

Open Source Software Release: A Guideline for the Publication of Software under the Open Source Status

Dr. Heidi Hohensohn; Ulrich Bretschneider and Jiayin Hang

C-LAB
Cooperative Computing & Communication Laboratory
Fürstenallee 11, D-33102 Paderborn
Ulrich.Bretschneider@c-lab.de
Heidi.Hohensohn@c-lab.de
Jiayin.Hang@c-lab.de

In the past few years, Open Source (OS) has become a topic for a far more heterogeneous community for both users and developers than it was before. Simultaneously, commercial aspects have gained a broad interest within the field of OS. More and more OS releases arise. The question becomes, what does it mean to publish software under an OS-license? Which working steps and aspects need to be taken into account for the publishing? Releasing sources under an OS license touches many different technical, legal, business and project management aspects. We developed a plan of procedure in the form of a workflow that involves all of these aspects. This plan will systemize the initial procedure of publishing software and gives decision support to everybody who deals with it. Our goal is to provide a checklist to expose potentially relevant questions at the right time and in a logical sequence.

1 Motivation

Deciding to choose an Open Source (OS) license when releasing software became more and more an alternative to closed source software (CSS) - in research and commercial activities as well as professional and private environments. OS releases are a question of different manufacturing levels of software, different motivations of initiating an OS project, and the several lobbies involved. These aspects influence a plan of procedure for initially releasing or starting an OS project and this chapter will provide a short overview of these aspects.

The different lobbies

In principle, an open source release comes into consideration for everyone who wants to publish their software. According to the original open source idea, first of all there are open source communities, whose members usually originate from private surrounding. Also this category of private users are “lone fighters”, who wants to make their source codes accessible to the public.

But possible candidates for the publication of software under the OS status are also software companies, using OS as a product accompanying strategy or to opening up a

market for additional services. Finally, an OS release is feasible for software, which has emerged from research activities of public institutions (for example universities, research institutes).

Multidisciplinary aspects are always affected. In every case, the decisions need to be made carefully, because in most cases they can not be reversed. A plan of procedure has to take into regard this heterogeneity.

The different motivations

There are different motivations to publish OS software. One intention for software companies is based on the consideration of making profit from additional business by distributing complementary services that directly connected to the software [Hohensohn/Hang 2003; Grassmuck, 2002, p. 34]. Additional services to accompanying a software product are e.g. consulting-, configuration-, implementation- and compiling services as well as trainings, documentations and special Add-Ons that are offered under proprietary status. Another incentive for software companies to engage in OS in this way arises from their hope to enhance its competitive position. For example, Netscape followed this idea when releasing its Internet Browser “Netscape Communicator” in 1998 under an OS license. By doing so, they hoped to get back market leadership, which was heavily threatened by Microsoft in the 90s [Osterloh/Rota/Kuster 2002].

A further motivation, especially for software companies, results from drawing benefit from Social Capital, an aspect that is discussed in the organizational theory [for more details see Osterloh/Rota/Kuster, 2002, p. 5]. The idea here is that the commercial organization utilizes the assignment of an OS community for research and development activities. For example, Netscape hoped to profit from Social Capital in that way when they released their Browser “Netscape Communicator” under a specific OS license and under the name of Mozilla [Grassmuck 2002, p. 307; Osterloh/Rota/Kuster 2002].

Beside these more business concerned aspects, traditional reasons for choosing the OS way are still relevant, like the possibility to increase reputation or to be driven by intrinsic motives, success and research ambitions, or just technical interest [Morner, 2000, p. 223; Lerner/Tirole, 2000, p. 21]. These reasons apply to OS communities, and private persons, that engage in OS projects, as much as research institutes. Research institutes especially see on the one hand a possibility to disseminate the results of their research activities and on the other hand a possibility to open up a door for follow-up activities by observing the software in use after a research project. In doing so, they can also document their assignment for the public. For example, the ILIAS software, an OS application for e-learning, was launched by the University of Cologne for this purpose¹.

The different manufacturing levels of software

The manufacturing level of the software to be released can range from a complete software product or just an add-on for a computer program to a program library or just a few lines of code. It is also possible to run the plan of procedure parallel to the development of an idea for a software code or line. Thus, software always stands in

¹ See <http://www.ilias.de>

the focus of interest at a certain level of completeness or simply as an idea for programming it.

2 The Procedure of Initializing an OS-Release

In general, when running an OS project appropriate preparations have to be arranged. These preparations can be organized systematically in several worksteps. The resulting process of preparation can be seen as the actual OS publication (compare figure 1), which finally results, depending on the aims of release, in a distinctive and long-term OS project. Depending on the objective of the release, there is more work to do when the OS project enters its long term running phase. This work concerns the maintenance of the project, which unlike the nonrecurring initiation process of preparation requires continuous effort (compare figure 1: community support and marketing of additional business).

This paper focuses on the aforementioned initiation process. The included worksteps therein are adequately introduced to its logic workflow. But the enumeration should not always be mistaken for sequential executing, some steps can be worked out in parallel like the illustration shows.

The intensity and necessity of editing every single working step depends on the objectives of release. So, some steps can be skipped according to previous decisions (e.g. step 4, if no commercial interests are involved).

The starting point of the process itself and each individual step does not strictly depend on the above mentioned manufacturing level of the software. Most of the steps, especially the first ones, can be passed through parallel to the manufacturing of the software and respectively to the formalization of the initial ideas for programming software.

In every case, at the end of the initiation process, the published software is held and further developed by a community in a long-dated OS project.

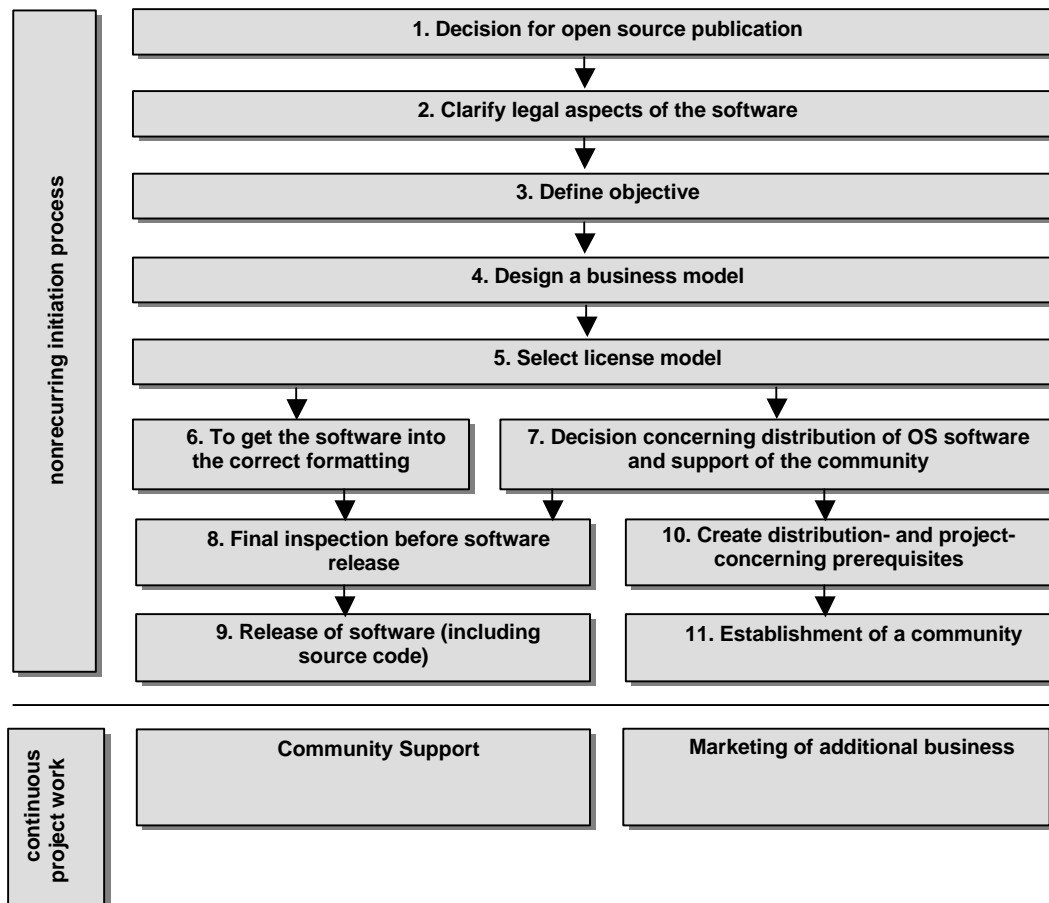


Figure 1: Plan of procedure for an OS software

2.1 Step 1: Decision for Open Source Publication

The first step of such a process is the fundamental decision whether or not underlying software should be transferred to the status of open source. The underlying importance of this fundamental decision is the fact that such a decision is irreversible in future. Finally, the decision depends on the simple economical circumstance, where the overall predicted benefit in case of the OS-alternative could be higher than that of other alternatives [Raymond 1999].

It is recommended to make the decision in the framework of a systematical and well considered decision making process [Steinmann/Schreyögg 2000, p. 158]. First, the corresponding framework of this project should be analyzed. For instance, the general social, technical and political trends and developments, as well as the risks and opportunities arising from the closer environment of involved parties, should be

analyzed. Additionally, the internal personnel, material and financial resources have to be examined. On the basis of this analysis, a final decision can be made in direct comparison with other possibilities that are available.

2.2 Step 2: Clarify Legal Aspects of the Software

If the decision is made in favor of the OS-publication, the contractual and copyright aspects affecting the software have to be solved. Contractual and copyright aspects of the software always result from the relationship between the involved parties that want to publish the software under the OS-status and a third party that can assert a legal claim on the software. If the OS publishing institution does not possess the copyright, corresponding transfers of the legal rights and approval for further utilization of the software will have to be clarified with the respective author as the owner of the copyright. In addition, contractual obligations of the software can derive from any agreements with third parties that are involved in any kind in the software.

2.3 Step 3: Define Objective

The task of this step is to define and formulate the goals to be pursued with the OS-publication of the software. Goals serve as a kind of guideline for the OS-publication within which the process can be controlled and aligned as required. For all participants, goals give orientation for realizing the project. Without a target-oriented alignment of the OS-publication the project threatens to deteriorate to a reactive adjustment to environmental changes.

The goals can be derived from the previously mentioned motivation pattern. So could, for example, the goal of a software company to generate additional business. Based on this goal, again more concrete goals on a lower level of abstraction should be again derived. In doing so, an individual hierarchy of goals can be built for every specific use case.

2.4 Step 4: Design a Business Model

This step refers specifically to the outlined objective of generating turnover by additional business or business opportunities. It only has to be carried out in the case when commercial interests are involved. The target of this step is to develop a business model for the additional business around the software. With this it is possible to generate specific sales volumes in future.

This means that the additional business has to be specified according to the definition and planned in detail. The results should be formalized in a business concept. In doing so, an action framework for the later realization of the business model is established and gives binding clues for all participating parties. In arranging a business model, the five dimensions of “market”, “profit model”, “resources focus”, “business strategy” and “enterprise organization” should be covered [Hohensohn/Hang 2003; Nehls/Baumgartner 2000]. These aspects correspond to the

components which are linked to the definition of business models in literature and present important parameters for the definition of business models.

So the dimension of market describes the relation between the OS publishing institution and the market, especially concerning additional business. Identification of target customers and market segments for the services of the additional business are of special importance. What is more, planning of essential marketing instruments like distribution, communication and pricing policies will have to be considered.

The question of which additional business the institution plans to generate revenues from, is a significant point within the elaboration of the business model. Within the dimension of profit model, the services and respective goods, that should be offered, must be designed. For instance, accessoires, trainings, consulting services or proprietary add-ons are distinguished in literature and in practice [Achtenhagen/Müller-Lietzkow/Knyphausen-Aufseß 2003, p. 470; Hohensohn/Hang 2003].

The dimension of resource focus deals with the appraisal of which kind and how many resources are required for the transfer of the business model at a later time. What comes to the forefront is the question of whether analogous resources are available in sufficient quantity and quality. Despite staff employment, it is also necessary to consider questions about factual and financial resources.

The dimension of business strategy refers to the development of measures that keep the supply of offered services and goods competitive in relation to competitors. For instance, the establishment of unique selling propositions is highly relevant in this context, because offerings are often very similar especially in IT-market surroundings.

The dimension of enterprise organization reveals the coordination between the internal processes of the OS software publishing institution and the business model. It is necessary to establish an adequate organization system or to adapt an already existing organization system of the publishing institution for the later sale of the additional services and goods.

Finally, the business model should be implemented after completely running the initial process. The marketing of the additional services and goods requires a permanent effort, therefore it belongs to the continuous project work at the end of the initiation process (compare figure 1).

2.5 Step 5: Select the License Model

After the specification of the OS strategy, the selection of a suitable OS license model can begin. The results of the preceding steps are the decision basis for choosing a license model. The aim of this step is to find a suitable license model for the OS software. The license model should be selected in such a way that it is helpful to the OS-strategy and of course adequate to the requirements of the OSI definition [see OSD 2003].

In general, there are two possibilities for the choice of a license model. On the one hand, an already existing license of other OS Softwares can be used as an example in order to choose a license adjustable to its own requirements. If necessary, this license can be continually adjusted. On the other hand, a completely new license model can

be defined. Choosing a new license model is always advisable when specific conditions are to be included. Otherwise, this possibility requires much additional work and expense.

Both alternatives have to take into account the question of compatibility when the considered software is to be combined with other software (e.g. program libraries, add-ons etc.), as much proprietary as OS software. In these cases the license conditions must fit to the ones of the considered software. That is to say that some licenses can mutually hinder the goals of the publications in mind. For example, if software components are added that are GPL licensed the conditions of the GPL stipulate that the considered software must also be licensed under GPL conditions.

2.6 Step 6: Get the Software into the Correct Formatting

Open source software as well as their source codes should comply with certain formal requirements so that they are easily accessible and understandable for all possible users and programmers. The user friendly format of the software increases the probability of a broader user community. These user communities refer in this context above all to those developers involved later in the process as well as the final users of the software. In this step, the developed software must be adapted so that it is usable and as user-friendly as possible for the OS community. For instance, the source code should be transferred into an easily and quickly understandable format, analogous software documentations should be offered and the delivery for the potential user should be simplified to the greatest possible extent.

2.7 Step 7: Decision Concerning Distribution of OS Software and Support of the Community

When considering the distribution of the software, the actual physical distribution of the software is meant. The support of the community comprises activities such as collection of suggestions for improvements of members, the moderation of forums, and the possibility of making information- and communication channels available. The community support is a working step possibly gaining importance at the longtime-running phase of an OS project and should be started immediately after the initiation process is completed (compare figure 1).

The distribution and the community support are in practice carried out frequently by the same carrier, although these tasks can also be conducted separately from the technical point of view. This procedural model should correspond to the usual practice and avoid the possibility of a separate view. Accordingly, a self-distribution from the perspective regarded here will involve a later community support.

An aim of this process step is to make a decision about a self- or a foreign involvement concerning these two aspects. Should these tasks be fulfilled by the publishing institution itself or by a third party? The results of step one and three are essentially the information basis for this decision.

For the goal of mere publishing the software without any further intention, it makes sense to give away the distribution into other hands and therefore to spare resources. For the foreign distribution as well as the transfer of the community support, open source portals in the Internet are suitable. Open source portals bring together the developer and customer of OS software in one market place [Hang/Hohensohn 2003, p. 60]. They serve as intermediaries between supply and demand and try to act between developers, users, distributors and commercial software providers. Besides this they often serve as developer platforms. The most well-known OS portals are e.g. SourceForge, BerliOS, Savannah, and NOW.

For strategic reasons however, it is sometimes more suitable to take the distribution and community support into one's own hands in order to keep full control and secure continuous access.

Due to the logical operational sequence of the procedural model, this step can run parallel to the previous step (step 6).

2.8 Step 8: Final Inspection Before Software Release

With this work step, it should be ensured before software is actually released that the processing of all preceding steps was carried out according to the target guidelines and in a correct manner. A monitoring process should be designed and take place in a methodical review. Detected failings from that review should be removed immediately.

2.9 Step 9: Release of the Software

If the control process is finished and possible weak points and bugs have been identified and eliminated, the software can be released. In case of a 3rd-party distribution and community support, this means that the software, including the source code, can be handed over. If the result of step 7 is that the distribution and community support should take place in house, the responsibility for the distribution should start. The corresponding step 10 "Distribution and Projekt Managerial Prerequisites" should be fulfilled until then. Simultaneously to the release of the software, the foundation of a community should start (step 11).

2.10 Step 10: Create Distribution- and Project-Concerning Prerequisites

In this step the corresponding prerequisites for the establishment of a community have to be created, if the decision is made in favour of self-distribution and in-house community support. The aim is to create a platform for the collaboration of the future community members. For this, the Internet serves as an important medium [DBUS 2003; Raymond/Lohmeier 2001].

In concret, a specific webpage should be installed and run. On the one hand, it will serve as a permanent distribution channel for the software and as a communication-

and information channel for the community support. On the other hand, a technical development environment including tools for coordination, communication and direct software development support, like CVS or bug reporting facilities, for the collaboration of the community-members has to be installed.

In order to identify and inform communities and individuals e.g. mailing lists, discussion forums and news groups represent important elements [Behlendorf 1999; DBUS 2003; Raymond/Lohmeier 2001]. Of course, an adequate download area for the distribution should be created. On the one hand, the source code should be available in this area. On the other hand, of course, the compiled software itself as well as corresponding software documentation.

2.11 Step 11: Establishment of the Community

The task of this step is to establish an OS community. The focus of this work step lies in acquiring and motivating project members as much as in the structural organizing of the community. In favor of that, the tasks to do can be divided into organizational and personnel aspects.

According to hierarchies in organizations, in OS communities often can be found hierarchic structures [Markus/Manville/Agres 2000, p. 21; Morner 2001, S. 5]. Commonly at the top of this hierarchy a key team is established in order to moderate the whole community. In order to guide the communities development according to its own goals and to benefit from its Social Capital (compare chapter 1) the publishing institution should take this role over. Furthermore, the organizational aspect of task sharing is discussed in the literature as an important factor of success [Lerner/Tirole 2000, p. 21; Markus/Manville/Agres 2000, p. 21; Morner 2001, p. 5].

In order to control the quantity and quality of a community's membership, the so called concept of "Managed Membership" stands the test in practice [Markus/Manville/Agres 2000, S. 21; Morner 2001, S. 5]. This means the conscious selection of the membership in order to keep the project under control.

In favor of leading a community in this way, it is very important to acquire and motivate its members. Because of this, adequate incentives for the members must be created. For example, there should be installed possibilities for the members to establish their own reputation, if this is an important motivation factor for taking part in OS communities (compare chapter 1).

Solidarity among members of communities can often be observed. This solidarity is last but not least based upon a corporate macro culture. A macro culture is a system of the members' fundamental values and attitudes towards the specification of relations among colleagues themselves as well as towards external people, opinions, ideas, ideals, roles within the community and their members. All members share and accept the macro culture. Thus macro cultures can coordinate and promote interdependent and even very complex activities of such communities [Jones/Hesterly/Borgatti 1997, p. 929]. This point applies also and notably to OS-Communities. Even though their members are only virtually interconnected, the installation of a culturally divided environment of the community allows for a successful cooperation.

After the initial foundation, depending on the objectives of release, the contact to the community should not be dropped. In order to generate revenues from the communities' work, a close contact should be established and cultivated, because in the end, the goal is to gently control the project in accordance with the objectives of the in house OS release. In this phase of continuous project work (compare figure 1: community support), mainly organizational and administrative tasks need to be accomplished to keep the OS project running.

3 Conclusion and further research

Releasing software under OS licenses is a complex and multidisciplinary task. The scheme presented here is a first approach to list and organize the relevant decisions to be made. This has to be continued and further tested; checklists and best practices have to be assigned to the single steps in order to provide an improving base for potential publishers from different backgrounds.

If the single steps of releasing software under the OS status are better understood by every involved persons OS software could be established for a more sustainable software process to an overall benefit.

References

- [Achtenhagen/Müller-Lietzkow/Knyphausen-Aufseß 2003] Achtenhagen, L.; Müller-Lietzkow, J.; Knyphausen-Aufseß, D. zu: Das Open Source Dilemma: Open Source-Software zwischen freier Verfügbarkeit und Kommerzialisierung, in: Zeitschrift für betriebswirtschaftliche Forschung (zfbf), 55 (2003), pp. 455 – 481.
- [Behlendorf 1999] Behlendorf, B. (1999): Open Source as a Business Strategy, in: DiBona, Ch.; Ockman, S.; Stone, M. (Ed.): Open Sources: Voices from the Open Source Revolution, O'Reilly Verlag, Sebastopol, CA, 1999.
- [DBUS 2003] DBUS (o. J.): DBUS! Deutschland, published in the Internet: URL: <http://www.dbus.de/eip/inhalt.html> [11.09.2003].
- [Grassmuck 2002] Grassmuck, V.: Freie Software zwischen Privat- und Gemeineigentum, Bonn, 2002.
- [Hang/Hohensohn 2003] Hang, J.; Hohensohn, H.: Eine Einführung zum Open Source Konzept aus Sicht der wirtschaftlichen und rechtlichen Aspekte, C-LAB Report, Vol. 2, No.2, 2002, published in the Internet: URL: <http://www.c-lab.de/home/de/reports/index.html> [21.07.2003].
- [Hohensohn/Hang 2003] Hohensohn, H.; Hang, J. (2003): Product- and Service Related Business Models for Open Source Software, Proceedings of the Net.ObjectDays 2003, Erfurt, Germany.
- [Jones/Hesterly/Borgatti 1997] Jones, C.; Hesterly, W. S.; Borgatti, S. P.: A General Theory of Network Governance: Exchange Conditions and Social Mechanism, in: Academy of Management Review, Oct. 1997, Vol. 22, Iss. 4, pp. 911 – 945.
- [Lerner/Tirole 2000] Lerner, J.; Tirole, J.: The Simple Economics of Open Source, Working Paper No. 7600, National Bureau of Economic Research, Cambridge, December 2000.

- [Markus/Manville/Agres 2000] Markus, M. L.; Manville, B.; Agres, C. E.: What Makes a Virtual Organization Work?, in: Sloan Management Review, Fall 2000, Vol. 42, Iss. 1, pp. 13 – 26.
- [Morner 2000] Morner, M.: Das Open-Source-Software-Phänomän – organisatorisch betrachtet: Wie die Organisation großzahliger Projektgruppen über das Internet gelingt, in: Zeitschrift Führung und Organisation (ZFO), 71. Vol. 4 (2002), pp. 219 – 225.
- [Nehls/Baumgartner 2000] Nehls, R. G., Baumgartner, P.: Value Growth: Neue Strategieregeln für wertorientiertes Wachstum – Ein Ansatz von Mercer Management Consulting, in: Management Consulting, München, 2000, pp. 79 – 93.
- [OSD 2003] The Open Source Definition, Version 1.9, published in the Internet: URL: <http://www.opensource.org/docs/definition.php> [16.10.2003].
- [Osterloh/Rota/Kuster 2002] Osterloh, M.; Rota, S.; Kuster, B.: Die kommerzielle Nutzung von Open Source Software: Der Einfluss von sozialem Kapital, in: Zeitschrift Führung und Organisation (ZFO), (2002), No. 4, pp. 211 – 217.
- [Raymond 1999] Raymond, E. S.: Der verzauberte Kessel, 1999, published in the Internet: URL: <http://www.oreilly.de/opensource/magic-cauldron/cauldron.g01.html> [10.09.2003].
- [Raymond/Lohmeier 2001] Raymond, E. S.; Lohmeier, M.: Software Release Praxis HOWTO, published in the Internet: URL: <http://gd.tuwien.ac.at/opsys/linux/DE-HOWTO/HOWTO-test/DE-Software-Release-Praxis-HOWTO.html> [11.09.2003].
- [Steinmann/Schreyögg 2000] Steinmann, H.; Schreyögg, G.: Management: Grundlagen der Unternehmensführung: Konzepte – Funktionen – Fallstudien, 5. ed., Wiesbaden: Gabler, 2000.