Go Proposal Process: Clarity & Transparency

Go Proposals, Part 2

Russ Cox
August 7, 2019

research.swtch.com/proposals-clarity

[*I've been thinking a lot recently about the Go proposal process, which is the way we propose, discuss, and decide changes to Go itself. Like nearly everything about Go, the proposal process is an experiment, so it makes sense to reflect on what we've learned and try to improve it. This post is the second in a series of posts about what works well and, more importantly, what we might want to change.*]

In the discussions I've had with contributors recently, the most important takeaway for me was that there needs to be more clarity about the overall process for making decisions and also more transparency, to make it easier to observe and follow along. The introduction of the proposal process was an important step toward those concerns, but more is needed.

Documentation

One of the explicit goals we listed in 2015 was to make the process clear to new contributors, but many people are unaware that the proposal process is documented at all. It is linked on the contributor guidelines, but many people who participate in the process do not send code changes and thus have no reason to read that doc. One idea raised was to have gopherbot comment with a link whenever the proposal label is added to a GitHub issue.

The proposal process description is also missing some details. It links to a few important talks from 2015 but does not explicitly say what the relevant material is; not many people are going to watch the entire talk to find out. And in some discussions people who have read the proposal process doc have told us they didn't even know the talks existed, despite being linked in the doc. Obviously those links are not working.

Also, we introduced a longer process, with two rounds of review, for Go 2 language changes, but it is only documented in a blog post. That blog post is linked from the proposal process doc, but having everything on one page would reach more readers.

I filed issue 33524 to make sure we update the README to stand alone.

Issue Status

Once you know about the proposal process, there is still not much clarity about the state of any particular proposal. The only labels we have are Proposal and Proposal-Accepted. (Proposals that are closed without the Proposal-Accepted label are the ones that have been declined.) We've already added two rounds of review for Go 2 changes, and we may want to formalize the idea of design drafts for large changes, prior to the usual proposal process (a topic for a future post). As the process gets more formalized, it would help if there was a clear answer to "where is this specific issue in the process?" Other bug trackers allow defining custom metadata fields that we could use to store this information. GitHub being GitHub, there is no direct support for any of this.

A possible solution to both of these issues—not knowing about the process and not knowing where the issue is in the process—would be for gopherbot to insert and maintain a block of text at the top of the issue description with a link to the process, information about the state of the issue, and any other information we might identify that would help people arriving at the issue learn where things stand. I filed issue 33522 for that idea.

Review Minutes

One of the explicit goals of the process was to "commit to timely evaluations of proposals" (quoting Andrew Gerrand's talk). Initially, proposals piled up without timely evaluation. In late 2015, Andrew started a regular meeting to help make sure that core Go team members delivered on our promise of timely evaluation. The primary activity of this meeting is to make sure proposals are moving along in the process, receiving feedback, and not being forgotten.

We documented the meetings explicitly in 2018:

> **Proposal Review**
>
> A group of Go team members holds "proposal review meetings" approximately weekly to review pending proposals.
>
> The principal goal of the review meeting is to make sure that proposals are receiving attention from the right people, by cc'ing relevant developers, raising important questions, pinging lapsed discussions, and generally trying to guide discussion toward agreement about the outcome. The discussion itself is expected to happen on the issue tracker, so that anyone can take part.
>
> The proposal review meetings also identify issues where consensus has been reached and the process can be advanced to the next step (by marking the proposal accepted or declined or by asking for a design doc).

Even so, it has come up a few times in online discussions and also at contributor day that people don't understand who the proposal review group is or what it does. More transparency here would help as well.

At about the same time as we documented the meetings (at least as best I remember), I created a "team" named @golang/proposal-review on GitHub to try to make it clear who was in the meetings. Unfortunately, I didn't know at the time that GitHub never allows non-members of an organization to view team membership lists, even when the group is "public." So while nearly all Go project contributors can see the list, everyone else cannot. For the record, today that group is Andy Bonventre, Brad Fitzpatrick, Robert Griesemer, Ian Lance Taylor, Rob Pike, Steve Francia, and me, although not everyone attends every meeting.

One suggestion made at contributor day was to publish minutes of the review meetings. All the actions we take are visible on the GitHub issues themselves, but there was no easy way to aggregate them and see the review work as a whole.

As of yesterday's meeting, we have started collecting minutes in issue 33502.

Keeping the minutes in an issue enables easy cross-referencing in both directions between the issues and the minutes; seeing the links to the minutes appear in the proposal issues should help with understanding the process. The minutes also record who participated in each meeting, so that membership is clear even without visibility of the GitHub team, which we could probably now delete.

Writing those minutes yesterday forced us to be a bit more careful about explaining the reasons why we did things, which should be helpful in making the process clearer and will likely result in clarifications in the proposal process document itself, once we have a few more meetings with minutes under our belts.

Decisions

The goal of the proposal discussions is to reach clear agreement on whether to accept or decline a proposal. What happens when there is not clear agreement?

The original proposal process document (2015) said:

> In Go development historically, a lack of agreement means decline. If there is disagreement about whether there is agreement, adg@ is the arbiter.

In 2016, we realized that the Go user community was large enough that most important decisions would not reach complete agreement. After discussion on issue 17129, we updated the doc to explain what happens in that case. At the same time, adg@ moved on to other work and I took on the arbiter role. The document now said:

> The goal of the final discussion is to reach agreement on the next step: (1) accept or (2) decline. The discussion is expected to be resolved in a timely manner. If clear agreement cannot be reached, the arbiter (rsc@) reviews the discussion and makes the decision to accept or decline.

This version made it seem like, in the absence of clear agreement, the arbiter could make up any answer, which of course is not the case.

In the 2018 revisions that documented the review meetings, I tried to add more details that process, but in what turned out to be a differently misleading way, unfortunately conflating proposal review with arbitration of proposal decisions:

> **Consensus and Disagreement**
>
> The goal of the proposal process is to reach general consensus about the outcome in a timely manner.
>
> If general consensus cannot be reached, the proposal review group decides the next step by reviewing and discussing the issue and reaching a consensus among themselves. If even consensus among the proposal review group cannot be reached (which would be exceedingly unusual), the arbiter (rsc@) reviews the discussion and decides the next step.

In retrospect, this conflation of proposal review and contended proposal decisions has been an unhelpful source of confusion about the weekly proposal review meetings, which are nearly entirely concerned with the review/triage/gardening described in the previous section. In contrast, the handling of contended decisions happens very rarely, maybe once a year. These two activities—review and deciding contended proposals—have historically been done by the same people, but that is not a fundamental requirement. It probably makes sense to separate these two activities so that they can be handled by different groups. I filed issue 33528 for this.

Next

Again, this is the second post in a series of posts thinking and brainstorming about the Go proposal process. Everything about these posts is very rough. The point of posting this series—thinking out loud instead of thinking quietly—is so that anyone who is interested can join the thinking.

I encourage feedback, whether in the form of comments on these posts, comments on the newly filed issues, mail to rsc@golang.org, or your own blog posts (please leave links in the comments). Thanks for taking the time to read these and think with me.

The next post is about how the proposal process should scale down to tiny changes and up to large ones.