World Meteorological Organization
COMMISSION FOR CLIMATOLOGY
Expert Meeting on WMO Climate Data Management System Strategy
Geneva, Switzerland, 12-14.XII.2018

EM-WCDMSS/Doc N(3.1)
Submitted by:
Steve Foreman
14.XII.2018

MANAGEMENT AND OWNERSHIP OF THE REFERENCE CDMS TOOL SET

This document will become document 6.1(3) of the 2019 workshop.

REVIEW COMMENTS

Version / No	Reviewer	Issue	Notes	Response
0.1/1	ET-DRM	OPenWIS Association as the owner of OpenCDMS		Updated para 2.3 with a reference to the correct rules governing OpenWIS Association

DISCLAIMER

Regulation 43

Recommendations of working groups shall have no status within the Organization until they have been approved by the responsible constituent body. In the case of joint working groups the recommendations must be concurred with by the presidents of the constituent bodies concerned before being submitted to the designated constituent body.

Regulation 44

In the case of a recommendation made by a working group between sessions of the responsible constituent body, either in a session of a working group or by correspondence, the president of the body may, as an exceptional measure, approve the recommendation on behalf of the constituent body when the matter is, in his opinion, urgent, and does not appear to imply new obligations for Members. He may then submit this recommendation for adoption by the Executive Council or to the President of the Organization for action in accordance with Regulation 9(5).

Chairperson, Publications Board
World Meteorological Organization (WMO)
7 bis, avenue de la Paix
P.O. Box No. 2300
CH-1211 Geneva 2, Switzerland

Tel.: +41 (0)22 730 84 03
Fax: +41 (0)22 730 80 40
E-mail: Publications@wmo.int

NOTE:
The designations employed in WMO publications and the presentation of material in this publication do not imply the expression of any opinion whatsoever on the part of WMO concerning the legal status of any country, territory, city or area, or of its authorities, or concerning the delimitation of its frontiers or boundaries.

The mention of specific companies or products does not imply that they are endorsed or recommended by WMO in preference to others of a similar nature which are not mentioned or advertised.

This document (or report) is not an official publication of WMO and has not been subjected to its standard editorial procedures. The views expressed herein do not necessarily reflect those of WMO or its Members.

MANAGEMENT AND OWNERSHIP OF THE REFERENCE CDMS TOOL SET

1 PURPOSE OF THIS DOCUMENT

1.1 The reference Climate Data Management (CDMS) tool set will be developed using open source principles. The purpose of the current document is to outline the governance regime that will form the starting point for developing the tool set. As the development project proceeds the governance procedures will be adapted by the project.

2 OWNERSHIP OF THE REFERENCE CDMS TOOL SET

2.1 The reference CDMS tool set needs to have a legal owner that:

2.1.1 Is the formal owner of the software for the purposes of copyright and trademarks;

2.1.2 Issues the licence for the software (this licence is likely to be the GPL 3 open source licence);

2.1.3 Enters into agreements with contributors to the software development that grants the development project a "*... perpetual, world-wide, non-exclusive, no-charge, royalty-free, irrevocable copyright licence to reproduce, prepare derivative works of, publicly perform, sublicense, and distribute [the] contributions and such derivative works*" (Fogel, 2018).

2.2. Some open source projects choose to use an umbrella organization (such as the Apache Foundation) as the owner of their software, others set up their own legal entity (such as the OpenWIS Association), and others use some other organization (for example, MySQL was owned by Oracle Corporation at the time this document was written).

2.3. Two candidates for ownership of the reference CDMS tool set are WMO and the OpenWIS Association (AISBL). The OpenWIS Association (AISBL) was registered in Belgium to mitigate risks to open source projects associated with software patents. The rules governing the OpenWIS Association (AISBL) are available at https://openwis.github.io/openwis-documentation/. In essence, the rules (at the time of writing) would require the OpenCDMS to be nominated by a member of the OpenWIS Addociation and for the project to demonstrate that it had sufficient resources to commit to be likely to succeed, that it would need to adhere to specified development standards. If OpenCDMS were accepted as an OpenWIS Association project, the OpenWIS Association Technical Committee would formally set up a Project Management Committee and appoint a suitably qualified Project Leader to chair that committee.

3 GOVERNANCE TOOLS USED BY THE OPEN SOURCE PROJECT

3.1 STAKEHOLDERS FOR GOVERNANCE

3.1.1 Although there are some notable exceptions, typical open source projects seek to reach decisions by consensus. This is because open source developers contribute to the project because they want to, and a decision making process that makes them feel excluded from the key decisions can result in them preferring to contribute to other projects instead. A formal decision making process is needed for those situations when reaching a consensus may take too long or when there is no clear best way forward. The formal decision making process has to involve a balanced set of representatives of the stakeholders who have an interest in the successful delivery of the project. In addition to the discussion fora used within the project, it

may be necessary to mirror some discussions on fora used by stakeholders (such as WMO Communities of Practice).

3.1.2 The key stakeholder groups for the open source project for the reference CDMS tool set are:

3.1.2.1 **Compliance**. This group of stakeholders is interested in the reference CDMS tool set assisting its users to comply with technical regulations, standards and recommended practices. Typical members of the group would be members of climate related expert teams. Their contributions to the project would be in advising the developers on how to interpret the regulations, standards and practices, and in auditing the resulting software system to confirm that it achieves what is expected in terms of compliance. This group will also need to make sure that decisions on algorithms and practices agreed by the project are proposed for inclusion the regulatory and guidance materials.

3.1.2.2 **Climate data managers**. Climate data managers will form the majority of users of the tool set. Their contribution to the project will be in terms of making sure the system will support the broader processes of climate data management and, probably, in writing user documentation for the project. They will also be one of the main sources of bug reports for the software.

3.1.2.3 **CDMS installers**. Experts from countries with well-developed climate data management practices often assist those in other countries that have less mature processes to implement software CDMS and processes around them. This group of people will have a good understanding of the challenges of introducing the tool set into organizations. Their feedback to the project will be useful in removing errors, improving the installation process, and improving the relevance of the system to climate data managers.

3.1.2.4 **Climate scientists**. In addition to their role as users of the data produced by the tool set, climate scientists will need to advise the project on details of the algorithms it uses (for example, for quality control). This advice will allow the project to proceed faster than if the full WMO regulatory procedures were to be used, while ensuring that the decisions of the project were guided by science rather than computational considerations.

3.1.2.5 **Software developers**. Software developers will make the majority of visible contributions to the project and are likely to be the most numerous of all the stakeholder groups.

3.1.2.6 **Other contributors**. Although software coders are the most obvious contributors to the project, OpenCDMS will need contributions from people with a wide variety of skills (such as software testing, integration testing, documentation, deployment, standards development), and from organizations that are willing to support OpenCDMS financially, by assigning staff to work on the project, or by making computing or other facilities available to the project.

3.1.2.7 **Potential users**. Potential users should be encouraged to participate in the project and provide feedback on whether the system will meet their needs. They should have access to a test version of the tool set so that they can provide specific feedback. Their feedback would be considered to be a contribution.

3.2 GOVERNANCE TOOLS

3.2.1 The main tools available to the project for governance are:

3.2.1.1 **Management team**. A modest number of people who contribute in a major way to the project form the management team for the project. The team is responsible for coordinating the work of the project and deciding when to invoke formal decision making procedures. Members of the management team elect a chair person from among the members of the management team. Selection of the management team is by vote among voting community members.

3.2.1.2 **Consensus**. Open source development projects work by constant communication between community members using communications tools that make conversations open to all to see and contribute. Community members can not only make their views known, but can also see the views of all other community members. A key feature of open source projects is that this openness usually results in a "meritocracy", in which the views of those community members who make the most relevant contributions to the project are most influential. In the majority of cases consensus is achieved naturally.

3.2.1.3. **Committing**. Committing is the action of including an item within a release of OpenCDMS. That item could be software code, documentation, algorithms or any other item that impacts on users. Only a limited number of community members have the authority to commit items, all though all community members may propose items. The authority to commit is granted by the management team. [Confirm the management team is the correct granter]

3.2.1.4 **Polling**. Sometimes it is necessary to choose between competing options. In this case, reaching consensus may be speeded up by surveying the contributors for their preference among the options and (always) allowing alternatives to be proposed. Fogel (2018) reports that proposals made during the polls can lead to unexpected resolutions to the issue under discussion. In addition to prompting for new ideas, the poll results in statistics that can be used to focus discussions leading to consensus. [**Question: should polling be confined to those entitled to vote, or should it be open to all?**]

3.2.1.5 **Voting**. Not all decisions can be made by consensus and the project may need to use a formal vote to reach a decision. Except where the vote concerns individuals (such as a vote to grant or remove voting privileges from an individual) the discussions and voting should be open – not only the overall outcome, but the individual votes. When a secret vote is held, not only are the votes secret during and after the vote, but also the fact of the vote is only disclosed to those entitled to participate in the vote (Fogel (2018)). Fogel (2018) also reports that such openness avoids those without voting rights feeling left out. "Approval voting" allows voters to vote for all options with which they agree; the results are simple to analyse and the approach is easy to explain. A significant output from the first stage of the project has to be the process for deciding who should be a voting contributor.

3.2.1.6 **Veto**. Some decisions may have a large adverse impact on the project that is not obvious to many of those participating in the discussions or a vote. Although such cases should be rare, the veto provides a means of triggering a detailed examination of the objection raised through the veto. A key decision of the first stage of the project is determining which contributors are to be given the right to raise a veto. A veto is followed by an intense period of discussion focussed on the reason for the veto, resulting in withdrawal of the veto, revised consensus or revised vote.

3.2.2 Although open to contributions from anyone, open source projects are not often wholly democratic and limit voting rights to individuals who have demonstrated both an allegiance to the project and that their judgement on the project is sound. Fogel (2018) reports that open source projects typically invite community members who have "committed" a large number of reliable code changes, or who have in other ways demonstrated the value of their contributions, to become voters. Existing voters decide, by secret vote, on whether a proposed

new voter is accepted. In the case of the reference CDMS tool set, submission of code is not a reliable measure of the value of an individual's contribution to the project, and so the voting community has to include individuals from each of the stakeholder groups. The initial voting community will be decided as part of the initiation of the project.

3.2.3 In the event that a voter is found not to contribute or to act against the interests of the project their voting rights may be removed following a vote by all other voters.

3.3 DEVELOPMENT ASPECTS OF THE REFERENCE CDMS TOOL SET TO BE GOVERNED

3.2.1 Although in theory open source projects could be self-governing, making decisions on both the governance and the development as the project evolved, successful open source projects consider their governance structure before problems arise (Fogel, 2018). The governance has to cover a number of design and implementation issues including what functionality has to be available in each release of the software, its architecture, overall design of the software, detailed changes to the software code, and the algorithms to be used (for example, for quality control):

3.2.1.1 **Contents of a release**. A release of the tool set will contain a mixture of changes that implement core capability that is necessary for conformance with the CDMS specifications (WMO-No. 1131) and changes that add values for some or all users or address known bugs. The plan for a release has to specify the functionality that has to be present when the release is published. Although in practice the contents may be agreed by consensus, holding a formal vote on the minimum content will give authority to efforts to ensure that the critical functionality is actually delivered. Having this list will also allow organizations contributing resources to the development to focus their resources on what is most needed.

3.2.1.2 **Architecture**. The architecture of the tool set determines how it fits together and how it interacts with the rest of the world. It is essential that the architecture is widely accepted, otherwise the software will become inconsistent, unreliable and difficult to maintain. There will be a temptation to keep adapting the architecture, that if left unrestrained would result in community members trying to chase a moving target. So, although the architecture should be developed by consensus, it should be agreed formally through a vote. Any changes to the architecture that are proposed as the project develops should use the same process.

3.2.1.3 **Overall design of the software**. The architecture of the tool set defines the building blocks of the tool set and the points at which those building blocks interact. Each of those blocks, and the interfaces between them and the outside world, need to be designed. The way these blocks are split into modules and the detailed specification of the interfaces are key constraints on the development of the software. These should be developed by consensus using polls as a way to achieve that. Votes should be used only rarely, as the need for a vote indicates that some of the contributors feel that the proposed design will not let the tool set achieve what they think it should achieve. As the project progresses, it is highly probable that some aspects of the design and modules that have already been implemented will need significant changes; the process for managing these should be the same as for the original design.

3.2.1.4 **Detailed changes to the software code**. Developing the detailed computer programs to make the tool set work lies in the field of expertise of the community members. The open source approach of people submitting proposed code for others to view and test usually provides a sound review process. In some cases, it may be necessary for someone (in a successful project this is often the community member proposing the contribution) to ask explicitly for a contribution to be reviewed. Although the discussion forums provide a formal record of the review process, there may be

aspects of the tool set that justify a more compact summary of the review (such as confirmation that the science algorithms have been translated correctly into software).

3.2.1.5 **The algorithms to be used**. Although the WMO technical regulations, standards, specifications and recommended practice give many details on procedures and algorithms to be used in climate data management, development of the software will uncover many areas in which detailed algorithms have to be implemented that are not in these references. In these cases, the algorithms used are likely to become *de facto* standards, so a formal review by climate scientists is appropriate. Agreement on the algorithm to use may be reached by consensus (perhaps guided by polls), or it may have to resort to a vote. In the case of a decision on an algorithm it may be appropriate to limit the vote to voting contributors from the compliance installation, user and climate science communities. Each algorithm agreed within the project should be submitted to the appropriate WMO technical commission for possible inclusion as a standard, specification or recommended practice. Just as for the software itself, the algorithms should be reviewed, documented and version controlled.

3.3 IMPLEMENTING THE GOVERNANCE REGIME

3.3.1 The initial governance regime will be set up by the team that initiates the open source project and may need to be modified in order to smooth the work of the project. Once the project is running satisfactorily and additional voters have been selected from among the contributors, maintenance of the governance regime will be by vote among the voting contributors. Governance procedures from other open source projects should be used as the starting point for developing the policies for the reference CDMS tool set.

3.4 MANAGING FINANCIAL SUPPORT TO THE PROJECT

3.4.1 Although funding is needed to support the project, it is not anticipated that the project itself will receive or manage financial contributions. All contributions to the project will be "in kind" through the provision of community member time (even if that time is paid for by a donor) or of infrastructure to support the project activities. Should WMO hold a trust fund or special account to support the project, the decision on how those funds should be spent would be made through WMO decision making procedures that took account of requests and recommendations from the project.

3.4.2 Decisions within the project are made by the contributors to the project. By assisting participation by contributors a donor is able to propose contributions that further the aims of the donor. This may be to ensure that the critical functionality of a release is provided or it may be to provide additional functionality. The contributions would, like all other contributions, only be accepted if they passed the review process.

4 REFERENCES

Fogel, K, 2018: Producing Open Source Software. https://producingoss.com/ (referenced 8 November 2018)