Business Opportunities of Companies Active in Open Source Projects

as

Special Study

at the

Faculty of Economics and Social Science

of the University of Bern

submitted to

Prof. Dr. Thomas Myrach

Institute of Information Systems

by

Matthias Stürmer

from Pfäffikon ZH

in the 9th Semester

Matriculation Number: 00-114-256

Address:

Wabernstrasse 51

CH-3007 Bern

(phone: +41 31 371 80 87)

(e-mail: matthias@stuermer.ch)

Bern, 2005-03-15

Table of Contents

1. Introduction

Participating as volunteer in an open source project (OSP) is a challenging but also personally rewarding possibility. Intrinsically motivated countless numbers of software developers spend their spare time working intensively in OSPs. On the other hand open source involvement of companies needs to be justified by economical reasons. Although especially in small firms ideological motivations might still be present, all profit-oriented institutions need to meet financial goals in the long run. Thus in this paper the focus is set on the business opportunities of companies that initiated an OSP or permanently participate in further development of one. First, the characteristics of company founded projects are examined. Second, economical reasons why companies successfully are able to participate in OSPs are described. And third, different activities are discussed how companies can work profitably with open source software.

1.1. Goals

The main goal of this paper is to provide support to companies involved in OSP development. They receive recommendations how they can generate revenue with their special position being the initiator or a very active contributor of an OSP. Commercial success of such activity is enabling them to further participate in the OSP and thus also advantaging the open source movement. Therefore it is necessary to find the motivations of OSP initiators, to elaborate advantages and challenges of companies starting and leading OSPs, to show ways of collaboration with external contributors and other companies and to develop strategies and recommendations of behaviour to successfully conduct business activities.

1.2. Methodology

This paper is based on literature and on eight interviews with OSP leaders[1] reproducing their statements within the context. For qualitative analysis of the conducted interviews the software MAX.QDA2 was used creating 191 text codings grouping together corresponding comments.

1 See Stürmer (2005)

2. Company Founded OSPs

As introduction into the area of business aspects of OSPs the two forms of initiating such a project, either by a community or by a company, are compared. To focus on the commercial perspective the major advantages of company founded or sponsored OSPs are described and the challenges during the initialisation phase and during further development of the software are discussed.

2.1. Comparison to Community Founded OSPs

Initiating an OSP may occur in various ways. On one hand there are so-called community founded OSPs like Linux, the Apache web server and GNOME which all were started by one or more individual programmers. The initial source code of the software was still incomplete at the time of publication and thus needed the support of other contributors. The initiators had to motivate other people to participate in development and to build up an organisational structure for collaboration.[2]

Such community founded projects have the advantage that the usual incremental and parallel growth of the community next to the code base introduces participants slowly to the community giving them the opportunity to shape the software's architecture and the community norms.[3] Also the absence of a profit oriented institution ideologically motivates contributors to invest personal time into the OSP because they don't feel exploited by a company as Daniel Hinderink, marketing coordinator in the TYPO3 content management system, knows from his experience with the community: *"I believe there are more advantages of this fact [that TYPO3 was initiated by a person] so far. Kasper [the initiator of TYPO3] earned lots of sympathy by not doing business with TYPO3 for a long time. It psychologically lowered the entry bar of the project a lot. My assumption is that community contributions are much harder to get for a project where a company is doing business with - just because people feel like helping the competitor or supporting somebody who has business interests. In the case of Kasper he as an individual still benefited from the success but he also had invested a lot into it. There's a much more personal relationship to the project if there is an individual programmer at the centre."*

2 See West/O'Mahony (2005), p. 1

3 See West/O'Mahony (2005), p. 7

On the other side there are also significant disadvantages starting an OSP by volunteers. West and O'Mahony name start-up costs as the main challenges of community founded OSPs. Building up organisational structure, collaboration technologies and technical infrastructure requires a large investment of the participating persons thus it's not surprising only few volunteer projects ever establish an active and productive online community.[4]

One way to meet these challenges of purely community founded projects is to sponsor the initial development work by a company but afterwards to "slacken the reins" so the community receives the control over the project. When the software company Infrae needed a rich text editor for their content management system they hired Guido Wesdorp to customize an existing editor. Besides company time he also invested also a lot of personal time into the project thus making it a partly sponsored and partly community founded project. Describing how the project was started the unusual mix of work and spare time is shown making this project possible: *"I probably wouldn't have done it if it wasn't for the time I could have spent at Infrae. [...] I liked the idea so much writing a What-You-See-Is-What-You-Get editor that I decided to propose to do the initial work and a lot of the maintenance work in my spare time. And to do the Kupu work that is actually interesting for Silva [a content management system] during Infrae hours. Infrae has written Silva and some customers wanted a What-You-See-Is-What-You-Get editor for that so that's why we went looking for a What-You-See-Is-What-You-Get editor for the first place. Kupu is only partially a company sponsored project but the company backup was definitively what made it possible to have got a feature complete editor. So I think I spent about half the time at the company and half the time at home. Kupu wouldn't be what it is now if it weren't for either one of those parts of time. It's really a sort of a bit weird mix. When I was at work the Infrae people were pushing me to get the Infrae work done and while I was at home I looked at Kupu from a completely different perspective, from a manager of an open source project. While I was at work I looked at it from an integrator's perspective. It was much fun to be these two persons at one time, also in relation to the company."*

4 See West/O'Mahony (2005), p. 2

2.2. Advantages

Looked at the advantages and challenges of community founded OSPs now company sponsored projects are closer examined. West and O'Mahony describe two aspects why external contributors are motivated to join development of a company sponsored OSP. Since the initial work on such a project was financed by a firm the community already receives a working basis of the software and can also expect the company to continue investment into the project.[5] Bård Farstad, founder of eZ systems initiating the eZ publish content management system, confirms this conclusion of West and O'Mahony by pointing out the consistency the company give to the project: *"The advantage is you get consistency. People are dedicated for development over time. You are working towards a goal and you work very focussed on that goal. That is the main advantage for a user. Also if a company is behind the project you get the possibility of getting the professional services and support you need, especially if you base your business on this solution. And you get the comfort to know in a month there will be a new release containing those features. You know what's going on."*

The second aspect described in the work of West and O'Mahony lays the focus on the responsibility a company has towards its project's customers thus guaranteeing the future progress of the software. The on-going sponsorship provides resources to support the further development of the software and assists in areas where no volunteer contributions are received.[6] Being in the position of the OSP initiator of Apache Lenya as well as director of the company wyona, Michael Wechner comments on his dependency on the OSP. He describes the responsibility he has towards his customers thus not being able to change technical things as quickly as in some spare time project: *"Here you see the difference between 'hobby' projects and those that are used in a commercial area. You will have some dependence on your customers and can't just say we will change the API like that. In software engineering it happens that after a while you realize you should have done things differently. If you are a single developer you can say, even if others depend on it: 'I don't care, I want to do it my way!' and restructure your entire API. But if you are in a commercial environment where people build on top of your software then you have to decide if you want that people continue to build on your software or if you want to do your own project."*

5 See West/O'Mahony (2005), p. 3

6 See West/O'Mahony (2005), p. 3

2.3. Challenges

Next to the significant advantages of company sponsored projects, West and O'Mahony also point out relevant challenges of such projects. Since they were initiated by a closed organisational entity it may be unclear to outsiders how they can contribute and where they can "make their mark" to the project. Also governance of the project by the company is an important issue. It can easily demotivate external contributors when it's not guaranteed by the company to include all participating parties in the decision making process.[7] Additional challenges mentioned by the project leaders of the investigated OSPs are described subsequently.

2.3.1. Restricted Participation

Depending on the situation the initiating company can restrict participation of external contributors. In the case of eZ systems, the company offers the content management system under a dual license and thus has to secure to maintain the full copyright on the software. Farstad explains the development process and how also external contributors can modify the application despite the restrictions: *"As an external developer you can't say I want to join the project and change this and that, because we do have all the copyright, because we have the dual licensing policy. This can be a disadvantage. Still we have external developers working on eZ publish as well but then they have to give the copyright to us. So to address that problem we have made eZ publish to be a framework. So it is designed to be extended. Normally you can just write an extension or a plug-in or a module that solves your problem. That's what third party developers mostly work on."*

2.3.2. Licensing Issues

As researched by Bonaccorsi and Rossi[8] and pointed out above the licensing issues are mostly solved corresponding the strategy the company pursues with the OSP. Once defined, it is a very risky move to change a project's license again as Hinderink experienced it in other projects: "*The problem is once you've decided e.g. for the GPL and later you change to something like the Zope License, many community members might get angry about it and feel exploited by the central company. There are implications for your business model. If you publish under GPL, build up a community that helps on many different levels and control is only done as quality*

7 See West/O'Mahony (2005), p. 8

8 See Bonaccorsi/Rossi (2003c)

assurance your business opportunities will be much narrower. Basically there remains the possibility to offer the traditional services like documentation, education and so on." Thus it is essential for companies to be informed about the implications of the different open source licenses and when the decision is made to stick to the strategy.

One opportunity for a company to release an OSP but to keep the freedom to change the project's license again is to let all contributing developers sign a committer's agreement thus preserving the full copyright on the software. Boris Kraft, chief software engineer of the software company obinary, explained their motivations of introducing such a agreement for the development of the content management system Magnolia: *"Until they can commit on their own they have to send us patches after signing some committer's license agreement. This is important for us since we want to keep the possibility to change the license. In general we had a lot of licensing questions lately, it's a very complex topic and unfortunately there are only few people who really know about it thoroughly. Simply said we are sort of afraid that if we took an Apache license someone could just take our project away."*

2.3.3. Community Relations

As observed by Osterloh, Rota and von Wartburg participants in open source movement form a norm-based community. They react very sensitively when companies brake the written and unwritten rules of the society and when a critical mass of participants opposes they can even force the company to undertake corresponding moves.[9] So it is always the intent of companies, next to do business with open source software, to keep the community in a good mood to maintain its support.

Hinderink recommends to companies controlling OSPs to clearly state their motives and to communicate transparently to keep its credibility: *"If you want to use open source as a vehicle to make money you have to state this clearly. You have to say exactly whereby and how you want to earn money. In areas where this is not the case and you intend to work together and build a community you also have to communicate this clearly. It's commercial where it is labelled and non-commercial where it is not. If you mix these up you'll lose credibility and respect."*

9 See Osterloh/Rota/von Wartburg (2002), p. 16

Another aspect of community relations concerns the visibility of the company in the entire project. Wechner opinions that for the project it is the best to almost disappear like he succeeded in removing his company's label away from the Lenya project: *"Again another issue is the visibility of the company in the open source project. For example we as a company dissociated ourselves from the project almost completely. Others know that we belong to this company only from our email addresses or because they know us personally. Most of the people just see the project and the Apache Foundation and any connection between the original company and the projects gets lost with time."*

For some firms it is even the initial motivation not to be dominant in the OSP as Wesdorp knows from his company Infrae: *"From the beginning on it was clear that Infrae didn't want to market this as their editor. They wanted it to be an open source project. Actually part of the reason that I started the open source project was because Infrae explicitly didn't want to maintain a What-You-See-Is-What-You-Get editor themselves because it was a huge amount of work."* So this move of Infrae to stay in the background of the project confirms the thesis of West and O'Mahony that the incentive of external contributions raises when there is room for profiling of the participants.[10]

2.3.4. Open up Development and Communication

Looking at failed communication of companies publishing OSPs West and O'Mahony observed so-called "glass house" situations. Outsiders observe the project's ongoing but only participate at the margins of the sponsor's development efforts.[11] The challenge to involve external contributors into the project was also realized by Wechner who observed such behavioural patterns in other OSPs: *"One thing that we did consciously was to open up our development. Initially, the developers came only from our company so we had a lot of internal communication. So to make sure that the outside world was informed about what was going on we consciously decided to communicate via the public mailing list only. So now we practically don't communicate about Lenya internally any more. This is very important because otherwise it would seem like we did something for our private business cut off from the outside world. I've heard of projects where this became a major problem - even*

10 See West/O'Mahony (2005), p. 8

11 See West/O'Mahony (2005), p. 3

in Apache projects. For example the Xalan project, the XSLT processing, was created by Lotus, IBM. So one office was located just in front of the other one and after they went out for lunch together they announced we will do this and that. They are really nice guys, the problem was just that the discussion didn't take place in the open. So if you come from a company or any other closed entity it's important that the communication gets shifted to the outside."

Similarly, Wesdorp perceives the Kupu project since it's unusual that a company sponsored OSP experiences so much participation from outsiders: *"If I look at other projects at Infrae - and I can't really imagine why they do it like that - but usually we don't have much outside developers. People can send patches and that sort of thing but that's not a community-like sort of thing. While at Kupu I really try to emphasize that community thing by letting as much people work on it as possible. But if you have a product which you try to sell you have to take more care having it stable and accessible and consistent all the time. From a community perspective of Kupu we can sort of decide to do strange, crazy things at times and even release that and if it's really bad we can subtract it from the feature set."* So basically, the initiating company Infrae leaves a lot of freedom to the active community members thus motivating them to invest personal spare time into the project.

2.3.5. Distinction Between Company Time and Spare Time

The distinction of the different time frames of employed programmers is another challenge for companies active in OSPs. How much time should company members be allowed to spend for OSP development during working hours? Wechner learned from his experience how to handle this challenge: *"Another important issue is to distinguish between company time and spare time. We had to learn that when we do changes in Lenya for a customer's project of course it's ok to do that during working hours. But if somebody wants to code a features for Lenya without relation to a customer project then it's better done in spare time. Otherwise a mixing of work and leisure time occurs which isn't good. At least this is my opinion."*

3. Business Reasons for OSP Activities

After looking at the nature of company founded OSPs and comparing them to community started projects the following chapter lists various economical reasons for OSP activities. The goal is to show the business motives of companies investing re-

sources into further development and distribution of OSPs. Bonaccorsi and Rossi list various motivations of companies doing activities in the area of OSPs. They distinguish between economical, social and technological motives as shown in table 1.[12]

Motivations of Open Source Firms
Economical
• Being independent of the price and licence policies of the large software companies
• Addressing the new model of software as a consumer-driven service (making money on complementary services)
• Obtaining indirect revenues by selling related products
• Affording innovation (by exploiting the R&D activity of the Open Source community)
• Hiring good IT specialists
Social
• Conforming to the values of the Open Source community (in order not to betray the trust of the Open Source developers)
• Sharing code and knowledge with the community (reciprocating in order to sustain cooperation)
• Thinking that software should not to be a proprietary good (in order to reduce market power of the large software companies)
Technological
• Exploiting feedbacks and contributions from developers of the Open Source community for cutting development costs and improve the software
• Exploiting feedbacks and contributions from the user community for testing and improving software
• Cutting hardware costs
• Promoting standardisation
• Addressing security issues

Table 1: Motivations of Companies to Participate in Open Source Projects, by Bonaccorsi and Rossi

Wichmann on the other hand lists four major motivations of companies investing resources into OSP advancement as shown in table 2. He names standardisation reasons, cost reduction, strategical motives and compatibility enforcement as key motivators for large companies participating in open source development.[13]

12 See Bonaccorsi/Rossi (2003a), p. 14

13 See Wichmann (2002), p. 18

Motivations of Open Source Firms
Standardisation: overcoming the ghost of Unix wars
• Keeping basic operating system up-to-date is costly
• Open source foundation allows focus on other elements of product bundle
• GPL license ensures that foundation cannot be hijacked but influence depends on resources devoted to project
Open Source software as low-cost component
• Product bundles containing open source cheaper than fully proprietary bundles
• Development is to a large extent customer driven
Strategic considerations
• Release of software as open source in nonessential areas
• Releasing software as open source can weaken competitor
Enabling compatibility
• Open source projects to make own software or hardware compatible with open source software
• Viral licenses may force the companies to provide open source software

Table 2: Motivations of Companies to Participate in Open Source Projects, by Wichmann

As companies are always profit oriented institutions these listed motivations by Bonaccorsi/Rossi and Wichmann are all of extrinsic nature. Mainly, two different intentions can be observed: Motivations to obtain strategical advantages and ones to reduce costs. In the first category, continuous development, marketing for the company, know-how of the software's architecture and feedback from the users were found in the conducted interviews. The second category of motivations includes contribution of documentation and the saving of licensing fees of which statements of the OSP leaders were recorded during the interviews. All these motivations are described in detail below forming an incomplete list of six commercial reasons why to initiate and further develop an OSP.

3.1. Continuous Development

When external contributors significantly participate in development of the open source software the company can save resources and invest them in certain areas of the project where they are needed.[14] Also Bonaccorsi and Rossi mention this cost reduction effect of the community.[15]

14 See West/O'Mahony (2005), p. 8

15 See Bonaccorsi/Rossi (2003a), p. 3

Looking at these savings on a long term they result in a strategical advantage bringing open source companies in an overall better position compared to companies working with proprietary software. Bernhard Bühlmann explains this effect describing the highly competitive environment of content management system providers his company 4teamwork is active in with the OSP Plone: *"We're in a way better position than vendors of commercial CMS. For example Obtree got sold twice already, and they have 20 developers. They developed a good product but now Plone is better than Obtree. We've already had a couple of Obtree customers who've switched to Plone. They say Plone is much better and easier to handle. Right now there are about 150 developers worldwide advancing Plone 7 days per week, 24 hours per day. This is a development power only big software companies like Microsoft, Novell or SAP can afford. For the middle range CMS provider it gets harder and harder to survive."*

Also Kraft's company obinary realized this strategical advantage working with open source software and even risked the initiation of a new content management system Magnolia: *"Another goal of course is that we don't have to keep on developing the CMS by ourselves alone so we hope this will change in the future, too. Also we hope Magnolia helps us finding more skilled developers who are motivated by working on a well-known open source project. Such persons don't have to work on a project which nobody will see but can participate in a cool open source project and can also make a name for themselves in the open source community."*

3.2. Marketing for the Company

By spreading the word about a software product its adoption increases and thus the need for professional support raises. So it is one of the main goals of companies starting a new OSP to create a larger market for the software as West and O'Mahony mention in their research paper.[16] Especially in dual licensed software there is the intention to raise the user base and to do marketing for the company by distributing the free version of the software.[17] Also Välimäki observed in his case study about dual licensed projects the freely available release as a marketing tool for the commercially licensed product.[18]

16 See West/O'Mahony (2005), p. 8

17 See Fremy (2001)

18 See Välimäki (2003), p. 14

As mentioned above the market of content management systems is already flooded with software solutions thus starting marketing effort for a new proprietary CMS would not have made economical sense for obinary. Therefore, Kraft decided to initiate an OSP and getting marketing at a much lower price: *"During development we thought about making it proprietary but we also knew that in today's market if we published yet another proprietary CMS we wouldn't succeed. Today you need a huge amount of marketing and sales effort to promote your CMS. Just considering this perspective we couldn't afford these expenses. So one reason to choose an open source license was to get cheap marketing and a large amount of feedback."*

A different strategy was followed by Bühlmann with his company 4teamwork. To promote Plone and increase its penetration in the web application area the company founded a non-commercial institution. His intent was to do marketing for the open source project and thus acquire new customers: *"The intention of 4teamwork always was to spread the word about Plone throughout the nation so we did lots of marketing. We founded SwissZope, an association of companies working with Zope. [...] We also went to the Internet Expo with this label which resulted in new projects. People read some showcases on swisszope.ch and followed the link to our company. So it's not really a community now, only three companies are still on the portal."*

Branding the label of an open source association like the Apache Software Foundation is large help for not yet known projects like the CMS Apache Lenya as Wechner knows: *"This [the Apache branding] has a large influence because today Apache is a real brand. Nowadays you don't have to explain to an IT manager why it's good or bad to use the Apache Web Server because over 50% of all the websites are hosted with the Apache Web Server. This confidence spills over to the other projects. Of course it's not sure that you can sell your project just because of this but at least it's a door opener, you don't have to do much selling any more. They still ask how many committers do we have and if we will become a top level project. [...] But with an Apache project you can make money because it just sells better than others."*

Another way to do marketing for companies is to post a list on the project's website of providers offering services for the OSP as Rothfuss mentions: *"In well-known projects like PostNuke it's helpful for small contributing companies to get listed in a directory as an implementation partner showing their competence in this area."*

3.3. Know-how of Software's Architecture

Knowing the software's architecture enables developers to rapidly accomplish customizations thus giving them a strategical advantage compared to when they would use proprietary software or existing OSPs. This was the main reason why obinary got contacted by Siemens: To do bug fixing on their implementation of the Magnolia CMS. Like this, the expected effect to demonstrate the technological skills of obinary and thus the increasing demand for professional support added up: *"The primary idea is to spread the word about Magnolia and its connection to obinary and simultaneously to show that we are able to handle such technologies. So we hope that people respond and say 'Hey it's a great tool you've created! Can you do our implementation and help us in some parts?' This actually occurred a couple of times and we hope it will continue to grow in the future. Of course, how much really ends up as a paid project we'll see, but before we were just a small software company and now we're an internationally known software provider. This also helps in current customer presentations where we can show what we can do and the customers don't just have to take our word but can see what we've actually achieved."*

The intentions were similar at Infrae when they needed a rich text editor for their CMS. Again, the knowledge about the software was one of the reasons to form a new OSP since the implementation into the existing software was a difficult task as Wesdorp remembers: *"They [infrae] needed a What-You-See-Is-What-You-Get editor. At that point there were some possibilities and we tried some. Most were either very hard to implement or didn't work on the browsers we had in mind like IE and Mozilla. We use Mozilla all the time. After some point when we tried some proprietary and some open source ones we decided to build one for ourselves."*

3.4. Feedback from Users

Receiving feedback of the users is valuable material for software companies since in this way they learn what their customers like the most and where future improvements have the highest benefits for the users. Investigating the database products of the company Sleepycat, Välimäki observed user feedback including bug reporting and feature requests is the main value of the community.[19] Likewise, Farstad experiences the value of the eZ publish community communicating the requested features

19 See Välimäki (2003), p. 11

of the software: *"We ask what are the features you'd like to see in the next version, what are the things we should work on. We take all that feedback and we see what are the most requested features. We do that in more detail every year at the conference. There we discuss the next releases and what we should work on. Specifically, what are solutions we could implement, and so on. That is the real power of the community because you get so much feedback on things you could do. If there is a bug they'll let you know immediately."* As in the Sleepycat case, users also send in bug reports for eZ publish thus saving the company a lot of time of testing and bug searching: *"Most of the bug reporting and the activities on the forum are external. If you ask MySQL they'll say the same. It's very valuable. So if a user reports a bug that is reproducible it saves us a lot of time both finding the bug and fixing it. We can just follow the steps and fix it. That is the biggest value you get from the community. It's not just the code that has value."*

3.5. Contribution of Documentation

Next to code contributions there are various other ways how community members can participate in the project. One way of a community to significantly save company resources is to contribute documentation material of the software and even translating it into other languages. Farstad organized an easy way for participants to work on the documentation and submit it to the repository: *"We had a community project going on for writing documentation. This is a large part when you have a large project. We had a lot of contributions in the documentation area. Users wrote a lot of documentation for eZ publish. [...] You just can log in on our webpage and change the documentation, it's Wiki style. That is a point which helps building the community because then people can contribute without even writing code."* Also, the translation of the software and its documentation is very valuable for eZ systems: *"Our system is also multilingual so you can do translations. [...] We've gotten about 20 translations, all of which are contributed by the community except the Norwegian and the English one."*

3.6. No Licensing Fees

As large software companies usually impose license policies for their products it is an strong economic incentive to use and develop open source software as Bonaccorsi and Rossi conclude in their research.[20] Also for Kraft this was one of the major reasons to develop an own CMS: *"On the other hand marketing alone doesn't help getting the investment back. So the reason for the first Magnolia release was primarily for our company to be able to avoid licensing fees and implement customizings faster."*

20 See Bonaccorsi/Rossi (2003a), p. 3

4. Business Activities

Now company founded OSPs were characterized and also business reasons to initiate and participate in OSP development were explained. To complete this overview of business opportunities of companies involved in OSPs some possibilities how to actually do business with open source software are described starting with a brief overview.

Summarizing the evolution of open source software in only three steps, Bonaccorsi and Rossi first mention the non-profit motivation of developers founding the first OSPs. Then, a hierarchical co-ordination emerged without proprietary rights. And finally, open source software also entered environments previously dominated only by proprietary standards.[21] Thus, the open source movement is now at a point where is economically interesting for companies to offer services and carry out other activities in the area of open source software. Possible business models are described in the next four chapters looking first at the main activity of such software companies, offering services for OSPs. Then a special focus is laid onto extension development since it was mentioned numerous times from different interviewees. Later, the possibility of a partnership program introduction is investigated and finally the model of dual licensing is briefly described.

4.1. Offering Services

In 2002, a survey among 146 Italian companies showed there are various possibilities of firms offering services for open source software.[22] Bonaccorsi and Rossi found 9 different services: Installation, support, maintenance, development of ad hoc solutions, distribution, marketing of software produced by other companies, consulting, training and research and development. Figure 1 shows the distribution of services offered by the investigated companies.

21 See Bonaccorsi/Rossi (2003b), p. 1

22 See Bonaccorsi/Rossi/Giannangeli (2003), p. 12

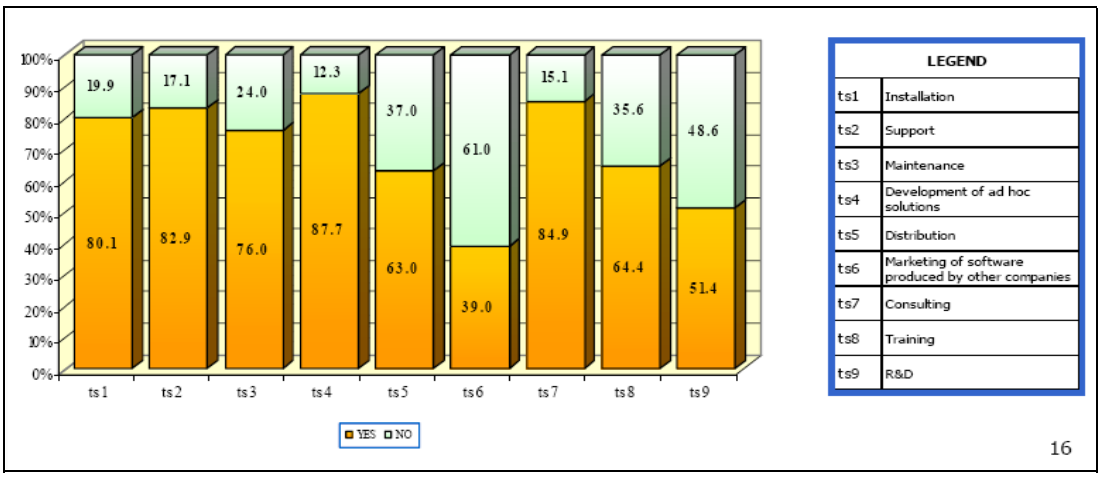

Figure 1: Services Supplied by Italian Firms Active in the Open Source Software Industry

Bonaccorsi and Rossi conclude *"the new business models of Open Source firms address the evolution of software from a commodity to a service driven market."*[23] The consequence of the increasing complexity of software is the raising demand for supporting services thus preparing nutrient soil for business activities. Especially in the open source movement the support of specialized software companies delivering services for OSPs is needed. Various successful software solutions are initiated by a community thus not offering an inherited possibility to obtain guaranteed service. There, companies can step in filling the gap and provide professional support, service level agreements and other contracts giving security to customers.

4.2. Extension Development

To enable extension development, a software must be built up in a modular architecture thus offering programmers to enhance functionality or do customizations without modifying the core of the software. Hinderink knows this powerful possibility is broadly used by developing parties of TYPO3: *"Being able to add new functionality at this level without touching the kernel and also implementing very easily - with the help of the quick starter - is very unique. For developers and agencies the very simple way of customizing is actually the only possibility to distinguish themselves because many use TYPO3 today."* In this way, commercial providers can further develop the program and may even sell the extensions when the licensing scheme is allowing it. Obinary, sole copyright holder of Magnolia, is planning using this opportunity: *"Also we want to offer the possibility to write commercial extensions so you*

23 See Bonaccorsi/Rossi (2003a), p. 14

have to pay for certain ones, like in Eclipse." In Plone there exists already such a proprietary solution enhancing the content management framework to a document management system: *"Recently a Plone desktop has been published, a connection of Plone to Windows. It's a commercial product, one of the first that you have to pay for. Using this you can manage your Plone directories in your Windows Explorer like regular Windows directories. It's like working on a file server. In this way you can use Plone as a document management system, too, not only as a content management system."* And in eZ publish, owners of the proprietary license may also sell their extensions, even over the official eZ publish website: *"Most of the extensions are GPL. But we also have a partner shop where all the partners can present their commercial extensions. Typically it's an integration with an ERP system or other larger things."*

4.3. Partnership Program

Mainly in company founded OSPs the initiating firm is interested in haven partnerships with its company customers. Offering unique services to the subscribers Kraft names the two partnerships obinary offers for Magnolia: *"Now we've introduced a partnership program which gives you a gold partnership for USD 1000 and a platinum for USD 4000 a year."* Also eZ system offers partnerships to its customers, especially they have many agencies regularly working with software as Farstad explains: *"A bit special about the eZ publish community is that we have a very professional community. You don't find this in many other projects. When you go to our forums you'll see most of the users actually are working in companies doing eZ publish. They base all their living on customizing, installing and integrating eZ publish for their customers. We have a large community but it is also a very professional community. It is not so much one person just doing it in his spare time, it's rather companies that are involved in our community. That's quite different from other communities."* Pointing out the possibility for partner companies to create income with eZ publish extensions Farstad describes the smart strategy of eZ systems: *"I know it attracts companies who want to earn money on the solution. And most of those companies join the community as well. That's what we see, there are many professionals who work with eZ publish. This is also part of our profile, we want to make other businesses successful with their products based on eZ publish. I don't think it hurts the community building at all. It just gets new members into your community. [...] We offer our partners to sell their extensions over our website.*

That's our philosophy. You as a partner should be able to make money because this is what you basically want to do as a company. That's what we're trying to do. At the same time we have the openness of the GPL version and the community. We want to support both worlds."

4.4. Dual Licensing

Most literature about open source business models focus on services associated with the OSP. Although already several software companies successfully use the principle of dual licensing only few papers have been published so far. Välimäki writes in his case studies about three popular open source companies, namely Sleepycat Software Inc., MySQL AB and TrollTech AS, how the dual licensing functions, where it strengths and weaknesses lay and under what circumstances an OSP might be successfully managed as a dual licensed software.[24]

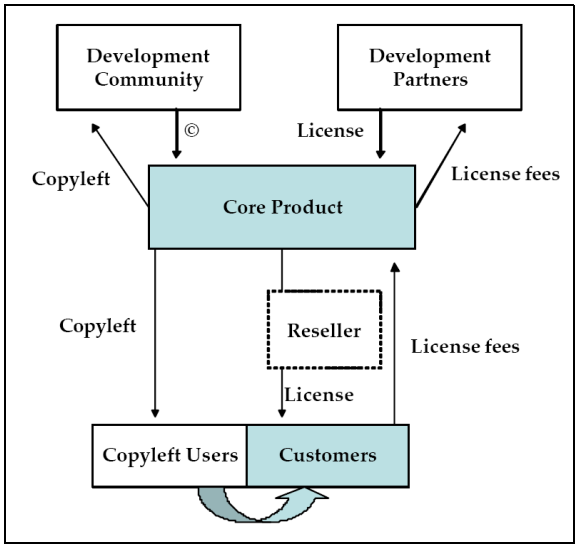

Figure 2: License Streams of a Core Product in a Simplified Dual Licensing Model

Basically, a dual licensed software is published under two different licenses, an open source license and a proprietary one. The major requirement for a company to legally accomplish this parallel publication is the possession of the full copyright of the software. Figure 2 shows the dual licensing model in a graphical way.[25]

24 See Välimäki (2003), p. 3

25 See Välimäki (2003), p. 7

Since the GPL and other open source licenses include a copyleft clause all changes and enhancements of the software have to be checked back into the project thus inhibiting companies from licensing their modifications under proprietary licenses. (shown on the left side of figure 2) When a customer wants to build a third party product and keep the source code closed he has to buy a proprietary license paying the license fees and thus becoming a development partner. (shown on the right side of figure 2)

EZ systems distributes its software with the dual licensing model. Farstad explains what is possible with and without the purchase of a professional license, the proprietary license of the software: *"eZ publish is GPL, so basically you can do whatever you want as long you follow the GPL. But you cannot make distributions, sell it, or make changes and not give back the changes as open source. So for example if you are a company doing system integration and you develop a component which you want to sell then you can buy a professional license and then you can make business on top of eZ publish. But it is exactly the same software, we haven't held back any features in the GPL version. The only difference is, if you have a professional license you can rebrand it and resell it without concerning the restrictions of the GPL."*

5. Conclusions

Although the first criteria of the open source definition,[26] "free redistribution", prohibits selling the software, numerous business opportunities for companies could be shown. Once the challenges of sponsored OSPs are accepted and where it was possible solutions are found, various ways how firms can benefit from their own or from existing OSPs are feasible. The incomplete list of six business reasons showed there are strategical advantages to reach with OSP activity on the long run and also cost reductions on short term. Finally, the actually revenue generating actions like offering different types of services or dual licensing the software identified specific possibilities to build up a successful business based on open source software. To conclude it remains to mention that open source software is free in the sense of availability but not at all in the sense of money thus paving the way for future successful open source companies.

26 See http://opensource.org

List of Figures

List of Tables

Glossary

CMS	Content Management System
OSP	Open Source Project
XSLT	Extensible Stylesheet Language Transformations

Bibliography

[Bonaccorsi/Rossi 2003a]
Bonaccorsi A., Rossi C., Comparing motivations of individual programmers and firms to take part in the Open Source movement. From community to business, Sant'Anna School of Advanced Studies, Italy, October 2003
URL: http://opensource.mit.edu/papers/bnaccorsirossimotivationlong.pdf
[requested: 2005-03-12].

[Bonaccorsi/Rossi 2003b]
Bonaccorsi A., Rossi C., Why Open Source software can succeed, Sant'Anna School of Advanced Studies, Italy, October 2003
URL: http://opensource.mit.edu/papers/rp-bonaccorsirossi.pdf
[requested: 2005-03-13].

[Bonaccorsi/Rossi 2003c]
Bonaccorsi A., Rossi C., Licensing schemes in the production and distribution of Open Source software. An empirical investigation, Sant'Anna School of Advanced Studies, Italy, October 2003
URL: http://opensource.mit.edu/papers/bnaccorsirossilicense.pdf
[requested: 2005-03-14].

[Bonaccorsi/Rossi/Giannangeli 2003]
Bonaccorsi, A., Rossi, C., Giannangeli, S., Adaptive entry strategies under dominant standards: Hybrid business models in the Open Source software industry, Sant'Anna School of Advanced Studies, Italy, March 2003
URL: http://opensource.mit.edu/papers/bonnacorsirossigiannangeli.pdf
[requested: 2005-03-13].

[Fremy 2001]
Fremy, P., Interview: Trolltech's President Eirik Eng, September 2001
URL: http://dot.kde.org/1001294012/
[requested: 2005-03-13].

[Osterloh/Rota/von Wartburg 2002]
Osterloh, M., Rota, S., von Wartburg, M., Open source - new rules in software development, University of Zurich
URL: http://www.iou.unizh.ch/orga/downloads/OpenSourceAoM.pdf
[requested: 2005-03-13].

[Stürmer 2005]
Stürmer, M., Open Source Community Building, Licenciate at the University of Bern, March 2005.

[Välimäki 2003]
Välimäki, M., Dual Licensing in Open Source Software Industry, in: Systemes d'Information et Management 1/2003
URL: http://opensource.mit.edu/papers/valimaki.pdf
[requested: 2005-03-13].

[West/O'Mahony]
West, J., O'Mahony, S., Contrasting Community Building in Sponsored and Community Founded Open Source Projects, to be published in the Proceedings of the 38th Annual Hawai'i International Conference on System Sciences, Waikoloa, Hawaii, 2005
URL: http://opensource.mit.edu/papers/westomahony.pdf
[requested: 2005-03-13].

[Wichmann 2002]
Wichmann, T., FLOSS Final Report – Part 2, Free/Libre Open Source Software: Survey and Study, Firms' Open Source Activities: Motivations and Policy Implications, Berlecon Research, July 2002
URL:http://www.berlecon.de/studien/downloads/200207FLOSS_Activities.pdf
[requested: 2005-03-12].