

Community Sports Management System Project Constitution

Ross D. Gardler, ross@saafe.org

1 Project Constitution

Note:

Inspired by the Krysalis Community constitution, which in turn is inspired by the xml.apache.org constitution. (Apache Software Foundation - all rights reserved)

This document defines the guidelines of the Community Sports Management Systems project (CSMS). It includes definitions of the various categories of membership, who is able to vote, how conflicts are resolved by voting, and the procedures to follow for proposing and making changes to the codebase of the Project.

This is a living document. Changes can be made by the Project Management Committee and must be approved by 2/3 of all Developers.

This document consists of the following sections:

- Our Mission
- Roles and Responsibilities: defines the recognized roles in the project
- Decision Making: defines how action items are proposed and voted on
- Communication: defines how users and developers communicate
- Source Repositories: define how the Project's source code is organized and developed
- Project Management: defines the roles and responsibilities of the Project Management Committee (PMC)
- Project Management Committee (PMC) Bylaws

2 Our Mission

Our mission is to build a community and software system to support the growth and development of sports clubs of all types. We aim to bring these valuable real world communities together in the virtual world of the Internet. The objective is to encourage more participation in sports clubs of all types by all people whether they be players, organisers or

supporters.

3 Roles and Responsibilities

The roles and responsibilities that people can assume in the project are based on merit. Everybody can help no matter what their role. Those who have been long term or valuable contributors to the project obtain the right to vote and commit directly to the document and source code repository.

There are five roles within the community, each role brings with it a set of responsibilities. These roles are not mutually exclusive. The roles are:

1. Members
2. Content Managers
3. Developers
4. Committers
5. Project Management Committee members

3.1 Members

Anyone can be a member.

Members are the people who participate in the community. They use the products of the Project. People in this role aren't contributing code or documents to the repository, however, they are contributing to the community by providing valuable feedback and requests for enhancements, as well as making the community stronger through their support.

This is by far the most important category of people as, without members, there is no reason for the project or for the community.

3.2 Content Managers

Content Managers also have the responsibilities of Members.

Members who contribute to the data, information and news available to the community on a frequent basis can be proposed as a Content Manager, or they may request Content Manager status. Any member can propose any other member as a Content Manager. Once proposed a vote of the existing Content Managers is carried out (see below for voting procedures). A result of at least three positive votes and no negative votes will result in the creation of a new Content Manager.

Content Managers are able to edit materials posted by other members (normally only the poster can edit their own materials). They also have the ability to create new sections within the community to support new sports, teams and interests. As such, the Content Managers are

Community Sports Management System Project Constitution

answerable to the Members and must endeavour to ensure the community facilities available are what the members require,

At times, Content Managers may go inactive for a variety of reasons. A Content Manager that has been inactive for 4 months or more may lose his or her status as a Content Manager. However, they will always be contacted via their last known email address before Content Manager status is removed.

3.3 Developers

Anyone can be a developer.

Developers are the people contribute to the project by writing code and documentation patches or contribute positively to the project products in other ways such as through open discussion on our developer lists. A developer's contribution is always recognized. In source code, all developers who contribute to a source file may add their name to the list of authors for that file. In documentation, they may similarly add their name to the authors list for any document they contribute to.

These credits will be reproduced in all source code distributions of the software and in all documentation.

3.4 Committers

Developers who give frequent and valuable contributions to a subproject of the CSMS project can have their status promoted to that of a "Committer" for that subproject.

A Committer has write access to the source code repository and gains voting rights allowing them to affect the future of the subproject. In order for a Developer to become a Committer, another Committer can nominate that Developer or the Developer can ask for it. Once a Developer is nominated, all of the Committers for a subproject will vote. If there are at least 3 positive votes and no negative votes, the Developer is converted into a Committer and given write access to the source code repository for that subproject.

At times, Committers may go inactive for a variety of reasons. A Committer that has been inactive for 6 months or more may lose his or her status as a Committer. However, they will always be contacted via their last known email address before commiter status is removed.

Committers have the right to add their details to the license under which all proucts of the project are released.

3.5 Project Management Committee (PMC) Member

Committers who frequently participate with valuable contributions may have their status

promoted to that of a "Project Management Committee Member".

This committee is the official managing body of the CSMS Project and is responsible for setting overall project direction. On Sourceforge (sourceforge.net) they are called admins. In order to become a Member of the PMC, someone on the PMC must nominate the Committer. The individual may then be approved with a 3/4 majority of the PMC.

The PMC has the right to secure the health of the community, and has the power to exclude any developer, content manager or member from the community as an extreme measure if he/she endangers the mission of the Project. All exclusions must be passed with a 3/4 majority vote. No vetoes are permmitted on such votes.

4 Decision Making

All Members, Developers and Content Managers are encouraged to participate in decisions, but the decision itself is made by those that have Committer status in the Project. In other words, the Project is a "Minimum Threshold Meritocracy".

Any person, whether a member or not, may vote on any issue or action item. However, the only binding votes are those cast by a Committer. If the vote is about a change to the source code or documentation and the primary author is a Developer and not a Committer, the primary author of what is being changed may also cast a binding vote on that issue.

The act of voting carries certain obligations. Voting members are not only stating their opinion, they are also agreeing to help do the work.

Each vote can be made in one of three flavors:

+1

"Yes," "Agree," or "the action should be performed." On some issues this is only binding if the voter has tested the action on their own system(s).

+/-0

"Abstain," "no opinion". If, as a commiter, you are unable to commit to helping bring the results of the vote to fruition you should abstain rather than cast a +1. The use of +0 can indicate your preference in the event of a tied vote. In this instance a +0 vote is taken as meaning "+1 but I can't help". A 0 or -0 vote is treated as "no opinion".

-1

"No." On issues where consensus is required, this vote counts as a veto. All vetos must contain an explanation of why the veto is appropriate. Vetos with no explanation are void. No veto can be overruled. If you disagree with the veto, you should lobby the person who cast the veto. Voters intending to veto an action item should make their opinions known to the group immediately so that the

Community Sports Management System Project Constitution

problem can be remedied as early as possible.

An action requiring consensus approval must receive at least 3 binding +1 votes and no binding vetos.

An action requiring majority approval must receive at least 3 binding +1 votes and more +1 votes than -1 votes.

All other action items are considered to have lazy approval until somebody votes -1, after which point they are decided by either consensus or majority vote, depending on the type of action item.

5 Action Items

All decisions revolve around "Action Items." Action Items consist of the following:

- Long Term Plans
- Short Term Plans
- Release Plan
- Release Testing
- Showstoppers
- Product Changes
- Appointment of a new Content Manager
- Appointment of a new Committer
- Appointment of a new PMC member
- Removal of Content Manager Status
- Removal of Committer Status
- Removal of PMC Membership
- Exclusion of a member

5.1 Long Term Plans

Long term plans are simply announcements that group members are working on particular issues related to the Project. These are not voted on, but Developers who do not agree with a particular plan, or think that an alternative plan would be better, are obligated to inform the group of their feelings by applying (and explaining) a veto.

5.2 Short Term Plans

Short term plans are announcements that a developer is working on a particular set of documentation or code files with the implication that other developers should avoid them or try to coordinate their changes. These are not voted on, but Developers who do not agree with a particular plan, or think that an alternative plan would be better, are obligated to inform the group of their feelings by applying (and explaining) a veto.

5.3 Release Plan

A release plan is used to keep all Developers aware of when a release is desired, who will be the release manager, when the repository will be frozen to create a release, and other assorted information to keep Developers from tripping over each other. Lazy majority decides each issue in a release plan.

5.4 Release Testing

After a new release is built, it must be tested before being released to the public. Majority approval is required before the release can be made.

5.5 Showstoppers

Showstoppers are issues that require a fix be in place before the next public release. They are listed in the status file in order to focus special attention on these problems. An issue becomes a showstopper when it is listed as such in the status file and remains so by lazy consensus.

5.6 Product Changes

Changes to the products of the Project, including code and documentation, will appear as action items in the status file. All product changes to the currently active repository are subject to lazy consensus.

5.7 Appointment of a new Content Manager

Members who contribute to the data, information and news available to the community on a frequent basis can be proposed as a Content Manager, or they may request Content Manager status. Any member can propose any other member as a Content Manager. Once proposed a vote of the existing Content Managers is carried out (see below for voting procedures). A result of at least three positive votes and no negative votes will result in the creation of a new Content Manager.

5.8 Appointment of a new Committer

A Committer can nominate any Developer as a new Committer or a Developer may ask for a vote on their acceptance as a Committer. Once a Developer is nominated, all of the Committers for a subproject will vote. If there are at least 3 positive votes and no negative votes, the Developer is converted into a Committer and given write access to the source code repository for that subproject.

5.9 Removal of Content Manager Status

If a Content Manager has been inactive for 4 months or more any other Content Manager can request the removal of Content Manager status from that person. No vote is taken until an attempt to contact the Content Manager at their last known email address and a further 4 weeks have past. If the Content Manager responds, agreeing that they are no longer active then no vote is required. However, if no response is recieved, or if a request is to retain their status a vote is taken. A result of 3 or more positive votes and no negative votes will result in the removal of Content Manager status.

5.10 Removal of Commiter Status

If a Committer has been inactive for 6 months or more any other Committer can request the removal of Committer status from that person. No vote is taken until an attempt to contact the Committer at their last known email address and a further 4 weeks have past. If the Committer responds, agreeing that they are no longer active then no vote is required. However, if no response is recieved, or if a request is to retain their status a vote is taken. A result of 3 or more positive votes and no negative votes will result in the removal of Committer status.

Removal of Committer status does not mean that credits in source code and license files is removed. These credits cannot be removed under any circumstances.

5.11 Appointment of a new PMC member

The existing PMC can nominate any Committer as a new PMC Member. New members are accepted on a 3/4 majority. There are no veto cotes in this cote.

5.12 Removal of PMC Membership

Any member of the community has the right to request a member of the PMC be removed. Such a vote will go to all members and no veto is allowed. A majority of 3/4 is required to remove a PMC member.

5.13 Exclusion of a Member

The PMC has the right to secure the health of the community, and has the power to exclude any developer, content manager or member from the community as an extreme measure if he/she endangers the mission of the Project. All exclusions must be passed with a 3/4 majority vote. No vetoes are permmitted on such votes.

6 Communication

The project obtains its strength from the communication of its members. In order for members to easily communicate with each other, the project has a variety of mailing lists. These lists, with the exception of the announcement lists, are not moderated and anybody is more than welcome to join them. However, you must be subscribed to post to a list.

To reduce the bandwidth demands on everyone, mail should not contain attachments and be in plain text. It is recommended that you place interesting material either within the body of the message or provide a URL for retrieval. If you do not have public space available for sharing of large data files please post a request for assistance on the relevant mailing list. Someone in the community will assist.

The Project's lists fall into the following categories:

6.1 Announcement Lists

Announcement lists are very low traffic designed to communicate important information, such as final releases of a subproject's code, to a wide audience.

6.2 User Lists

User lists are for users of a product to converse about such things as configuration and operating of the products of the project.

6.3 Developer Lists

Developer lists are for the developers of the project. On these lists suggestions and comments for code changes are discussed and action items are raised and voted on. For the developer community, these lists are the very center of the project where all the "action" is.

6.4 Commit Lists

The commit lists are where all cvs code commit messages are sent. All committers are required to subscribe to this list so that they can stay aware of changes to the repository.

7 Source Repositories

The project's codebase is maintained in shared information repositories using CVS on sourceforge.net. Only Committers have write access to these repositories. Everyone has read access via anonymous CVS.

7.1 Coding Conventions

Java Language source code in the repository is recommended to be written in conformance to the Code Conventions for the Java Programming Language as published by Sun (<http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html>).

7.2 License

All source code committed to the Project's repositories must be covered by the most recent version of The Mozilla Public License at the time of the contribution. Binary files can, as an alternative, contain a copyright and license that allows redistribution under the same conditions as the Mozilla Public License.

7.3 Status Files

Each of the project's active source code repositories contain a file named status.xml which is used to keep track of the agenda and plans for work within that repository. The status file includes information about release plans, a summary of code changes committed since the last release,

7.4 Branches

Groups are allowed to create a branch for release cycles, etc. They are expected to merge completely back with the main branch as soon as their release cycle is complete.

7.5 Changes

Simple patches to fix bugs can be committed then reviewed. With a commit-then-review process, the Committer is trusted to have a high degree of confidence in the change.

Doubtful changes, new features, and large scale overhauls need to be discussed before committing them into the repository. Any change that affects the semantics of an existing API function, the size of the program, configuration data formats, or other major areas must receive consensus approval before being committed.

Related changes should be committed as a group, or very closely together. Half complete projects should never be committed to the main branch of a development repository. All code changes must be successfully compiled on the developer's platform before being committed.

The current source code tree for a subproject should be capable of complete compilation at all times. However, it is sometimes impossible for a developer on one platform to avoid breaking some other platform when a change is committed. If it is anticipated that a given

change will break the build on some other platform, the committer must indicate that in the commit message.

A committed change must be reversed if it is vetoed by one of the voting members and the veto conditions cannot be immediately satisfied by the equivalent of a "bug fix" commit. The veto must be rescinded before the change can be included in any public release.

7.6 Patches

When a specific change to a product is proposed for discussion or voting on the appropriate development mailing list, it should be presented in the form of input to the patch command. The patch must be inserted in the patch tracking system for consideration by the committers.

The patch should be created by using the `diff -u` command from the original software file(s) to the modified software file(s). For example:

```
diff -u Main.java.orig Main.java >> patchfile.txt
```

or

```
cvs diff -u Main.java >> patchfile.txt
```

All patches necessary to address an action item should be concatenated within a single patch message. If later modification to the patch proves necessary, the entire new patch should be posted and not just the difference between the two patches.

8 Project Management Committee (PMC) Bylaws

The Project Management Committee (PMC) was formed by the CSMS founders in October 2002. This Committee consists of 3 founding members, one of whom is the founding Chairman.

The term of the Chairman is one year. There is no term limit for members.

8.1 Roles

The PMC is responsible for the strategic direction and success of the CSMS project. This governing body is expected to ensure the project's welfare and guide its overall direction. The PMC may not necessarily participate in the day-to-day coding but is involved in the overall development plans, the alleviation of any bottlenecks, the resolution of conflicts, and the overall technical success of the project.

8.2 Meetings

Community Sports Management System Project Constitution

The PMC do not meet regularly. However, they are expected to be in close communication on a regular basis.

Formal meetings may be called by any PMC member and all members are expected to make all reasonable efforts to attend. These formal meetings are to discuss issues, determine strategic direction, and forward progress. These meetings may take place online, via teleconference, or via other means deemed effective by the PMC.

The PMC has an annual meeting at which time a new Chairman is elected. The old Chairman maintains membership status with no extra privileges.

8.3 Membership

PMC members may resign at any time. The Chairman may resign as Chairman at any time without resigning membership to the PMC.