HTRC Software Development Process for Arriving at Technical Decisions

Read about the steps HTRC staff takes for making technical decisions.

How a technical team arrives at technical decisions is an issue that all technical teams grapple with at some point in their life. HTRC has matured to the point where a documented process will help team productivity. In aligning with HTRC principles, this document outlines decision-making and voting policies for HTRC software development. The document draws from the rules observed in Apache foundation software development.

Software development in HTRC is guided by below principles; decisions are limited to the following scope.

HTRC Principles

- HTRC provisions stable and available services to the community; as such stability has to be favored over novelty
- HTRC espouses itself as an open source project
- Everyone is equal. All people with an opinion are entitled to express that opinion and, where appropriate, have it considered by the community
- HTRC long term sustainability depends on community contributions

The scope of decision-making covered by this policy includes:

- Modifications to service architecture and service interactions
- Package releases
- Build/development environment
- New features [1]
- Unit development [2]
- Code maintenance and bug fixes [3]

[1] **New features.** Decision outcomes that emerge from the technical team on new features are advisory only. New features have to be considered in the broader context of community need and resource restrictions.

[2] **Unit development.** Once the need for unit change and action is agreed upon, team agreement is not needed on more detailed implementation decisions. That is implementation proceeds under lazy consensus.

[3] **Code maintenance and bug fixes** are a necessary part of running a production system. Developers should coordinate fixes through the operations manager but code maintenance and bug fixes don't require team agreement on how proceed. That is, once coordinated, fixes proceed under lazy consensus.

Policy

The process of arriving at decisions is in two phases. The first phase engages in consensus building. Where consensus exists, a vote is not needed. In some cases (it is hoped rare), a vote is called. The process is adapted in full from the Apache Software Foundation page https://community.apache.org/committers/decisionMaking.html and https://www.apache.org/foundation/voting.html. The process gradually binds the process towards more formality through steps where the lesser step is ineffective.

Consensus building

Lazy Consensus: Lazy consensus means that you don't need to get explicit approval to proceed, but you need to be prepared to listen if someone objects. Lazy consensus applies to code modifications, bug fixes, and unit modifications only.

Consensus Building: For items larger than code maintenance and bug fixes, make a proposal to the mailing list and discuss the options. There are mechanisms for quickly showing your support or otherwise for a proposal and building consensus amongst the community. The +1, +0, -0, -1 notation (see provided URLs) is used for building consensus.

Once there is a consensus people can proceed with the work under the lazy consensus model. **Voting:** Occasionally a "feel" for consensus is not enough. Sometimes we need to have a measurable consensus.

VOTING

Voting follows the voting system outlined (with a small modification to who votes) in the Apache Software Foundation page on voting: https://www.apache.org/foundation/voting.html. Note the fractional votes and what they are intended to convey. Only HTRC STAFF members have binding votes. This may change as the community becomes more active. Votes should run for at least 72 hours to provide an opportunity for all concerned persons to participate regardless of their geographic locations.

In the event that this policy yields a process or decision that has the potential to negatively affect the future sustainability of HTRC, its staff contentment, or its community, HTRC Executive Management reserves the right to override the decision.