IT Innovations and Open Source: A Question of Business Ethics or Business Model?

Markus Nüttgens

1 Best Practice for Free?

The internet makes data become an increasingly transient good. Almost any song and even entire movies are available in a digital form – not always to the delight of the copyright owners. Another type of good, which can be considered as predestined to be distributed over the channel *internet*, is software. Vendors use the internet for direct sales purposes as well as for making software explicitly available free of charge, what offers interested people the possibility to test preliminary software versions *(beta versions)* and to report possible software errors *(bugs)* to the vendor. This concept of distributed quality assurance isn't very innovative. The idea of Open Source even incorporates the integration of all involved people and all parts of a software product's lifecycle via an open license model (DiBona et al. 1999, Feller et al. 2005).

To put it in a nutshell, the core idea of Open Source is that users do not only participate in the testing process of software but also actively improve the software itself by performing changes on the source code level. To give users the possibility to do so, the source code of Open Source software is often freely available over the internet – this can be considered as the very opposite of common software business models. This paradigm gives the opportunity to incorporate the skills and experiences of software developers all over the world to improve the quality of a software product and to extend it with new features. Several experiences of the last years have shown that such collaborative and open software development processes can decrease the amount of software bugs to a much higher extent compared to non public source code products. It is important to mention that the term Open Source does not mean that the usage of such software produces no costs – the biggest part of the Total cost of Ownership (TCO) is still given by the rollout, user training and support.

The example of the prominent operating system Linux shows that Open Source has finally changed from an idealistic idea to a threat for commercial software providers. One of the currently most successful open software projects is the "Apache" web server that occupies the top position among the most used web servers (Apache 2006). From about 76 million web server installations world wide, in February 2006, the Apache web server had a market share of around 68 percent (Netcraft 2006).

Even governments begin to think about the usage of Open Source software for public service purposes more often. One driver of such thoughts might be the suspicion that sensitive data could be accessible for third parties, if commercial software is used. A famous example that impressively underlines that the publication of a software's source code is no security risk at all is given by the encryption software "Pretty Good Privacy" (PGP) (PGP 2006). In this case, the accessibility of the source code greatly improves the security level of the software by offering third parties the possibility to take an insight into the software to look for possible problems.

These developments won't leave the consulting business unaffected. To give customers strategic advices for the design of information systems and to consult a customer on the operative level during the rollout of solutions, the competence portfolio must be extended accordingly. Thereby the consulting companies decrease their dependence from single software vendors. Above this, the possibility to proactively develop specific solutions arises. If this development is performed together with the customer, a new dimension of customer loyalty emerges.

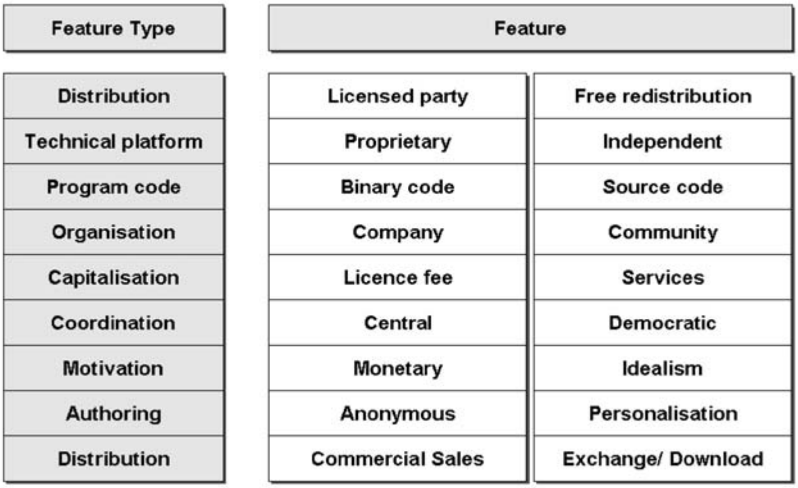
In the following section the concept of Open Source will be explained shortly (for a detailed introduction see Nüttgens and Tesei 2000). After that, upcoming action options for software companies and consultants are introduced and discussed with respect to the Open Source concept.

2 What is Open Source?

Open Source refers to the free availability of software source code to use and change it according to ones personal needs. This approach at first seems to be the total opposite of classical software sales models that normally protect the software source codes to avoid an uncontrolled spreading of the software.

Currently the business models for Open and Closed Source appear to be contradictory. Figure 1 compares the core aspects of these business models. While traditional business models tend to match the left column, Open Source business models are more likely to match the right one.

In the following, the terms *free software* and *open software* are used synonymously for Open Source software. They express that software can be used, copied and distributed by anyone. This might be done free of charge or not – but in any case the software source code must be made available to every interested person. Open Source software is not always free of charge per definition. Fees can arise for distribution services for example



Feature Type	Feature	
Distribution	Licensed party	Free redistribution
Technical platform	Proprietary	Independent
Program code	Binary code	Source code
Organisation	Company	Community
Capitalisation	Licence fee	Services
Coordination	Central	Democratic
Motivation	Monetary	Idealism
Authoring	Anonymous	Personalisation
Distribution	Commercial Sales	Exchange/ Download

Figure 1: Characteristics of software markets

but generally not for licensing issues. Proprietary software can be considered as the opposite of the Open Source concept. The usage, distribution or modification of proprietary software generally requires an approval by the copyright owner.

2.1 The Genesis of Open Source

Ever since the very beginning of software development in the 1960s and 1970s, programs were shared among development teams for reading, changing or reusing purposes (software sharing communities). In 1982 the companies IBM, HP and DEC released commercial versions of the UNIX operation system for their own hardware. Members of various research teams (from universities or even other companies) were recruited and they worked on commercial software to a greater extent from then on. The simple communication infrastructure started to become increasingly insufficient and so the software sharing communities slowly started to dissolve. A vacuum originated within the software production area that motivated companies to fill it with commercial software. This development resulted in the decrease of so called free software. Companies and research institutions decided so use proprietary software to a greater extent. The technological development supported this trend as operating systems were often directly dependent to hardware so that all processors needed their own proprietary operating system very soon.

For reanimating the cooperative spirit of the software sharing communities, the former MIT employee Richard Stallman initiated the GNU (GNU's not UNIX) project and founded the Free Software Foundation (FSF) (see Stallman 2006; FSF 2006). At that time, the project goal was to develop a free and open UNIX operating system (Müller 1999:17). Although a UNIX kernel was never realized, the huge set of free software systems and development tools from the GNU project enabled the development of Linux in the early 1990s. A student named Linus Torvalds started to work on a free UNIX kernel for personal computers with Intel 80386 microprocessors. His rapid success attracted other developers over the internet who started to support him in his efforts to develop the operating system Linux as the first full functional free UNIX. At this time Linux wasn't noticed by many developers. It took another 5 years until the idea of Linux was broadly perceived. At the end of the 1990s an increasing number of developers began to focus on the internet and on Linux. The World Wide Web (WWW) finally made the internet become a mass medium whereby the amount of potential new developers (metaphorically spoken) exploded. Companies that sold proprietary UNIX (like HP, DEC or IBM) had a quite bad marketing at that time. So Microsoft was able to gain huge market shares for its Windows operating system with aggressive marketing campaigns. By 1994 Linux had reached a solid stability level and had become prominent as a development platform. The available programming languages and tools encouraged developers to start new projects for Linux respectively to port already existing projects to this platform. Because universities and research institutions were very interested in Linux, more and more projects (in the fields of programming languages, databases, graphics or desktop for example) were initiated. As the number of fields that were addressed by free software increased, the whole free software community grew accordingly. This also led to an increasing amount of special interest groups. As a consequence of this upsizing, there were a lot of different opinions of useful definitions of the term "free software" soon. There have always been various different licenses for free software and by the engagement of companies like Netscape, Troll Tech or IBM several new ones were added.

The term Open Source became prominent by the Open Source Initiative (OSI) – a non-profit organization with the goal to introduce the spirit of open source to a broader public. The public success of this term was reflected by many articles about Linux and about the publication of the source code of the Netscape browser that were published in 1997 and 1998. On February 23rd 1998 the Netscape Company finally announced that they will officially use the term Open Source. Soon companies like Corel, Sun Microsystems, IBM, SCO, HP, Oracle, Informix and SAP followed by making similar announcements. Nowadays Open Source and the idea behind it is established and noticed by a broad public. The mentioned characteristics of Open Source software are subsumed in the Open Source Definition (OSI 2006a). As the appellation Open Source is descriptive, it can not be registered as a trade mark. Because of the free software community's need for a reliable identification of open source software, the OSI introduced the certification mark “OSI certified” (OSI 2006b). If software is labeled with this mark, the OSI certifies that the software is distributed under a license that is conformable to the Open Source Definition. A discussion of the currently most common Open Source licenses can be found in St. Laurent (2004). As licenses are no goods, they can not be registered at the U.S. Patent and Trademark Office as well. An OSI certification is performed in two steps:

The OSI publishes a list of licenses that are conformable to the Open Source Definition. If a new license should be added to this list, it has to be submitted to the OSI for a discussion. New licenses are accepted if the subscribers participating in this discussion have no objections.

To mark software with the “OSI certified” mark, the software must be distributed under a license that can be found on the OSI approved licenses list (OSI 2006c).

2.2 Development and Distribution

The Open Source approach is based upon certain development and distribution models that are briefly explained in the following. Until 1997 the development process of free software was practically not documented and solely apparent from known projects like Linux. In his essay “The Cathedral and the Bazaar”, Raymond tried to describe factors of a successful open source development process for the first time. According to this essay, the open source development model is also known as the “Bazaar method”. In the latest version of his essay, Raymond analyses the way the development of Linux took place and extracts the following rules for Open Source development processes out of it (Raymond 2000):

1. Every good work of software starts by scratching a developer’s personal itch.
2. Good programmers know what to write. Great ones know what to rewrite (and reuse).
3. “Plan to throw one away; you will, anyhow.” (Fred Brooks, “The Mythical Man-Month”, Chapter 11).
4. If you have the right attitude, interesting problems will find you.
5. When you lose interest in a program, your last duty to it is to hand it off to a competent successor.
6. Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging.
7. Release early. Release often. And listen to your customers.
8. Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.
9. Smart data structures and dumb code works a lot better than the other way around.

10. If you treat your beta-testers as if they're your most valuable resource, they will respond by becoming your most valuable resource.
11. The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.
12. Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.
13. "Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away." (Antoine de Saint-Exupéry)
14. Any tool should be useful in the expected way, but a truly great tool lends itself to uses you never expected.
15. Provided the development coordinator has a communications medium at least as good as the internet and knows how to lead without coercion, many heads are inevitably better than one.

On the basis of this development model, a complementary distribution model for open source products was established. Developers provide their software on the internet. Interested users can search for software on their own, test and use it. Also users can get into direct email contact with the developers and thereby participate in the development of the software. Another possibility is given by the distribution of software bundles. Contrary to proprietary software, where customers have to pay license fees, fees for free software only arise for the service of providing. This means that free software that is acquired once can be freely passed down to others. The most common distribution types are (1) the download from the internet, (2) the ordering and shipping of CDs or DVDs and (3) the free give away of CDs or DVDs as attachments to magazines.

3 Strategic Options for Consultancies

Open Source challenges the development of proprietary software and consecutive *classical* consulting services while it simultaneously provides big chances for strategic reorientations. Figure 2 shows a strategic action framework for IT related consulting services in the open source context. The dimensions of this framework are (1) the consulting focus (industry oriented vs. software oriented) and (2) the chosen development approach (Open Source vs. proprietary software). While "industrial orientation" means that the provided services of a consultancy focus on a certain industrial sector (e.g. retail or banking), "software orientation" means that the services are aligned with a specific software application and its feature range. The characteristics of the development approach dimension are self describing.

Each of the four fields of this framework is related to a particular kind of consulting service:

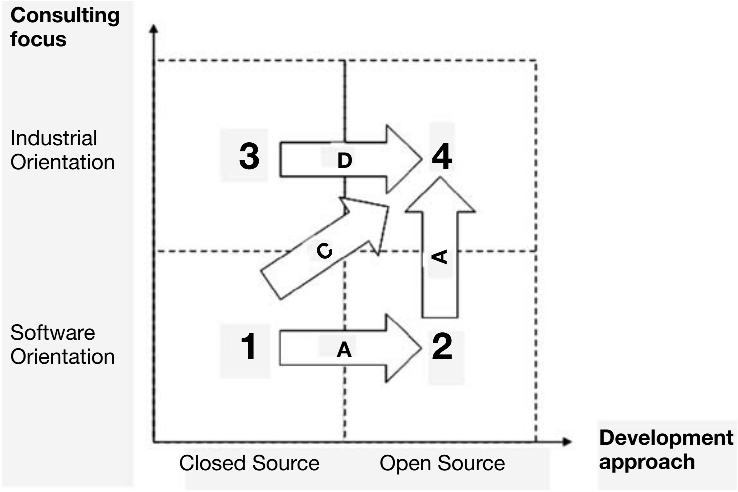


Figure 2: Strategic action framework for IT related consulting services

Field 1
Description: Introduction and customizing of proprietary software systems.
Providers: IT consultancies in the area of office and ERP systems.

Field 2
Description: Introduction and customizing of Open Source systems.
Providers: Companies that emerged from the Open Source community. These companies test, document and distribute Open Source software. Consulting services offered include introduction, user training and technical support.

Field 3
Description: Development of technical-organizational solutions by the usage of proprietary software systems.
Providers: Process consultancies with high industrial sector competences that implement proprietary software systems on the basis of business process analysis.

Field 4
Description: Development of technical-organizational solutions by the usage of Open Source systems.
Providers: Consultancies that develop and implement solutions by using specific Open Source systems that may have been self-developed or extended.

If one reflects the described potentials of Open Source systems and then assesses them superior to proprietary solutions, the conclusion is that positions within field 1 and 3 will lead to competitive disadvantages in the medium-term. The usage of closed software technologies complicates the customer individual configuration and development of solutions.

Consultancies that are positioned in field 2 will hardly participate on profitable projects while it's still possible for IT oriented service providers to exploit this field's potential. Focusing on field 4 can be considered as the most promising strategy because Open Source based solutions of technical-organizational problems offer the greatest benefit for consulted customers. Consequently following the Open Source spirit, consulting services might be openly documented and thereby be made available in a standardized format (as reference models for example). Beyond that, strategic customer-provider alliances can be established on the basis of this open knowledge management. These are advantageous for both parties involved as the cooperative development of detailed problem solutions can be realized. Taken these assessments, transformation strategies can be formulated according to the arrows in figure 2.

Strategy A: Gain technical Open Source competence.

If the core competence is concentrated on the technical level, the expansion into the Open Source sector is a feasible strategy to become a specialized IT service provider.

Strategy B: Gain industry sector competence.

Based upon a distinct technical competence in the Open Source area, the build-up of industry sector competence makes it possible to offer more sophisticated consulting services to the customer – especially with respect to the development of industry specific solutions. In addition, the mobilization of a broad industry sector competence is a good occasion to increase the consulting royalties. If the market share of a consultancy is too small, the development of industry sector competence offers the opportunity of creating unique selling propositions. Building up strategic alliances with other consultancies that are focused on industry sectors might be an alternative.

Strategy C: Gain technical Open Source and industry sector competence.

This strategy is mostly congruent with strategy B. The development of Open Source competence has the higher priority because of the former technology focus.

Strategy D: Gain overall Open Source competence.

The goal of this strategy is to gain a substantial overview of available Open Source solutions to use them target-oriented (and maybe even linked). A lot of technical competence must be developed as well to be able to enhance existing systems. In this strategy, the build-up of strategic alliances with other, technical-focused, consultancies might be an alternative.

In the short- and mid-term, pure Open Source providers also have the possibility to move from field 2 to field 1 to link successful proprietary systems with Open Source systems or to transform proprietary software into Open Source projects. This reveals that the adaptation strategies of the present software vendors must be considered as well.

4 Outlook

The concept of Open Source offers an alternative way for developing and distributing software. It combines already existing knowledge about software development and software distribution and effects of the internet economy. Due to the increasing standardization level of royalty-free software interfaces and formats, the importance of proprietary software solutions is assumed to decrease accordingly. If the service levels are comparable, customers feel more attracted to Open Source distributors to avoid strategic dependencies on vendors of proprietary software solutions. Present software costs will remain existent to some extent, but in the context of Open Source, these costs are no longer software license fees but service royalties. Opened and linked collaborations appear to produce more sophisticated solutions for company problems. The potentials arising offer consultancies the possibility to realign their strategic focuses. Today companies no longer ask the question "Why should we use products whose quality and enhancement is not guaranteed by any company?" but rather "Why should we buy software whose quality is no subject to public discussions?" and "Why should the control of central systems be given into the hands of another company?"

In the field of operating systems, the Open Source concept is already highly established – mainly because of the engagement of distributors like Redhat and SuSE. Time will show if a similar development will take part in the field of application software. The consulting business can play a key role therein.

References

Apache Software Foundation (ed) (2006) The Apache HTTP Server Project. URL: http://httpd.apache.org

DiBona C, Ockman S, Stone M (eds) (1996) Open Sources: Voices from the Open Source Revolution. URL: http://www.oreilly.com/catalog/opensources/book/toc.html

Feller J, Fitzgerald B, Hissam SA, Lakhani KR (eds) (2005) Perspectives on Free and Open Source Software. Cambridge et al., The MIT Press

Free Software Foundation (FSE) (ed) (2006) The Free Software Foundation. URL: http://www.fsf.org

Müller M (1999) Die Philosophie des GNU und die Pragmatik des Open Source. In: Open Source – kurz & gut, Cologne et al., O'Reilly & Associates, Inc., 17–19

Netcraft Ltd. (ed) (2006) Web Server Survey 2006. URL: http://news.netcraft.com/archives/web_server_survey.html

Nüttgens M, Tesei E (2000) Open Source. In: Scheer A-W (ed) Veröffentlichungen des Instituts für Wirtschaftsinformatik Saarbrücken, 2000, No. 156, 157, 158

Open Source Initiative (ed) (2006a) The Open Source Definition. URL: http://www.opensource.org/docs/definition.php

Open Source Initiative (ed) (2006b) Certification Mark and Process. URL: http://www.opensource.org/docs/certification_mark.php

Open Source Initiative (ed) (2006c) Licensing. URL: http://www.opensource.org/licenses/index.html

PGP Corporation (ed) (2006) PGP Source Code Downloads. URL: http://www.pgp.com/downloads/sourcecode/index.html

Raymond ES (ed) (2000) The Cathedral and the Bazaar. URL: http://www.catb.org/~esr/writings/cathedral-bazaar

Stallman R (2006) About the GNU project. URL: http://www.gnu.org/gnu/thegnuproject.html

St. Laurent AM (2004) Understanding Open Source and Free Software Licensing. Beijing et al., O´Reilly Media Inc.