

DecisionMaking

Decision-Making Process in the Git Project

=====

Introduction ----- This document describes the current decision-making process in the Git project. It is a descriptive rather than prescriptive doc; that is, we want to describe how things work in practice rather than explicitly recommending any particular process or changes to the current process.

Here we document how the project makes decisions for discussions (with or without patches), in scale larger than an individual patch series (which is fully covered by the SubmittingPatches document).

Larger Discussions (with patches) ----- As with discussions on an individual patch series, starting a larger-scale discussion often begins by sending a patch or series to the list. This might take the form of an initial design doc, with implementation following in later iterations of the series (for example, [adding unit tests](#) or [config-based hooks](#)), or it might include a full implementation from the beginning. In either case, discussion progresses the same way for an individual patch series, until consensus is reached or the topic is dropped.

Larger Discussions (without patches) ----- Occasionally, larger discussions might occur without an associated patch series. These may be very large-scale technical decisions that are beyond the scope of even a single large patch series, or they may be more open-ended, policy-oriented discussions (examples: [introducing Rust](#) or [improving submodule UX](#)). In either case, discussion progresses as described above for general patch series.

For larger discussions without a patch series or other concrete implementation, it may be hard to judge when consensus has been reached, as there are not any official guidelines. If discussion stalls at this point, it may be helpful to restart discussion with an RFC patch series (such as a partial, unfinished implementation or proof of concept) that can be more easily debated.

When consensus is reached that it is a good idea, the original proposer is expected to coordinate the effort to make it happen, with help from others who were involved in the discussion, as needed.

For decisions that require code changes, it is often the case that the original proposer will follow up with a patch series, although it is also common for other interested parties to provide an implementation (or parts of the implementation, for very large changes).

For non-technical decisions such as community norms or processes, it is up to the community as a whole to implement and sustain agreed-upon changes. The project leadership committee (PLC) may help the implementation of policy decisions.

Other Discussion Venues ----- Occasionally decision proposals are presented off-list, e.g. at the semi-regular Contributors' Summit. While higher-bandwidth face-to-face discussion is often useful for quickly reaching consensus among attendees, generally we expect to summarize the discussion in notes that can later be presented on-list. For an example, see the thread [Notes from Git Contributor Summit](#) by James Ramsay.

We prefer that "official" discussion happens on the list so that the full community has opportunity to engage in discussion. This also means that the mailing list archives contain a more-or-less complete history of project discussions and decisions.

#2

1 ## 2026-10-31 19:44:53 UTC

1 ## 2026-10-31 19:44:54 UTC