: terminology • type: correction • type: duplicate • type: formatting • type: figure • type: admin 	<ul style="list-style-type: none"> • The issue involves normative content; resolution requires group consensus. • The issue only involves wording and not normative content. • The issue involves defining a term or other glossary-related work. • The issue is fixing a recognized problem in the current version. • The issue overlaps another issue and the two should be merged. • The issue involves fixing formatting. • The issue involves a figure that is missing or needs to be revised. • The issue is administrative and NOT about the deliverable.
<ul style="list-style-type: none"> • status: unassigned • status: in-progress • status: needs-review • status: blocked • status: on-hold • status: deferred • status: abandoned • status: PR-needed • status: PR-in-progress • status: PR-completed • status: PR-accepted • status: PR-merged • status: last-call 	<ul style="list-style-type: none"> • The issue is new and has not yet been assigned to anyone. • The issue has been assigned and work is in progress. • A resolution (or concrete step forward) has been proposed and needs review. • Progress is currently blocked; the block should be explained in a comment. • Progress is currently on hold; the reason should be explained in a comment. • There is consensus this issue can be deferred to a subsequent version. • There is consensus this issue can be abandoned. • There is consensus on this issue and it is now waiting for a PR to be submitted. • The issue is linked to a PR that is in progress. • The issue is linked to a PR that is complete and waiting for review. • The issue is linked to a PR that has been accepted and is waiting for merge. • The issue is linked to a PR that has been merged; this issue can now be closed. • The issue has been resolved by some other mechanism documented in the comments and is now in 5 day last call.