Add Feature Triaging Guide

10/11/2021 04:14 PM - jeremyevans0 (Jeremy Evans)

Ruby added a bug triaging guide in June 2019, and since then I've used it to reduce the number of open bugs in the tracker from over 1400 to about 300. Ruby currently has over 1200 open feature requests on the issue tracker. From a cursory review, many of these have already been implemented, and many are unlikely to be desired. I would like to add a feature triaging guide to Ruby and start triaging feature requests, with the goal of having open feature requests be features not yet implemented that the Ruby core team would like implemented or would consider patches for. By doing so, we can make it easier for potential contributors to Ruby to easily see possible contributions they could make.

I've added a pull request with a draft of the proposed guide: https://github.com/ruby/ruby/pull/4953

---

#1 - 10/12/2021 01:56 AM - Dan0042 (Daniel DeLorme)

+1

It would be good to cut down on stale feature requests, and the recommendations in the draft are all very reasonable.

I want to point out that I've made a number of feature requests that I would probably close myself if I had the ability. Unfortunately a submitter is not able to close his own requests. Also if the status has been changed to "feedback" I can't change it back to "open" after I've given the requested feedback. If these limitations were fixed I think it would improve the ability of the system to triage itself.

#2 - 10/12/2021 02:04 AM - mame (Yusuke Endoh)

Thank you always for your continuous and great contribution, I really appreciate it.

I read the guide draft. I understand that the two new things are as follows:

- We can close a three-month stale feature request after a committer responded "unlikely to be considered".
- We can close a one-year stale feature request even if a committer responded nothing.

They may be somewhat arguable, but personally, I think it's fine to try this if Jeremy will operate the triage with this policy. If there are any problems, we can think about adjusting the guide later.

#3 - 10/12/2021 02:52 AM - naruse (Yui NARUSE)

- Already Implemented Feature Requests should be closed
- Feature Requests That Would Probably Not Be Considered can be closed
- Newly Requested Features must not be closed unless Matz or maintainer explicitly rejected it

I agree these. I think those principle are right and it follow the current operation.

- Stale Feature Requests can be closed

I'm unclear about this.

I know some OSS projects introduce autoclose and it contributes the transparency of their issues. But unfortunately Ruby has many important but stale feature ideas for example namespace, macro-like something, defer, parser rewriting, and so on. I believe those feature requests shouldn't be closed...

- Reopening Triaged Feature Requests

I roughly agree this, but I think it needs some discussion.

- Why only Ruby developer can reopen it? (maybe the current permission only allows ruby committers to reopen..)
- the description which notes "when you reopen, you should add something new" sounds reasonable
- guideline of forking feature requests sounds useful, but I think this is not feature triaging.
Stale Feature Requests can be closed

I'm unclear about this. I know some OSS projects introduce autoclose and it contributes the transparency of their issues. But unfortunately Ruby has many important but stale feature ideas for example namespace, macro-like something, defer, parser rewriting, and so on. I believe those feature requests shouldn't be closed...

Note that this isn't an autoclose (I really hate autoclose). With autoclose, there is no judgment involved from a committer. In both the bug triaging and feature triaging guides, the committer is evaluating the issue and determining the appropriate course of action.

Stale feature requests would only be closed if:

- No committer indicates the feature is desired; and
- The committer doing the triaging believes the feature would not be desired.

In general, if there is doubt about whether the feature would be desired, the committer doing the triaging should not close it. I can update the guide to make this more clear if you think it would help.

- Reopening Triaged Feature Requests

I roughly agree this, but I think it needs some discussion.

- Why only Ruby developer can reopen it? (maybe the current permission only allows ruby committers to reopen...)

My use of Ruby developer here means anyone who is contributing to Ruby, not just a committer. My intention here is that any Redmine user could reopen a feature request that was closed due to triaging, if they think the feature is valuable and matz (Yukihiro Matsumoto) has not explicitly decided against it. However, I'm not sure if the Redmine permissions allow that. Maybe hsbt (Hiroshi SHIBATA) knows? If not allowed, maybe we can change it so any Ruby developer can request the issue be reopened, and any committer can reopen it if they think the feature may be desired.

- guideline of forking feature requests sounds useful, but I think this is not feature triaging.

Agreed. I just want to make sure people are aware that a decision made during triaging is not final, and can be easily reversed. I can remove the section on submitting a new feature if desired.