

Go Proposal Process: Representation

Go Proposals, Part 6

Russ Cox

October 3, 2019

research.swtch.com/proposals-representation

[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 sixth in a series of posts about what works well and, more importantly, what we might want to change.]

Who is Represented?

At the contributor summit, we considered the question of who is well represented or over-represented in the Go development process and who is under-represented.

The question of who is well represented matters because diverse representation produces diverse viewpoints that can help us reach better overall decisions in the process. We cannot possibly hear from every single person with relevant input; instead we can try to hear from enough people that we still gather all the important viewpoints and observations.

On the well represented or possibly over-represented list, we have members of the Go team; GitHub users who have time to keep up with discussions; English-speaking users; people who keep up with tech social media on sites like Hacker News and Twitter; and people who attend and give talks at Go conferences. On the under-represented list, we have non-GitHub users or users who can't keep up with GitHub discussions; non-English-speaking users; "heads down" users, meaning anyone who spends their time writing code to the exclusion of engaging on social media or attending Go conferences; business users in general; users with non-computer science backgrounds; users with non-programming language backgrounds; and non-users, people not using Go at all. As we consider ways to make the proposal process more accessible to more users and potential users, it is worth keeping these lists in mind to check whether new groups are being reached.

The proposal minutes help reach users who are on GitHub but can't keep up with all the discussions, by providing a single issue to star and get roughly weekly updates about which proposals are under active discussion and which are close to being accepted or declined. The minutes are an improvement for the "can't keep up with GitHub" category, but not for the other categories.

Reaching non-English-speaking users is one of the most difficult challenges. We on the Go team have attended Go conferences around the world to meet users in many countries; at many conferences there is simultaneous translation for the talks, which is wonderful. But there isn't simultaneous translation for our proposal discussions. I don't know whether the most significant proposals, like the latest version of the generics proposal, have been translated by users into their native languages, or if non-English-speakers muddle through with automatic translation, or something else. Twice in the past we've had questions about proposals that primarily affected Chinese users—specifically, whether to change case-based export for uncased languages and whether to build a separate Go distribution with different Go proxy defaults for China. In both these cases, we asked Asta Xie, the organizer of Gophercon China, to run a quick poll of users in a Chinese social media group of Go users. That was very helpful, but that doesn't scale to all proposals.

Reaching “heads down” programmers and business users, those not attending Go conferences or engaging in Go-related social media, probably means branching out to more non-Go-specific places to publicize the most important Go changes, and possibly Go itself.

The final group is users with non-computer science or non-programming language backgrounds. Go proposals are usually written assuming significant familiarity with Go and often also familiarity with computer science or programming language concepts. In general, that’s more efficient than the alternative. But especially for large changes it would help to have alternate forms that are accessible to a larger audience. Ideas include longer tutorials or walkthroughs, short video demos, and recorded video-based Q&A sessions.

What Does Representation Mean?

Part of Go’s appeal for me as a user, and I think for many Go users, is the fact that it feels like a coherent system in which the pieces fit together and complement each other well. In my 2015 talk and blog post, “Go, Open Source Community,” I said that “one of the most important things we the original authors of Go can offer is consistency of vision, to help keep Go Go.” I still believe that consistency of vision, to keep Go itself consistent, simple, and understandable, remains critically important to Go’s success.

In an interview with IT World in 2000, Dennis Ritchie (the creator of C) was asked about control over C’s development. He said:

On the other hand, the “open evolution” idea has its own drawbacks, whether in official standards bodies or more informally, say over the Web or mailing lists. When I read commentary about suggestions for where C should go, I often think back and give thanks that it wasn’t developed under the advice of a worldwide crowd. C is peculiar in a lot of ways, but it, like many other successful things, has a certain unity of approach that stems from development in a small group.

I see much of that same unity of approach in Go’s design by a small group. The flip side is that if you have a small number of people making decisions alone, they can’t know enough about the million or more Go developers and their uses to anticipate all the important use cases and needs. That is, while a small number of ultimate designers provides consistency of vision, it can just as well result in a failure of vision, in which a design fundamentally fails to plan for or adapt to a requirement that becomes important later but was not seen or well understood at the time.

A decision can be *both* elegant for its time and short-sighted for the future. For example, see the recent critique of Unix’s *fork* system call, “A `fork()` in the road,” by Andrew Baumann *et al.*, from HotOS ’19. In fact, nearly all designs end up being short-sighted given a long enough time scale. We’d be fortunate for all our designs to last the 50 years that *fork* has. The most important part is to guard against short-term failures of vision, not very long-term ones.

To me, the most important place for broad representation and inclusion is in the proposals and discussions, because, as I said above, diverse representation produces diverse viewpoints that can help us reach better overall decisions in the process. It can help us avoid at least the short-term and hopefully medium-term failures of vision. At the same time, I hope we can maintain a long-term consistency of vision in the design of Go, by the continued active involvement of the original designers. It seems to me that having both the original designers and a diverse set of other voices in our proposal discussions and consistently working toward consensus decisions will lead to the best outcomes and balances the desire for a consistency of vision against the need to avoid failures of vision.

Next

Again, this is the sixth 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 will be about how best to coordinate efforts across the Go community and ecosystem.