Usergrid Contribution Workflow

<table>
<thead>
<tr>
<th>Proposal Name</th>
<th>Usergrid Contribution Workflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Dave Johnson</td>
</tr>
<tr>
<td>Proposed</td>
<td>June 23, 2014</td>
</tr>
<tr>
<td>Revised</td>
<td>August 25, 2015</td>
</tr>
<tr>
<td>Status</td>
<td>DRAFT</td>
</tr>
</tbody>
</table>

This is a plan for how Usergrid contribution works in detail for both an external contributor and a project committer. The Usergrid project has always used GitHub Pull Request mechanism for code reviews and will continue this practice.

Now that Usergrid is at Apache, we've got to follow written ASF policy concerning code contributions. Every contribution must be made under the terms of the Apache Software License (ASL), and for any substantial new contribution the contributor must have an ICLA on file with the ASF, and all releases must be created from the ASF's Git repo – the canonical source for the Usergrid code-base.

We've also got to try to interpret evolving and unwritten ASF policies around using GitHub usage. We would strongly prefer to work in a GitHub repo that is managed by the project's PMC and sync back to the ASF Git repo frequently, but at this point the ASF Infra team is only prepared to offer us a read-only repo at GitHub. This means that we cannot simply press the GitHub merge button to process an accepted Pull Request. Instead, we must follow a set of manual steps linked to below. Hopefully, this will get easier as ASF Infra modernizes and policy is documented.

Overview

Our official GitHub presence will be the apache / usergrid repo on GitHub, which is a read-only mirror of our ASF Git repo. All commits to the master branch will be done as GitHub Pull Requests (PRs) against apache / usergrid, or as patches submitted via JIRA. Existing ASF Infrastructure will assist with echoing of GitHub PR comments to the appropriate Usergrid mailing list(s), and closing PRs

Process for an external contributor

External contributors (i.e. those who are not Usergrid committers) will follow this process for contributing code:

Usergrid External Contributors Guide

Process for a committer accepting external contributions

Usergrid committers will follow this process to accept external contributions (the manual steps I mentioned before):

Usergrid Committers Guide

Process for a committer

Because we are using GitHub as our code review system, all contributions to the master branch must go through code review via the GitHub Pull Request mechanism. If you're a committer, here's what you'll need to do to make a contribution to the master branch:

- File a JIRA issue for the work you will be doing.
- Create a Git branch for the work with a name that references the JIRA issue number.
- Push your new branch and create a Pull Request for it marked as "Do not merge, for review purposes only"
- Do the work for your new feature, commit it and push it.
- Ask a fellow committer to review and accept your PR. Once they are satisfied with the PR...
- Your fellow committer will follow the Usergrid Committers Guide process (linked to above) to merge your contribution back to ASF Git.

Process for a release manager

Once this plan is in effect, here's how a release manager will create an Apache Usergrid release.

- Tag release on ASF Git repo
- Pull from tag on Apache Git repo and create release files
- Follow normal release process (which is currently TBD)