Apache Hadoop Project Bylaws

Table of contents

1 Introduction..... 2
2 Roles and Responsibilities..... 2
3 Decision Making..... 4

Copyright © 2014 The Apache Software Foundation. All rights reserved.

1 Introduction

This document defines the bylaws under which the Apache Hadoop project operates. It defines the roles and responsibilities of the project, who may vote, how voting works, how conflicts are resolved, etc.

Hadoop is a project of the Apache Software Foundation. The foundation holds the trademark on the name "Hadoop" and copyright on Apache code including the code in the Hadoop codebase. The foundation FAQ explains the operation and background of the foundation.

Hadoop is typical of Apache projects in that it operates under a set of principles, known collectively as the "Apache Way". If you are new to Apache development, please refer to the Incubator project for more information on how Apache projects operate.

2 Roles and Responsibilities

Apache projects define a set of roles with associated rights and responsibilities. These roles govern what tasks an individual may perform within the project. The roles are defined in the following sections

- **Users**

  The most important participants in the project are people who use our software. The majority of our developers start out as users and guide their development efforts from the user's perspective.

  Users contribute to the Apache projects by providing feedback to developers in the form of bug reports and feature suggestions. As well, users participate in the Apache community by helping other users on mailing lists and user support forums.
- **Contributors**

  All of the volunteers who are contributing time, code, documentation, or resources to the Hadoop Project. A contributor that makes sustained, welcome contributions to the project may be invited to become a Committer, though the exact timing of such invitations depends on many factors.
- **Committers**

  The project's Committers are responsible for the project's technical management. Committers have access to all subproject subversion repositories. Committers may cast binding votes on any technical discussion regarding any subproject.

  Committer access is by invitation only and must be approved by consensus approval of the active PMC members. A Committer is considered emeritus by their own declaration or by not contributing in any form to the project for over six months. An emeritus committer may request reinstatement of commit access from the PMC. Such reinstatement is subject to consensus approval of active PMC members.

Copyright © 2014 The Apache Software Foundation. All rights reserved.

Significant, pervasive features are often developed in a speculative branch of the repository. The PMC may grant commit rights on the branch to its consistent contributors, while the initiative is active. Branch committers are responsible for shepherding their feature into an active release and do not cast binding votes or vetoes in the project.

All Apache committers are required to have a signed Contributor License Agreement (CLA) on file with the Apache Software Foundation. There is a Committer FAQ which provides more details on the requirements for Committers

A committer who makes a sustained contribution to the project may be invited to become a member of the PMC. The form of contribution is not limited to code. It can also include code review, helping out users on the mailing lists, documentation, testing, etc.

- **Release Manager**

  A Release Manager (RM) is a committer who volunteers to produce a Release Candidate according to HowToRelease. The RM shall publish a Release Plan on the *common-dev@* list stating the branch from which they intend to make a Release Candidate, at least one week before they do so. The RM is responsible for building consensus around the content of the Release Candidate, in order to achieve a successful Product Release vote.

- **Project Management Committee**

  The Project Management Committee (PMC) for Apache Hadoop was created by the Apache Board in January 2008 when Hadoop moved out of Lucene and became a top level project at Apache. The PMC is responsible to the board and the ASF for the management and oversight of the Apache Hadoop codebase. The responsibilities of the PMC include

  - Deciding what is distributed as products of the Apache Hadoop project. In particular all releases must be approved by the PMC
  - Maintaining the project's shared resources, including the codebase repository, mailing lists, websites.
  - Speaking on behalf of the project.
  - Resolving license disputes regarding products of the project
  - Nominating new PMC members and committers
  - Maintaining these bylaws and other guidelines of the project

  Membership of the PMC is by invitation only and must be approved by a consensus approval of active PMC members. A PMC member is considered "emeritus" by their own declaration or by not contributing in any form to the project for over six months. An emeritus member may request reinstatement to the PMC. Such reinstatement is subject to consensus approval of the active PMC members.

Copyright © 2014 The Apache Software Foundation. All rights reserved.

The chair of the PMC is appointed by the ASF board. The chair is an office holder of the Apache Software Foundation (Vice President, Apache Hadoop) and has primary responsibility to the board for the management of the projects within the scope of the Hadoop PMC. The chair reports to the board quarterly on developments within the Hadoop project.

The chair of the PMC is rotated annually. When the chair is rotated or if the current chair of the PMC resigns, the PMC votes to recommend a new chair using Single Transferable Vote (STV) voting. See http://wiki.apache.org/general/BoardVoting for specifics. The decision must be ratified by the Apache board.

3 Decision Making

Within the Hadoop project, different types of decisions require different forms of approval. For example, the previous section describes several decisions which require "consensus approval" approval. This section defines how voting is performed, the types of approvals, and which types of decision require which type of approval.

- **Voting**

  Decisions regarding the project are made by votes on the primary project development mailing list (general@hadoop.apache.org). Where necessary, PMC voting may take place on the private Hadoop PMC mailing list. Votes are clearly indicated by subject line starting with [VOTE]. Votes may contain multiple items for approval and these should be clearly separated. Voting is carried out by replying to the vote mail. Voting may take four flavors

  - **+1** "Yes," "Agree," or "the action should be performed." In general, this vote also indicates a willingness on the behalf of the voter in "making it happen"
  - **+0** This vote indicates a willingness for the action under consideration to go ahead. The voter, however will not be able to help.
  - **-0** This vote indicates that the voter does not, in general, agree with the proposed action but is not concerned enough to prevent the action going ahead.
  - **-1** This is a negative vote. On issues where consensus is required, this vote counts as a **veto**. All vetoes must contain an explanation of why the veto is appropriate. Vetoes with no explanation are void. It may also be appropriate for a -1 vote to include an alternative course of action.

  All participants in the Hadoop project are encouraged to show their agreement with or against a particular action by voting. For technical decisions, only the votes of active committers are binding. Non binding votes are still useful for those with binding votes to understand the perception of an action in the wider Hadoop community. For PMC decisions, only the votes of PMC members are binding.

Copyright © 2014 The Apache Software Foundation. All rights reserved.

Voting can also be applied to changes made to the Hadoop codebase. These typically take the form of a veto (-1) in reply to the commit message sent when the commit is made.

- **Approvals**

  These are the types of approvals that can be sought. Different actions require different types of approvals

  - **Consensus Approval -** Consensus approval requires 3 binding +1 votes and no binding vetoes.
  - **Lazy Consensus -** Lazy consensus requires no -1 votes ('silence gives assent').
  - **Lazy Majority -** A lazy majority vote requires 3 binding +1 votes and more binding +1 votes than -1 votes.
  - **Lazy 2/3 Majority -** Lazy 2/3 majority votes requires at least 3 votes and twice as many +1 votes as -1 votes.

- **Vetoes**

  A valid, binding veto cannot be overruled. If a veto is cast, it must be accompanied by a valid reason explaining the reasons for the veto. The validity of a veto, if challenged, can be confirmed by anyone who has a binding vote. This does not necessarily signify agreement with the veto - merely that the veto is valid.

  If you disagree with a valid veto, you must lobby the person casting the veto to withdraw their veto. If a veto is not withdrawn, any action that has been vetoed must be reversed in a timely manner.

- **Actions**

  This section describes the various actions which are undertaken within the project, the corresponding approval required for that action and those who have binding votes over the action.

  - **Code Change**

    A change made to a codebase of the project and committed by a committer. This includes source code, documentation, website content, etc.

    Consensus approval of active committers, but with a minimum of one +1. The code can be committed after the first +1, unless the code change represents a merge from a branch, in which case three +1s are required.

  - **Product Release**

    When a release of one of the project's products is ready, a vote is required to accept the release as an official release of the project.

    Lazy Majority of active PMC members

  - **Adoption of New Codebase**

Copyright © 2014 The Apache Software Foundation. All rights reserved.

When the codebase for an existing, released product is to be replaced with an alternative codebase. If such a vote fails to gain approval, the existing code base will continue.

This also covers the creation of new sub-projects within the project

Lazy 2/3 majority of PMC members

- **New Branch Committer**

  When a branch committer is proposed for the PMC

  Lazy consensus of active PMC members
- **New Committer**

  When a new committer is proposed for the project

  Consensus approval of active PMC members
- **New PMC Member**

  When a committer is proposed for the PMC

  Consensus approval of active PMC members
- **Branch Committer Removal**

  When removal of commit privileges is sought **or** when the branch is merged to the mainline

  Lazy 2/3 majority of active PMC members
- **Committer Removal**

  When removal of commit privileges is sought. Note: Such actions will also be referred to the ASF board by the PMC chair

  Lazy 2/3 majority of active PMC members (excluding the committer in question if a member of the PMC).
- **PMC Member Removal**

  When removal of a PMC member is sought. Note: Such actions will also be referred to the ASF board by the PMC chair.

  Lazy 2/3 majority of active PMC members (excluding the member in question)
- **Modifying Bylaws**

  Modifying this document.

  Lazy majority of active PMC members

- **Voting Timeframes**

  Votes are open for a period of 7 days to allow all active voters time to consider the vote. Votes relating to code changes are not subject to a strict timetable but should be made as timely as possible.

Copyright © 2014 The Apache Software Foundation. All rights reserved.

- **Product Release - Vote Timeframe**

  Release votes, alone, run for a period of 5 days. All other votes are subject to the above timeframe of 7 days.

Copyright © 2014 The Apache Software Foundation. All rights reserved.