Chem4Word Governance

Roles and Responsibilities

In order to have a smoothly running project, formal roles with corresponding responsibilities are established. A member of the community may have multiple roles.

**Users**

Users are community members who have a need for the project. They are the most important members of the community: without them, the project would have no purpose. Anyone can be a user; there are no specific requirements. Users should be encouraged to participate in the life of the project and the community as much as possible. User contributions enable the project team to ensure that they are satisfying the needs of those users. Common user activities include (but are not limited to):

* advocating for use of the project
* informing developers of project strengths and weaknesses from a new user’s perspective
* providing moral support (a ‘thank you’ goes a long way)
* writing documentation and tutorials
* filing bug reports and feature requests
* participating on the discussion board

Users who continue to engage with the project and its community will often find themselves becoming more and more involved. Such users may then go on to become contributors, as described below.  
  
**How to become one**: Use Chem4Word

**Contributors**

Contributors are community members who submit patches to the project. These patches may be a one-time occurrence or occur over time. Expectations are that contributors will submit patches that are small at first and will only grow larger once the contributor has built confidence in the quality of their patches.  
  
NOTE: before a contributor’s first patch is put into the repository they must sign a Contributor License Agreement or assignment agreement. The patch can be submitted and discussed but it can’t actually be committed to the repository without signed paperwork.  
  
**How to become one** : Submit a patch to the [Chem4Word project](http://chem4word.codeplex.com/).

**Committers**

Committers are contributors who have shown dedication to Chem4Word, high technical prowess and the ability to work well with contributors and users. The committers have responsibilities beyond the contributors. In particular, committers formally decide on whether patches are entered into the main code repository and add those requests. Additionally they verify with Outercurve Foundation staff that a potential contributor has a signed CLA before committing a patch to the repository. A committer will use lazy consensus to decide on whether to commit a patch from a contributor. If the discussion is no longer moving towards a consensus, the PMC must vote via lazy consensus on whether the patch should be applied.  
  
  
**How to become one**: Be a contributor and be nominated to the PMC as a committer. Nominations should be sent to [The Project Leader](mailto:clyde.davies@gmail.com) . You may nominate yourself.

Project Management Committee (PMC)

The project management committee has responsibilities beyond committers that include participating in strategic planning, release planning, and approving changes to the governance model. It also makes decisions when community consensus cannot be reached.  
  
The PMC has final say over who can become a committer and will use lazy consensus for approval. Discussion over committer nominations will be done in private.  
  
Membership of the PMC is by invitation from the existing PMC members. A nomination will result in discussion and then a vote by the existing PMC members. PMC membership votes are subject to consensus approval of the current PMC members and will be done in private.  
  
**How to become one**: Be invited by a current PMC member and have nomination approved by the PMC.

**PMC Chairperson**

The PMC Chairperson is a member of the PMC whom the Outercurve Foundation staff consider the primary point of contact or first point of contact for the project for purposes of business operations including domain registrations, and technical services (e.g. code-signing). The Chairperson does not have any increased authority and is simply a convenient point of contact for the Foundation.  
  
**How to become one**: Have nomination approved by members of the PMC.

Lazy Consensus

Lazy consensus is a very important concept within the project. It is this process that allows a large group of people to efficiently reach consensus, as someone with no objections to a proposal need not spend time stating their position, and others need not spend time reading such mails.  
  
For lazy consensus to be effective, it is necessary to allow at least 72 hours before assuming that there are no objections to the proposal. This requirement ensures that everyone is given enough time to read, digest and respond to the proposal. This time period is chosen so as to be as inclusive as possible of all participants, regardless of their location and time commitments.

**Voting**

Not all decisions can be made using lazy consensus. Issues such as those affecting the strategic direction or legal standing of the project must gain explicit approval in the form of a vote. Every member of the community is encouraged to express their opinions in all discussion and all votes.

**Transparency**

Building community trust in the governance of an open-source project is vital to its success. To that end, decision making must be done in a transparent, open fashion. No decisions about the project’s direction, bug fixes or features may be done without community involvement and participation. Discussions must begin at the earliest possible point on a topic; the community’s participation is vital during the entire decision-making process.