Interacting with the Open Source Community - Effective models for a Traditional IT-business

Dr. Wolfgang Thronicke and Frank Berger

Siemens Business Services C-LAB, Fürstenallee 11, 33100 Paderborn
frank.berger@c-lab.de wolfgang.thronicke@c-lab.de,
WWW home page: http://www.c-lab.de/dis

Abstract. Open source has emerged as a powerful concept in software development and marketing in the early 21th century. This paper explores how open source communities work and how they co-exist or thrive in co-operation with "traditional" software companies. This analysis of given best-practices is backed up from first-hand experiences gained in the German NOW project focusing on efficient usage of open source concepts in industrial contexts.

1 Introduction

During the last decade open source software has gained an increasing percentage in the statistics of used software and sometimes competes with traditional solutions of software vendors. A prominent example is the Apache[1] web-server which is a direct rival to the Microsoft Internet Information Server[2].

Aside from such success stories open source software has already been existing since the appearance of the Internet or – more generally spoken – computer networks. There has been a considerable development from shared code snippets to organizations as the FSF[9].

This contribution will give a concise analysis of the open source phenomenon in the next chapter and then proposes types of collaboration for the industry in order to participate in the open source community or include open source into their value-chain. This is supplemented by a description of the German NOW project which researches the modes of open source adoption by the industry.

2 Situation briefing

2.1 What is open source? A short answer.

The term "open source" has already been defined on various places[3, 4]. In short open source software is software where the user of it has unrestricted access to the source code. What he is allowed to do with the sources and what not is usually defined in the open source licenses[6]. Be aware that open source does not generally mean "without copyright" or "free to use" although it is mostly used in publication with the interpretation of "free" open source[5].

There is a clear distinction between open source and free-ware which most times excludes the source but allows a royalty-free use without any warranties. The history of free-ware itself reaches back as far as that of the open source. Notably even software companies have been providing small tools or helper applications “free of charge” since the beginning of the PC/workstation era.

Aside from this open source is more than just people writing software for free. The term embraces various aspects: a community, a certain commitment to a way of software design where the motivation to write and the deployment of the “product” are part of unique processes aside from pure business driven companies[7].

2.2 About the open source community

In order to understand how open source works it is important to identify the driving force of open source software development. In principle their are three contributing factions as depicted in figure 1. Historically individual programmers were the first to create open source software, mostly using resources from their employment site, especially universities. In this process these institutes were likely to allow the deposit of open source software on their existing server infrastructures[1].

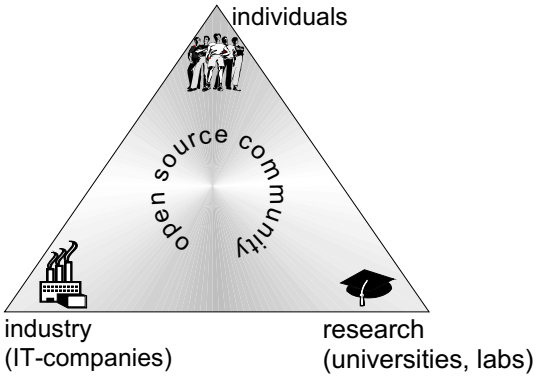


Fig. 1. The open source community triangle

Several research faculties have been providing their own software on a near open source type of license, too. Actually this means some type of federal support of the open source community. Later the IT-industry entered the open source arena. Since this is a highly innovation and technology driven economical branch, open source is a great playground to develop new technologies and try them out in a broad community, gather feedback and perhaps even acquire skilled programmers in the process[2].

[1] FTP servers in the beginning

[2] This subsumption is of course not detailed enough. Actually a complete book can be written on this subject alone.

What is nowadays called the open source community can be seen as a loosely group of writers *and* users of open source. This means people working and accepting the rules and code of conduct comprised by this type of software. This "code" allows users painlessly to report bugs to the developers and exchange ideas how the open source project should develop. On the other side in contrast to contractual handling by software companies, the developers usually decide and rank what to do first and when to release. This means aside from the different forms of open source licenses the community lives in a mostly contract-free habitat where common goals direct the next steps of software evolution. Up to now using open source has been with tacit understanding that users do not ask for damages[3].

2.3 About open source success

Open source software has gained great interest especially since its use has become popular and is endorsed even by the government[12][4]. Many IT solutions base on the use of open source components. The Apache web-server[1] is a good example, the collection of GNU tools[10] another.

The main question here is: What counts as a success? Since "free" open source usually can not be measured in financial units or return of investment, few objective gauges remain of which the number of press articles is not really one of them[5]:

- *Number of installations.* Based on the number of installations there are some impressive open source projects: The GNU compilers (with a long tradition), Samba, Apache, and of course Linux. Comparisons still remain difficult and only make sense if they are relating to the same type of software. For instance Linux may be installed on many hosts, whereas a web server like Apache is likely to be installed only once per company, but they are accessed by thousands of users.
- *Number of users.* The number of users usually relates to the installations. One of the nice things in the realm of open source is[6]: *if you don't like it, you don't use it.* Some software may of course be used unnoticed from their users like a Samba server, or an Apache web-server[7].
- *Size of active open source community.* The active open source community consists of the developers and maintainers and the group of users that actively give feedback and hence foster the development. This can be gauged in the newsgroup traffic, or in mailing lists.
- *The vitality.* Living open source software is constantly adapting to new user needs and new environments (operating systems). This vitality is expressed in the number of bug-fixes and releases.

[3] Lawyers have discovered open source as a new fruitful field of exploitation, where in order to protect the rights of an open source user every developer may be made responsible for any error causing damage. More information about this topic can be found in [8].

[4] A search for the word "open source" and "government" finds examples for this all over the world.

[5] The number of articles is more hype-driven.

[6] most times

[7] This demonstrates how nicely open source solutions integrate in the software world, too.

Evidently all this points are interrelated. Without installations there are no users. Without users there is no foundation for the community.

It must be distinguished between personal and industrial use of open source, too. Depending on the type of open source software, success can be limited to the personal area, for instance a recipe database and management system. On the other side open source used in commercial scenarios directly contributes to the value creation chain. This view is quite suitable to compare open source solutions with those coming from traditional software vendors. And in fact, particularly due to the increased "success" story around Linux, the market share of open source is growing steadily[11].

2.4 Open source in the IT industry

The former section already contains the key point for using open source in commercial contexts: It can endorse the value chain. Several modes of using open source are possible:

- *As developing aid*. Open source tools have been in use by developers for a long time. Version management and compilation are two prominent examples. In this role "free" open source software is a cost-effective tool to use, because no license fees are to be paid.
- *As product component*. Open source software may be a cost-reducing part in more complex solutions, reducing investments and allowing a competitive pricing of products.
- *As "professional" supported product*. With the development and deployment of open source in focus a viable business model is to offer professional support and consulting for customers. In return they benefit from guaranteed response times on errors or assistance if problems occur.
- *As strategy for development and marketing*. Companies doing their own research in the IT area may chose open source as one key strategy to implement new technologies and access a broad range of user opinions. One example is the Alphaworks project from IBM[15], or the new Sun Java Network[14].

From a more abstract viewpoint open source has implemented a working distributed software engineering methodology where a virtual community[8] has been formed to pursue a common objective. Despite the flexible handling of open source projects and their non-rigid approach they perform highly effective and provide high quality software as output.

3 Making open source work in commercial contexts

Although the header of this section sounds at first highly contradictory stating that something free accessible by nature can work in a usually closed commercial environment there are connecting points as already discussed in section 2.

It is premature to believe that most open source software projects are self-sustained. There are always resources involved that are not for free:

[8] A *virtual community* is a community of persons who have never met in reality.

– Storage on a sever,
– Internet access,
– E-Mail and other communication services, and
– the development tools and environment.

The development environment is usually contributed by the community member himself, whereas the former clearly relates to "community resources". These have often been "sponsored" by companies or universities in former times as part of a personal webspace. Without going into details, now using specialized service portals like Sourceforge[16] or Savannah[17] have become the standard for open source projects. These portals provide all the necessary infrastructure and services to support community development and management.

3.1 Modes of cooperation between industry and community

Supporting these "meta" open source organizations defines the first way for a company to engage in this domain:
Collaboration type: Sponsoring of the open source community in general.

However, this results only a general honoring of the open source community and getting noticed in it. In order to form a closer co-operation with the community in some projects which are relevant for the value chain it can be sensible to offer hosting of open source projects at dedicated accounts.
Collaboration type: Hosting of selected open source projects.

The advantage of this approach is to have direct access to the results of the project. Of course, doing so, requires to "play by the rule" and give unrestricted access and keep the site up and running. It must be mentioned that placing such a host in the company Intranet may pose a security problem!

With this form of collaboration the active participation is only the next step. A company can assign developers or a complete team to the task of managing the open source project. In practice, the company developer team becomes the core of the project with an active layer of external developers and users. This model is very suited if a commercial version of the open source software exists as product, for it renders a maximum of control over the software together with the beneficial work of the community.
Collaboration type: Manage in-house the open source projects.

Of course, these cooperation models only constitute a part of possible realizations. We refer to these as "direct" because they focus on the open-source community, and there is no difference between the open source version and the one distributed to paying customers. Another variation can be described as follows: A product version and an open source version exist in parallel. The product version is released as open source as soon as a new product release is available.[9]. Depending on the involvement the open source part can be maintained in-house or completely in the open source domain.

The indirect cooperation with the community can be seen in the example set forth by Nvidia Cooperation which provides binary (and not open source!) drivers for the

[9] This model is employed with the ghostscript postscript interpreter or the binary ports of GNU tools from Cygnus Solutions

XFree Windows Systems (http://www.xfree.org) that is used by many open source unix operating systems. Thereby the position of the open source alternative is strengthened and the open source project is supported.

3.2 Risks

Aligning the business models of a company with open source is no guarantee for success, nor a way to "save" a slackening business. First of all, costs are involved and the principles of the community have to be abided. Worst example are companies placing the words "open source" on their sites and not providing any download.

"Owning" an open source project implies a degree of responsibility. To divert the developing in-house members from an open source project (and the community) can easily result in annoying the community. On the other side the community can compensate shifts in man power. It is possible to transfer a project from an in-house open source project to Sourceforge and invite external community members to adopt a more active part in the project. The open source community often reacts very fast to new technologies. A company with a more traditional setting may have the problem to keep their projects alive[10].

Aside from legal issues that have been mentioned earlier in this paper there is always the risk of failing to create an open source community. The dynamics of open source projects are still not completely understood and the factors what makes some projects work ("the sexy ones") and others not really need to be analysed.

3.3 Advantages

Some benefits have already been pointed out in the introduction as the usage of existing open source solutions or reusing open source components for new products.

The advantages are clearly seen for companies with emerging technologies in their focus. The community gives quick and accurate responses to what they want and where they want to go. This can give an open-source-aware enterprise the edge before other competitors. Working closely with the community helps to maintain short release cycles and efficient bug reporting and fixing, hence improving the code at a fast pace with moderate costs.

The effects of open source can be to penetrate the market and set de-facto standards, or even a common one. Or it can be used to propagate an existing standard.

4 NOW - A German research project to bridge the gap

The German research project NOW explores how commercial oriented enterprises can benefit from the software engineering practices pertaining in the open source community. NOW is the German acronym for 'Advantages of using the open source concept in commerce and industry'. The project consortium consists of three industrial members and one university partner[11].

[10] Of course, open source software has a life-cycle, too. But strangely some software may have an afterlife of its own, despite "better" solutions.

[11] Siemens AG, Siemens Business Services, 4soft, Technical University Munich.

4.1 Goals of the project

The primary goal of this project is to point out an effective model how a joint endeavor between the open source community and a "traditional" software company works effectively. The project therefore investigates the problem area from various angles:

- *Business models.* One result of the project will be working business models for interested companies. Clearly, there are already working models but they do not apply on a development project level. Aside from "common" open source models, for example apparent in the "Linux"-industry, small and medium sized companies need new business models to utilize the potential power of open source software projects in their own business processes.
- *Software engineering methodologies.* Open source software development differs from prevalent in-house software engineering processes. In order to combine both NOW analyses the differences and communities and develops a suitable model for this type of software design.
- *Community Management.* An important question to be answered is, how to find right partners in the open source community for an open source project and how to manage the project. Therefore a community platform based on portal technologies - of course, open source - is being created to serve as virtual meeting point between interested companies and open source developers. This portal (see section 4.2) will also contain material or links to material of interest for creating and managing open source projects.

4.2 Portal technologies

During the project time an internal and and external portal is being set up. The geographical separated project partners utilize this NOW-portal to control, communicate with each other, and coordinate the project. The external public portal serves the same purpose for the user community. The NOW portal infrastructure is a combination of different open-source portal solutions. What is already discovered is, that a single portal based on a pure community portal like PHP-Nuke or one dedicated to open source projects can not meet the requirements laid out in the NOW-project. Therefore a mixed architecture as depicted in figure 2 has been implemented using reliable open source solutions:

- Apache[1] as web-server,
- MySQL (*http://www.mysql.com*) and PostgreSQL (*http://www.postgresql.org*) as database,
- PHP (*http://www.php.net*) as implementation solutions for PHPNuke (*http://www.phpnuke.org*) and GForge (*http://gforge.org*) sub-portals.

Other portal solutions have been evaluated during the project, but the current selections offers the best stability and extensibility.

PHP-Nuke is a PHP-based Content Management Portal which stores the supplied information in a MySQL database. This role based portal includes different modules. The most useful modules for the NOW-project are the discussion board (Forum), the calendar for project planning, the news archive, and the download area. All these modules and functions allow an efficient communication of the portal users.

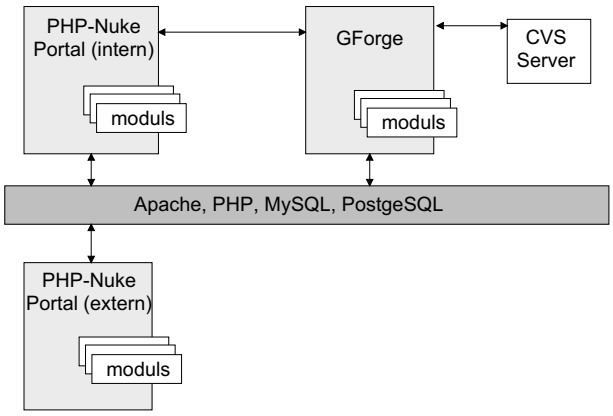


Fig. 2. Portal architecture

GForge is PHP-based, too, and it offers the user all services necessary to develop open source software. All informations are stored in a PostgreSQL database. The most prominent modules of this role based portal are the task-manager for assigning jobs to the developers and perform a simple project management and tracking, mailing-lists, bug-tracker, forums, and a CVS[12] repository interface. The GForge CVS interface plugs in a CVS server storing the software sources. The license type for an open source software project can be chosen from various license models: The GNU General Public License, the GNU Lesser General Public License (LGPL), the Mozilla Public License (MPL), etc. (see [6])

PHP-Nuke and GForge are role-based[18]. Users take different roles in different development projects. A role is an association of different rights. For instance a user can be owner of a discussion board and manage the entries and users while others have only read or write access to articles.

4.3 Advantages of NOW for joint open source and industrial software development

NOW will establish a bridge between the open source community and "traditional" software producing companies. The main benefit is that is does not solely address the needs of the developers, as this would be a simple copy of Sourceforge or other repositories. In the NOW-portal deciders and managers can interact, too, and get information on legal issues or tips how to combine the internal value chain with an open source development approach. For skilled developers this can be a platform for finding new jobs in projects, if suitable compensation models can be derived in the business models. But one key effect may be to remove the uneasiness most managers feel when turning to open source based software production as this can have great new business opportunities.

[12] http://www.cvshome.org

5 Conclusion

This contribution has summarized the actual open source situation with a special focus on the commercial "integration". Although it is currently en vogue to wear the open source label, not all business models succeed. There are success stories and they must be carefully analyzed. The basic issues for a prosperous combination of open source and traditional software development are:

- Understand, how the community works,
- implement a collaboration model with the community, and
- secure all legal issues.

The national project NOW will provide a lot of insight by using a portal approach to manage and monitor the development of such a mixed community using some selected open source projects for hosting. Additionally it provides an analysis of viable business models with one focus placed on the legal repercussions.

Our future work will be centered around this project, technically augmenting the portal with methods compliant to both open source and industrial needs, and cast a deeper look into the mechanisms driving open source.

References

1. http://xml.apache.org
2. http://www.microsoft.com/iis
3. The Open Source Definition. The Open Source Initiative. 2003 http://www.opensource.org/docs/definition.php
4. Open Sources: Voices from the Open Source Revolution, O'Reilly, 1st Edition January 1999 ISBN 1-56592-582-3.
5. Tim O'Reilly. Ten Myths about Open Source Software, 1st Nov 1999, http://opensource.oreilly.com/news/myths_1199.html
6. http://opensource.org/licenses/
7. Eric S. Raymond, The Cathedral and the Bazaar, http://www.firstmonday.dk/issues/issue3_3/raymond/
8. Thomas Schiffner. Open Source Software - Freie Software im deutschen Urheber- und Vertragsrecht. München: VVF - Rechtswissenschaftliche Forschung und Entwicklung, Bd. 690. 2002. ISBN 3-89481-465-9.
9. The Free Software Foundation Europe. http://www.fsfeurope.org
10. The GNU Project. http://www.gnu.org
11. Soreon Reseach GmbH. The Market for Open Source Software in Germany. 2003 http://www.soreon.de/html/studien/opensource_studien.htm
12. Gabriele Albers, Bundesregierung setzt sich für Linux ein. Financial Times Deutschland, 6.7.2001
13. http://www.samba.org
14. http://www.java.net
15. http://alphaworks.ibm.com
16. http://sourceforge.net
17. http://savannah.gnu.org
18. Theis, F., Kuhn, S., Langham, M., Mueller, J., Wang D., Ziegeler, C.: Portale und Webapplikationen mit Apache Frameworks. Frankfurt 2003, ISBN 3-935042-36-1, Software and Support Verlag GmbH

This work has partly been funded by the German NOW project under grant number 01 ISB 04 D.