Leveraging open source licenses and open source communities in hybrid commercial open source business models

Author. Karl Michael Popp.

SAP AG
Corporate Development
Dietmar-Hopp-Allee 16
69190 Walldorf
http://www.drkarlpopp.com
karl.michael.popp@sap.com

There are many books and articles about open source business and open source business models. The goal of this paper is to provide and describe commercial use of open source software, how this use is embedded in hybrid commercial open source business models and how these business models leverage open source licenses and communities to create value for customers and software vendors.

1. Approach, overview and contribution of this paper

The general research idea behind this work is that a framework can be created to describe commercial open source business models as combinations of different criteria. Goal of this paper is to find and describe the criteria by answering the question how software vendors leverage open source software and open source communities as well as multiproduct and multilicense strategies in commercial open source business models. We start with showing how open source software is used to create business models for software vendors, how this creates value for the vendors as well as customers and that the corresponding business models are hybrid business models.

Based on literature and own research, an overview of commercial use of open source is presented, which is followed by a definition and classification of open source business models. This classification can be used by CEOs and business developers to create new business models. The paper closes with showing the value of these business models for software vendors and customers and a summary.

2. Commercial use of open source software

In the software industry there is a lot of activity around open source, like investments in open source projects and increasing use of open source software overall [1][2]. Accepting the co-existence of open source software and commercial software, the question arises how software vendors leverage open source in their business models. This paper tries to provide an overview of commercial business models leveraging open source software and open source communities, no matter if the company participates in open source development or not. For a commercial software company, open source software is either software that is licensed to that company under an open source license or software that is provided by that company to customers under an open source license [3], [4].

1.1 Open source licenses as a key factor for the variety of business models

An open source license comes with rights and obligations and the search for the optimal license continues [1], [3]. The license creates limitations in creating business models around open source software.

For example, a company using open source software as part of its products, the limitations can be described as follows.

A software vendor may make use of the rights, like usage or redistribution of the open source, but it also has to fulfill the obligations, like delivering the copy of the license text with the software or revealing the source code of a software product.

The key point for a commercial company is if it is willing and able to comply with the open source license terms. So the rights and obligations have to be analyzed diligently to make sure there is no violation of the license terms and the license terms are not in conflict with the commercial company´s business model [4], [5]. If this is ensured, the company can leverage this open source software.

Another restriction is, that some licenses do not allow modifications of the software. This would exclude the ability of a commercial open source company to provide maintenance, because the open source code must not be changed.

But the limitations of open source licenses can also be an advantage for software vendors providing open source software, which will be shown later, when we talk about dual licensing.

1.2 Suppliers of open source software for commercial use

In a simplified view, open source software can be supplied by a community or by a commercial company [6]. We speak of **community open source** and **commercial open source** respectively.

Community involvement with open source products means that a community of people provides creation, maintenance and support for an open source software [7]. Sometimes the community even provides presales and sales activities for companies offering an open source version and a commercial version of their software. In most of the cases the community provides these services free of charge.

By providing an open source licensed version of a product, a software vendor has the opportunity to outsource certain activities, like development, maintenance and support, to the community [8].

There are, of course, differences between a company and the open source community in providing open source software. These differences are important to understand, because they influence a customer´s software license selection and they also create niches for companies to establish a business. The differences are listed in Figure 1.

Community vs. Commercial Open Source

	Community Open Source	Commercial open source
Example	Eclipse	SuSe Linux
License	Open source license, standard terms only	Commercial license, customized terms possible
Consulting	Some help by community, free of charge	Paid consulting services to customer needs
Main-tenance	Community provides new versions	Paid maintenance to customer needs
Support	Community supports without guaranteed service level agreements	Paid support to customer needs with guaranteed service level agreements

Figure 1: Commercial open source vs. community open source

Community open source is usually offered under an open source licenses with standard terms that apply to all users of the software. If somebody chooses to use the software, consulting is usually provided by other members of the community free of charge. For community open source, maintenance and support is done by the community member. For the support provided by the community, there is usually no definition of service level agreements.

The situation for commercial open source is different in many ways. The license usually is a commercial license, which can have standard terms, but may also have terms specifically customized to fit the needs of a single customer. Consulting will be provided by a professional service organization. Maintenance will be provided based on a commercial maintenance and support contract, which will explicitly specify the service level agreement for support services.

2.Classification of open source business models

Based on a general classification of business models [9] we will have a look at open source business models. The following section arguments along the lines of [6].

2.1 Classification of open source business models

Figure 2 shows a classification of generic business models. The business models relevant for commercial open source business are marked in bold. In this general classification of business models, software classifies as an intangible product, see the corresponding column “Intangible” in Figure 2. Software can be created or written

("Inventor"), distributed ("IP Distributor") or licensed or rented to customers ("IP Lessor"). In addition, the customer needs services to run and maintain the software, like implementation, support and maintenance services. These classify as "Contractor" business [6]. We assume here that all open source businesses make use of at least a subset of these four business models. No matter if it is a community or a commercial software vendor, one or many of these business models are applied. By choosing a specific selection of business models, a so-called **hybrid business model** is created. Creating a hybrid business model means combining different business models with their specific goals, requirements and cost structures.

Since these business models are models on a type level, there might be different implementations of how a certain business model is run. An open source community might run the Inventor business for creating software in a different way (leveraging the community) than a commercial software vendor (leveraging a proprietary development team), from a process as well as from a resource perspective. But on a type level, both run the same type of business called Inventor.

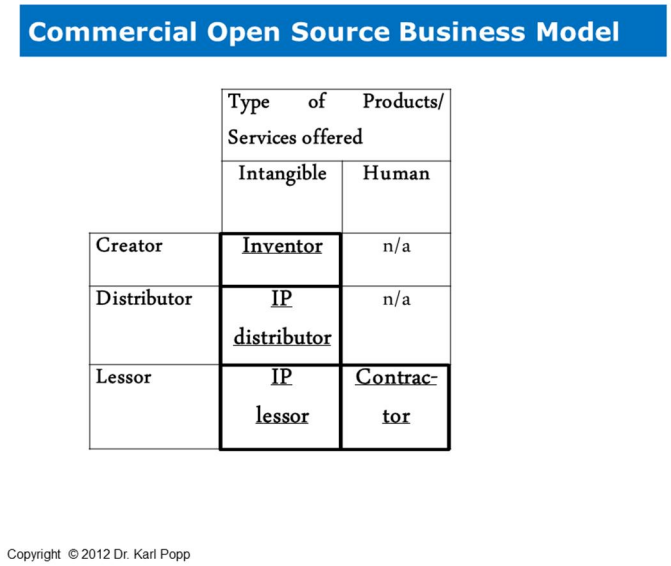

Figure 2: Commercial open source business model

It is important to note that business model type level and business implementation level are design dimensions for describing existing and designing new open source business models [10]. So creating a new open source business might start with selecting one or more type level business models and then select from existing or new implementations for each of the business models to create a business.

Going forward, we will analyze existing commercial and community open source business models as a selection of a subset of the business models identified here: Inventor, IP Lessor, IP distributor and Contractor.

2.1.1 Community open source business model

The open source community business model usually makes use of the following business models: Inventor, IP Lessor and Contractor.

For the community, the **Inventor business** is what the community is most involved in. It is about creating open source software and engaging with the community members to coordinate the work and collect the contributions of the community members.

The **IP Lessor business** is also important for the community. The IP lessor business defines the terms and conditions of the open source license and makes the software available to customers. The license is defined by the community and all customers using the software have to comply with it. In some cases, there are multiple different licenses for an open source software that a customer can choose from.

The **Contractor** business contains all human services to customers. The community typically provides these via email and they contain services like maintenance, support, translation for country specific versions and the like. They are all carried out by community members. In almost every case, the customer does not pay for these services, but the customer has no rights to enforce any of these services and he does not have service level agreements, like a definition of minimum answer time for support incidents.

The community can serve two types of customers: software vendors and (end) customers. For software vendors, the open source community works as a supplier of software, for the customer, the open source community works as a software vendor licensing software to the customer.

These two relationships differ in the way that customers and software vendors might make use of the software. Customers usually license the software for internal use only. Software vendors license software for internal use and/or for distribution to customers. Often open source software is included in commercial software and provided to customers by the software vendor. In this case, the software vendor has to make sure he complies with all licenses of all open source software he is including in his software product.

2.2 Commercial open source business models overview

In the last section we described the community business model, now we turn to the commercial open source business model. As mentioned before, a commercial software vendor does not have to implement all of these business models, but can rather build a unique business model by selecting a subset of available business models. One basic difference to community open source is that the IP Distributor business model is an option for commercial companies.

The history of commercial open source companies shows that in the beginning the companies focused on services around open source software, which matches the Contractor business.

The next step was to build distributions for open source software, like e.g. for Linux. This matches to the IP Distributor business model.

Today, we find all kinds of hybrid business models around open source. Companies are building software and donate it, completely or partially to the open source community (Inventor business model) [11]. Commercial software vendors often package or change or extend existing community open source software, so the community acts as a supplier of open source software to the software vendor. In some cases the software vendor does not use existing open source software from a community, but chooses to offer its proprietary software under a dual licensing strategy, e.g. under a commercial and an open source license.

Please note that there are at least two delivery models for open source software: either the software is distributed to the customer and run at the customer´s site or the software is provided in a hosted/on demand delivery model.

2.2.1 Commercial services for open source

Since open source licenses are free of charge, many commercial companies first and foremost focused on providing services around open source software [10]. The expectation was simply that customers would still need services and since the license was free, that customers would have more money to spend on services.

Commercial open source companies provide the following services for open source software: Hosting, Maintenance, Support, Consulting and Extension or adaption of open source software to a customer´s needs.

Hosting services mean providing hardware and access to that hardware running open source software. Maintenance services consist of the following activities: building future versions, bug fixes and upgrades and providing them to the customers. Support services contain of accepting, maintaining and resolving incidents that the customer has while using the software. Consulting services mean planning and executing the installation and go-live of customers´ system landscapes containing the software.

Extension or adaption of open source software based on customer´s requests is designing, programming, testing and delivering open source software that has been modified or expanded. Examples for extensions and modifications are:

- Functional Extensions for open source applications with country-specific functionality or customer specific functionality;
- Extending the usage scenarios for open source to additional countries by adding additional translations of user interfaces;
- Adapting open source software, e.g. to make modifications of open source software to run on a currently unsupported hardware platforms.

2.2.2 Commercial licensing business for open source

In the industry, we see three ways how commercial open source companies offer software to customers executing the IP Lessor business model:

- Offer or redistribute open source software only, no commercial software offered. In this case, the software vendor needs a hybrid business model containing one or more revenue streams to fund the open source business.
- Offer identical products under two licenses (dual license model) [12].
- Offer different versions of the same product under two licenses to customers (dual product model).

For a commercial open source company, there are two choices for dual licensing: dual license strategy for identical products or dual product strategy with dual licenses, which will be explained below.

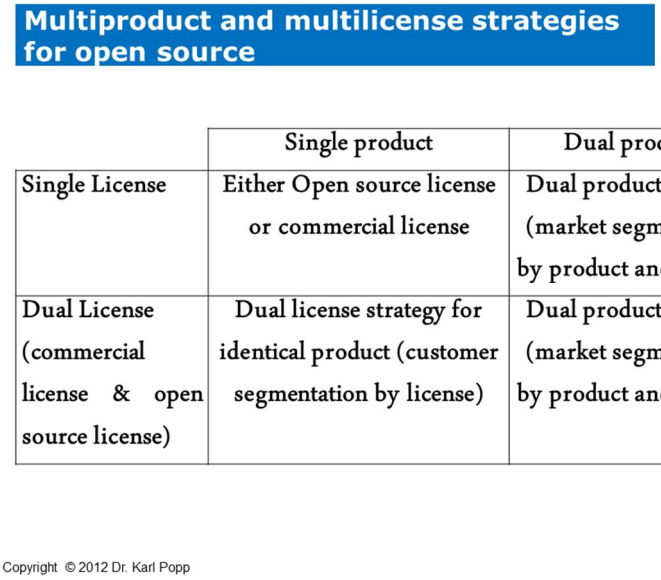

Figure 3: Multiproduct and multilicensing strategy for open source.

Dual license for identical products

Following the **dual license strategy for identical products** [12] a commercial open source company would offer a single product under an open source license and under a commercial license. There are good reasons for the company and customers alike to have the choice between the two licenses As mentioned earlier, customers could choose the commercial license for several reasons, like to ensure they get support service level agreements, warranty or liability from the software vendor. The commercial open source software vendor could use a license that does not allow commercial use of the open source software to force commercial users to buy a commercial license.

Dual product with dual licensing

In the **dual product strategy with dual licensing**, where two similar products are offered under two different licenses, the software vendor usually applies versioning. This could mean that a product with limited functionality can be licensed under an open source license and the full product is available under a commercial license.

There are basically two examples for this strategy, freemium [13], [14] and customer specific version of open source under commercial license. Freemium in the context of open source means that a free version of a product under an open source license exists with restrictions compared to a commercial version of the product, like e. g. a reduced set of functionalities. The customer has to pay a premium, a commercial license fee, to get the full version of the product.

3. Value of commercial open source for customers and vendors

Let us take a look at how value is generated for software vendors offering and customers using commercial open source software.

3.1 Customer view: Value of commercial licenses for open source software

Commercial open source vendors offer open source licensed software to their customers. There are different ways software vendors can add value to the open source software like:

- By packaging [15]: the software vendor creates a distribution by shrinkwrapping open source software and distributes that to the customers (IP distributor business model). The customer can rely on professional configuration of the package and does not have to have expert knowledge on open source.
- By providing a commercial license [16] with significant differences to the open source license, like warranty and liability, no copyleft effect, clearly stated usage scenarios of the software and others. So the customer gets some license (or contract) terms that he could not get in the open source license.
- By creating customer specific adaption or integration of open source software as commercial software. When a customer needs Perl on an exotic hardware platform with 64 bit support, he will contact a

company that is specialized in this business and order that specific adaption of the open source software under a commercial license.

- By omitting advertising in the commercial version of the software while the open source software is containing advertising.
- By providing better service level agreements, more storage space or other features for a higher service fee. This case applies for open source software in a hosted or on demand delivery model.

But even without the extra value a customer might decide against an open source and in favor of commercial open source. This is the case, if e.g. a customer needs customized license terms, runs open source in a mission-critical environment and thus needs service level agreements in support or if he needs maintenance provided in a different way than via the open source community. In many business contexts it makes also sense to have liability and warranty provisions from a supplier when using open source. In most of the existing open source licenses there is exclusion of any warranty or liability [3]. This is another reason why companies might choose commercial open source over community open source.

3.2 Value of commercial open source software and communities for software vendors

Knowing about the availability of open source software and open source business models, the question arises, how value is generated for software vendors not only by leveraging the "free" open source software, but also by leveraging the open source community.

3.2.1 Leveraging the open source community for commercial purposes

Besides providing open source software to customers, software companies can leverage open source and the open source community for their business in the following ways [6]:

- Leverage the open source community as supplier, as development resource, sales, maintenance or support resource.
- Leveraging the open source community as product owner, maintainer and supporter and
- Leveraging the open source community as sales channel.

To create a commercial open source business model, software companies choose one or several of these levers. This is why there is no single open source business model out there. Let us look closer at the different advantages of a commercial open source model.

3.2.2 Leverage the open source community as a supplier

Software vendors often use the open source community as a supplier of software. Almost any commercial software on the market contains components that are under an open source license or the solutions use open source software as a runtime environment. The main reasons to use open source "as supplied material" are quality and cost advantages. Quality advantages have been shown by several studies in the following way: open source software with a community of significant size has a higher quality than similar commercial software.

Regarding cost advantages: If the community is inventing the software, it carries the cost of development. There is no sunk cost for a company to develop the software.

So for the software vendor, open source comes for free and provides a significant cost advantage compared to programming a similar, proprietary functionality from scratch. The software vendor might also include and ship the open source software with its solutions. As mentioned before, there is no license fee paid from the software vendor to the open source community. As every open source software comes with license terms, the software vendor has to conform to the license terms of the open source software used [17].

If the software vendor ships the open source software, the software vendor is responsible for support and maintenance of the software shipped, which includes the open source software. So the software vendor has to make sure he is able to maintain and support the open source software.

Since compliance with the license terms of open source software used is important, help for doing that is available. One company specialized on analyzing open source usage and on analyzing the attached license terms is Black Duck Software [18]. They offer a tool that automatically analyses the source code and determines the open source software used. Then you can determine if you can comply with the license terms and if you want to continue to use the open source software.

3.2.3 Leveraging the open source community as product owner, maintainer and supporter

The software vendor might decide to donate the source code to the community and let the community drive the product innovation as well as maintenance activities. By doing that, the software vendor further lowers its cost to develop and maintain software.

Commercial software vendors might have different reasons for donating software to the open source community, e.g.:

- To create visibility of a software company´s expertise. This is especially interesting for small companies to gain visibility and reputation within a larger community of subject matter experts.
- To get rid of the cost for product development, maintenance and support if a product is in the late stages of the product lifecycle or if the product is commoditized or did not create enough revenue.

But how do you make the community owner of a product? You donate proprietary software to the community and make it open source. The following picture shows the flow of products. From this point in time, you lose some control over the product and trade this for the cost saved and the innovation speed of the product.

At the same time, if the community is big enough and active, the quality of the software increases. There is also a good chance that the community, due to its heterogeneity, is a better breeding ground for evolutionary innovation of the product.

3.2.4 Leveraging the community as sales channel

Software vendors might leverage the community to endorse products via viral marketing. It works like this: community members like the software and endorse its use (at companies). The companies can choose between the open source and the commercial license of the software. If it chooses the commercial license, it will be provided by the software vendor.

Depending on the open source license chosen, the software vendor can force customers into a commercial license for commercial use. This is the case for open source licenses, which do not allow commercial use of the software under the open source license. Another case is the customer wanting to do proprietary changes on the software and keep the ownership of the software, which conflicts with copyleft licenses. Copyleft licenses enforce that all versions, including modified and extended versions, are available for free to the community.

4. Summary and outlook

Here it was shown that open source software is used by commercial software vendors, how open source software creates value for the vendors and customers. Based on a modeling concept for business models it was shown that hybrid business models are used and that these business models can be combined to create different hybrid business models of commercial open source companies. In addition, an overview was given, how commercial open source companies leverage open source software and open source communities for their purposes.

Goal of this paper is to find and describe the criteria to be used to describe commercial open source business models. Several proposed criteria were presented: the business models in use, the way the commercial company leverages licensing strategies and the way the company leverages open source communities for its commercial purposes. Further research will focus on the hypothesis that the set of possible open source business models is the Cartesian product over the sets of combinations of business models, the set of multilicense and multiproduct strategies and the set of ways to leverage the open source community.

The evolution of open source business and commercial open source business is still underway. We will see in the future, which new hybrid business models will be created, just like the ones we recently saw emerging in open source on demand applications or open source software in cloud environments.

References

[1] J. Lerner and J. Tirole, “Economic Perspectives on Open Source,” in *Perspectives on free and open source software*, vol. Volume 15, J. Feller, B. Fitzgerald, S. Hissam, and K. Lakhani, Eds. MIT Press, 2005, pp. 47-78.

[2] A. Deshpande and D. Riehle, “The Total Growth of Open Source,” in *Open Source Development Communities and Quality*, 2008, vol. 275, no. December 2006, pp. 197-209.

[3] T. Jaeger, O. Koglin, T. Kreutzer, A. Metzger, and C. Schulz, “Die GPL kommentiert und erklärt,” in *Die GPL kommentiert und erklärt*, Institut für Rechtsfragen der Freien und Open Source Software, O’Reilly, 2005, pp. 1-24.

[4] A. Onetti and S. Verma, “Open Source Licensing and Business Models,” *ICFAI Journal of Knowledge Management*, vol. VII, no. 1, pp. 68-95, 2009.

[5] S. Krishnamurthy, “An Analysis of Open Source Business Models,” in *Source*, vol. 54, no. February, J. Feller, B. Fitzgerald, S. A. Hissam, and K. R. Lakhani, Eds. The MIT Press, 2005, pp. 267-278.

[6] K. Popp and R. Meyer, *Profit from Software Ecosystems: Business Models, Ecosystems and Partnerships in the Software Industry [Paperback]*. Books on Demand, 2010, p. 242.

[7] M. Stürmer and T. Myrach, “Open source community building,” *Lutterbeck Bernd et al Open Source Jahrbuch*, pp. 219–234, 2006.

[8] L. Dahlander, M. G. Magnusson, Jürgen Bitzer, and Philipp J H Schröder, “Business Models and Community Relationships of Open Source Software Firms,” in *The Economics of Open Source Software Development*, Elsevier, 2006, pp. 111-130.

[9] T. W. Malone et al., “Do Some Business Models Perform Better than Others?,” *Social Science Research*, no. May, 2006.

[10] J. Lindman and R. Rajala, “How Open Source Has Changed the Software Industry: Perspectives from Open Source Entrepreneurs,” *Technology Innovation Management Review*, no. January, pp. 5-11, 2012.

[11] J. Henkel, “Open source software from commercial firms–tools, complements, and collective invention,” *Zeitschrift Für Betriebswirtschaft*, vol. 4, no. 4, pp. 1–23, 2004.

[12] M. Välimäki, “Dual Licensing in Open Source Software Industry,” *Business*, vol. 8, no. 1, pp. 63-75, 2003.

[13] K. J. Bekkelund, “Succeeding with freemium,” NTNU, 2010.

[14] N. Pujol, “Freemium : attributes of an emerging business model,” *Organization*, no. December, 2010.

[15] A. I. Wasserman, “Building a Business on Open Source Software,” in *Cases in Technological Entrepreneurship Converting Ideas into Value*, Edward Elgar, 2009.

[16] D. Ascher, “Is Open Source Right for You?: A fictional case study of open source in a commercial software shop,” *Queue*, vol. 2, no. 3, pp. 32-38, 2004.

[17] M. A. Cusumano, *The Business of software*, vol. 44, no. 3. Free Press, 2004, pp. 15-18.

[18] C. E. Bagley and D. Lane, “Black Duck Software,” 2006.