Distr.
GENERAL

Working Paper No.6
10 May 2009

ENGLISH ONLY

**UNITED NATIONS STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE
CONFERENCE OF EUROPEAN STATISTICIANS**

**EUROPEAN COMMISSION
STATISTICAL OFFICE OF THE
EUROPEAN COMMUNITIES (EUROSTAT)**

**ORGANIZATION FOR ECONOMIC COOPERATION
AND DEVELOPMENT (OECD)
STATISTICS DIRECTORATE**

Meeting on the Management of Statistical Information Systems (MSIS 2009)
(Oslo, Norway, 18-20 May 2009)

Topic (i): Sharing of software and components

COOPERATION IN DEVELOPMENT OF OPEN SOURCE SOFTWARE

Invited Paper

Prepared by Harri Lehtinen, Statistics Finland and Rune Gløersen, Statistics Norway

I. INTRODUCTION

1. Statistics Sweden, being the vendor and major stockholder of the PC-Axis software suite, investigates different possibilities for reducing their cost and responsibilities caused by the increased usage and expectations from the wide community of PC-Axis users. In recent years, parts of the software development and maintenance have been carried out by the other Scandinavian NSI's, resulting in the inclusion of different new modules in the software suite. All products included in the PC-Axis software suite are provided under one commercial license set out by Statistics Sweden to cover the cost of development, maintenance and marketing of the suite.

2. As one alternative model for future cooperation in the development of the PC-Axis software, a model for cooperation based upon the principles from development of Open Source Software (OSS) was outlined, from which we have derived a general model presented in this paper. Even though the original paper was targeted for PC-Axis, we think the model should be applicable for any cooperation in software development, with the objective to provide the resulting software as Open Source. The objective of this paper is to serve as an input to the general discussion on how to organise OSS development, while keeping sufficient control on resources spent and the quality of the products.

3. Open source development spans from what could be regarded as somewhat idealistic, to being a platform leaving the business of adding value to OSS to become highly commercial and lucrative. The objectives for development of software for statistical production and analysis made by statistical organizations can be placed somewhere in between these two. We are not interested in gaining business out of the products developed, but we have limited resources to put into the development, and we need to ensure that our resources

are spent according to our defined objectives and strategies and that the products fulfil our needs. This means that we need to govern the OSS developments, and that we need to take responsibility for the quality.

4. This paper outlines an Open Source Community Model (OSCM) to support these responsibilities in the development of OSS for statistical use.

II. AN OPEN SOURCE COMMUNITY MODEL

5. There seems to be an increased interest in common software development between statistical organizations, at least within certain areas. We experience that off-the-shelf software do not always fulfil our needs, and that we all spend extensive resources in development of software with similar or overlapping functionality. We also meet regularly to exchange experiences and to demonstrate our latest developments. Our experience is that implementation of software developed by others is often complicated and costly. Cooperation in the development of OSS should encourage developers from different statistical organizations, and from outside, to contribute to the development of reusable, modular, functional and high quality software.

6. The reason for establishing an OSCM is to keep control of the cooperation according to the objectives defined among the funding partners, and to ensure that progress is made at an acceptable and controllable cost, whilst keeping up with the best achievable quality in development and maintenance of the software.

7. An OSCM must be based on a well defined organizational model, anchored to a responsible ownership which needs to be well defined and accepted. The OSCM creates opportunities to reap the benefits of fruitful contributions to the development of a software suite, but resources must be allocated to ensure that the contributions are of sufficient quality and within the objectives of the software usage.

8. Furthermore, the OSCM must ensure that the usage of the code and contributions to the software are held within the principles set out by the OSS licensing rules chosen by the community for the specific suite or components. Legal aspects and copyright issues must be considered and dealt with.

9. OSS enables possibilities for commercial companies to perform business based on usage of the software. This typically comprises support services, integration with other software or integration into the users own environment, training and other user related initiatives. Commercial companies can also have an interest in adding functionality or software components to the software suite.

III. WHAT IS OPEN SOURCE?

10. Open source software (OSS) is defined as computer software for which the source code and certain other rights normally reserved for copyright holders are provided under a software license that meets the Open Source Definition[1]

11. The Open Source Definition is used by the Open Source Initiative to determine whether or not a certain computer software considered open source. There are certain criteria which have to be met before software can be considered open source. These criteria include the following ten points:

1. Free Redistribution
2. Source Code
3. Derived Works
4. Integrity of The Author's Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License

[1] http://en.wikipedia.org/wiki/Open_source_software

8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must Be Technology-Neutral

IV. OPEN SOURCE LICENSES

A. Recommended types of licences

12. There is a large variety of OpenSource licenses to choose from. They generally fall into two main categories:

- Licenses that allow use of source code for derived products, with few preconditions. Also called a permissive license. (BSD, MIT-X11, Apache License)
- Licenses that allow use of source code for derived products, but require the derived product to be licensed under the same terms as the original source code, referred to as a Copyleft license. (GPL, EUPL, LGPL, MPL).

There are of course differences within these two groups of licenses as well, mostly in relation to attribution clauses.

13. We recommend that OSS software developed under the conditions of an OSCM set up by statistical organizations should be licensed under a strong Copyleft license. This includes the basic software and all related components. This will require any derived work to be issued under the same license terms as the original software. For this particular purpose, we recommend two licenses:

14. *GPL (The GNU General Public License).*
The GPL is a widely used free software license, originally written by Richard Stallman for the GNU project. The GPL is the most popular and well-known example of the type of strong Copyleft license that requires derived works to be available under the same Copyleft. The definition of derived work is not only related to the software source code, but direct, binary linking is also considered to be a derived work.

15. *EUPL (European Union Public License)*
The EUPL is a license that has been created by the European Commission. It's compatible with the GPL version 2, and contains a strong Copyleft clause. The EUPL's main strength is its focus on being consistent with the copyright law in the 27 Member States in the European Union.

16. The GPL is the most widespread license in use today, and it has also been tried in European courts. Although the EUPL and the GPL are compatible, GPL is the most used license in international Open Source projects.

17. *Dual-license*
Dual-license is a license mostly used by commercial companies, where the recipients can choose which terms they want to use or distribute the software under. It is mostly used for market segregation business models. Many OpenSource companies choose this licensing model so that they can harvest benefits from an OpenSource development model together with a traditional software licensing model. The OSCM would also have to be the sole copyright holder of the source code in order to be able to re-license the software under different terms. It is unlikely that an OSCM established by statistical organizations will have a business model that would require the use of a Dual-license scheme.

18. *Differentiated licensing*
Differentiated licensing means having different licenses for different components in the software. Typically one would license the main application under the GPL (an example could be the PC-Axis Windows version), and provide the core libraries with a LGPL license. (Lesser GPL/Library GPL). LGPL has an equally strong

Copyleft clause as GPL, but the main difference is that it allows linking. This means a third party that wants to create add-ons based on the core libraries can do so under their own license. If these add-ons require additional functionality or code in the core libraries, then the Copyleft clause will require them to license their additional code under the same license.

19. The most beneficial model to use for all parties, including commercial third parties, seems to be the Differentiated licensing model. This will always keep the core components of a software suite under a Copyleft license, without the possibility of re-licensing, but still make it possible for a third party that wants to make commercial add-ons to do so.

B. Practical consequences

20. The source code will be open for everyone. It should be stored in a revision control system, using services like OSOR, GoogleCode, Sourceforge, CodePlex. OSOR should be preferred. Software developed under these principles will be freely accessible under GPL for applications, and LGPL for software components

21. The OSCM decides which new software to accept into the software family. Any company or organization can support and develop the software developed, provided by the terms of the GPL and the LGPL. No company or organization can monopolize the software.

V. BUILDING AN OPEN SOURCE COMMUNITY (OSC)

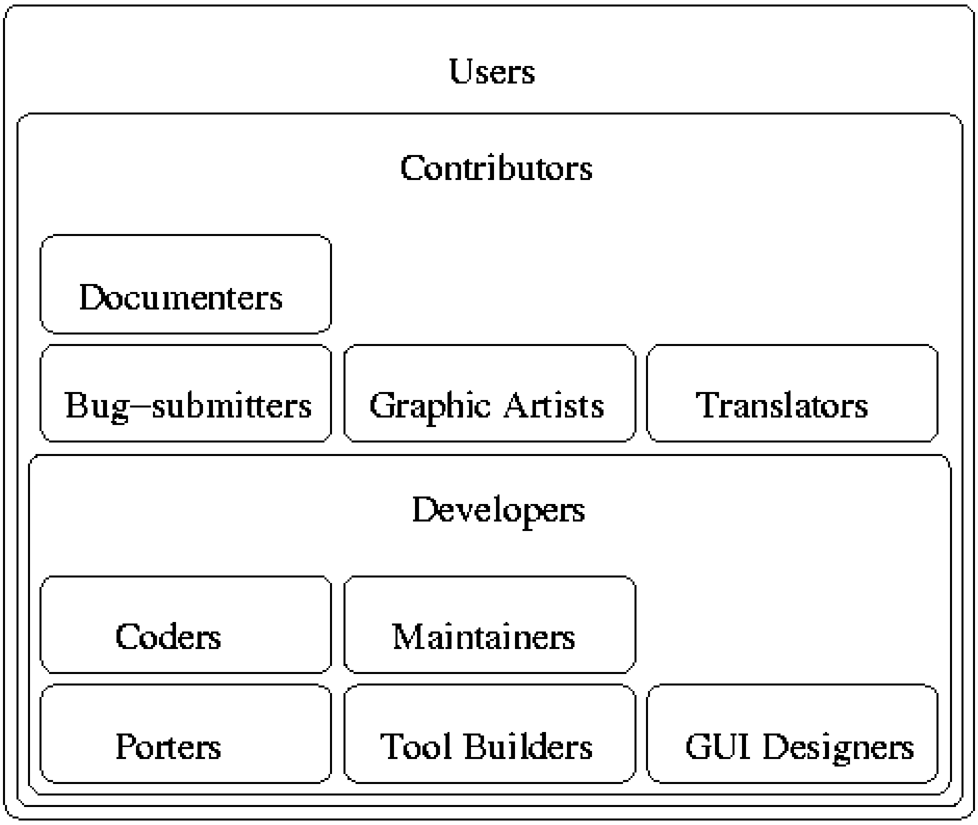
22. There are many different organizational models within Open Source projects, although most of them adhere to common principles for how they are organized. A development team is set up to take care of the operations of the community. The most important distinction between a top down proprietary/commercial development organization is that an Open Source project usually does not have a manager role. When this role is seen in Open Source projects, they are chosen by the development group based on merits and skill, and not necessarily on organizational hierarchy. There is a need to have tools to facilitate a decentralized and open organization, using mailing lists/forums, web based bug tracker-software, and code repository with a version control system.

23. For an OSC established among statistical organizations, most likely a consortium should be set up among the interested parties to define the roles and responsibilities to be held by the owners, i.e. initially those who provide resources in terms of developers from their own organization. The consortium will be the body to determine overall strategies and to decide upon development plans and other work to be carried out by the development team. However, a decentralized nature of the organization should be encouraged.

24. Figure 1 on the next page illustrates the common roles within the development team in an Open Source project. Multiple roles can be held by one or more persons. The need for these roles can be determined by the core team depending on the type and size of the project.

25. A group consisting of members from the IT function of the organisations forming the OSC should be formed to propose the roles and organization of the development team. In any case, the decisions related to the projects direction will in this case be a balancing act between the members needs and what the users, contributors, and developers outside the core development team want. All choices made in relation to the community projects direction should be done in an open way, either using communication channels like mailing lists, forums, or the project web page. Efforts should be taken to attract new contributors and developers to gain the benefits of using an Open Source development model.

26. The core project team will initially consist of people designated by the OSC members. Rules on how to include developers outside the core team should be made. Most Open Source projects will include external developers and contributors, based on merit, which (depending on what role this person has) is measured by the code quality the person writes, adherence to code standards etc.



[2]

Fig. 1

27. A repository should be chosen to provide information about the community, and to give access to the code maintained. The Open Source Observatory and Repository for European public administrations (OSOR http://www.osor.eu) should be investigated for this purpose. Access to the repository should be given to any developers and contributors who meet the rules set.

28. The roles of possible commercial contributors should be discussed and planned for. Finally, the way to set up and handle the end-user community should be considered.

VI. SOME CONCLUSIONS

29. The model outlined is only applicable when different organizations decide to take the responsibility for common development and maintenance of a specific software product or products forming a software suite. The objective of the model is to show that OSS development should be well organized, ownership must be defined, cooperation should be regulated by formal agreements, and that sufficient funding will be needed not only for development but also for future maintenance. Furthermore, external contributions to the development of the software must be handled according to the regulations defined by the license, which means that sufficient resources must be provided to follow up on this.

30. OSS itself is not a shortcut to reduce cost of development, but the cooperation in the development of the software could be a contribution to cost reduction. Whether this will succeed or not, depends on the commitment of those taking part in the development and the way the development project is driven, not necessarily on the licensing principle of the software. However, if you have been able to successfully develop

[2] http://www.oss-watch.ac.uk/resources/roles.xml

an open source project that gains momentum among users, then having followed the OSS principles could start to pay back, in terms of external contributions, user driven developments and perhaps commercial interest. In any case, reaching a high number of users should be beneficial for the funding partners.