"Free and Open Source Software and its Implications on the Local Software Industry (Vienna and Surrounding Areas)"

Bachelorthesis of Mag. Jakob Perz

Seminar aus Wirtschaftsinformatik (0602), WS2009/10
Wirtschaftsuniversität Wien
Thesis Advisor Prof. Dr. Rony G. Flatscher

Table of Contents

1. Introduction

Open source software (OSS) – or, more generally put, „openness“ of organisations – is one of the thriving ideas in today’s IT markets. From a bunch of nerdy developers it slowly made its way into the heads of both global leaders and corporate executives alike. Large, well-known international corporations make headlines with open source business models: for example, think of Google with its Android platform for mobile phones, Apple, which based the successful OSX line of operating systems on BSD, Sun, which runs the MySQL project and several Java-based open source initiatives.

But once the marketing buzz wears off, what will remain? Is OSS a fully functional, lasting alternative to traditional commercial business models?

And even more importantly: what does this all mean for local companies in Vienna? Can they benefit from OSS, one way or another, as well?

In this paper, the author tries to find answers to these questions by examining and comparing both global OSS developments and the local situation in the Vienna area.

OSS is a broad topic; it influences more markets than just the IT sector. Therefore, some clarification, or rather restriction, is needed:

This paper explores Free *and* Open Source Software (FOSS). With all the marketing buzz about „openness“, it’s sometimes hard to distinguish between various degrees of freedom. A definition and explanation of FOSS has been attempted in chapter 2.

Furthermore, this paper targets the *local software industry*, meaning software developing companies, companies who create code in order to make money.

For this reason, not any company which technically belongs to the IT sector is included; a mere supporting or consulting business, for example, doesn't qualify as a producer of software.
What is more, mere usage of open source products (produced by others) doesn't count either. Today, almost everyone uses OSS. Be it knowingly, for example the ever popular Apache webserver or Linux as a server operating system, or unknowingly, for example Linux as a firmware within a TV set.

Another differentiation is to be made concerning universities: these organisations usually contribute significant amounts of code to various OSS projects. Since they do not qualify as commercial entities, however, they are not included in this paper.

Naturally the resources available for a bachelor thesis are not sufficient to scientifically explore this topic to the full extent. Also the statistics available on Vienna's IT sector are not as finely grained as one would wish for.
Nevertheless, the auther is confident that, under the given circumstances, a valid approach to produce useful information on the topic has been made.

2. Free and Open Source Software

This chapter gives a compact overview of open source, its main ideas and its history. It also defines the term FOSS.

2.1. Free as in „Freedom“

In English, the term „free“ can have two different meanings, according to the Merriam-Webster dictionary: either „you don’t have to pay for it“ or „no strings attached, liberal“.

It's important to understand that, in the context of the open source movement, "free" refers to the latter. Therefore, free OSS can be copied, altered, re-used. As OSI (the open source initiative) puts it: "The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources." [OSI10]

This means, basically, that it's perfectly legit to charge a fee for "free software" - despite the fact that most of the well-known FOSS projects, like the linux kernel or the apache webserver, come absolutely free of charge. If and how this affects business models which are built on providing FOSS will be evaluated in chapter 4.

2.2. Open Source

Programmers write code (or source code) in a programming language, which is then transformed into machine-executable code by a suitable compiler. In this traditional model, the source code is proprietary, a well-kept company secret, for obvious reasons: access to the source code reveals "how things work". It allows the savvy reader to understand and if necessary, modify the program. Machine-executable code, on the other hand, contains the most basic pieces of information only.

Using several techniques commonly referred to as "reverse engineering", it is possible to guess how a piece of machine-executable code works on the inside and, to a small degree, it's even possible to modify the code. Though this is legal in most countries, it is a complicated and rather limited approach.

In the light of these facts, the advantages of obtaining a program's source code become apparent.

For once, it allows to study the program's inner operations in order to gain knowledge, which can be later used on self-developed code.

Another advantage is flexibility: whoever has access to the source code has the possibility to modify it, which means a program can be fitted exactly to one's respective needs.

Lastly, having the source code also adds to security considerations. If the manufacturing company decides, for instance, to end-of-sale or end-of-life the product, it can be further supported and extended by the customer himself. The same applies to unforseeable events like the manufacturer's bankrupcy or changes in licensing and pricing schemes.

One example is the City of Munich, which operated most of its workstations on the operating system Microsoft Windows NT. As this operating system reached its end of life, Microsoft announced it would stop support and supply for security patches. The city was forced to upgrade to a newer version or change the platform ([CASS08]).

2.3. FOSS vs. Commercial Open source Software (COSS)

The acronym FOSS - Free and Open source Software - describes the combination of "free software" and "open source". Simply put: you get the source code and you may use it to your liking.

In contrast to FOSS, the term COSS - Commercial Open source Software - emerged as the open source movement gained in popularity to differentiate between wholly free and open products and others, which use open source rather as a marketing idea.

Quite often, COSS comes with parts which are open and freely distributable while other parts of the package, mostly the more interesting ones, remain closed.

Dirk Riehle, however, defines COSS as "Commercial open source software projects are open source software projects that are owned by a single firm that derives a direct and significant revenue stream from the software." ([RIEH09]). Riehle's definition doesn't seem entirely logical, since companies who make money off pure FOSS projects do exist, for example Vienna's Linbit (see appendix).

To be precise, the author therefore suggests to define COSS as any software package which includes free and open source parts, but closed, proprietary or otherwise restricted parts as well and/or restricts the rights to distribution and alteration of the source code using a tight license.

This doesn't necessarily mean that those proprietary parts are not open source, but in machine-executable code only (although that is probably one of the most common approaches). It can also mean that parts of the software are open source, but licensed in a restrictive way, for example if one needs to sign a non-disclosure agreement before gaining access to the source code, or if modification and redistribution of the source code is severly limited. This restriction is valid as such a proceeding would violate OSI's definition [OSI10] of free software.

In real life, it's hard to draw a clear line between FOSS and COSS. Many companies develop a licensing schema of their own or add exceptions to the GPL. For example, MySQL releases its free product under GPL "with linking exception", which basically means that programs may link to the MySQL library, yet are not required to be open source ([STAL70]).

Another example of a COSS business model is the frequently found "free for private use" license: you get an OSS and you may use it for private or academical purposes, but as soon as you use it commercially, you are required to buy a license. This restricts the right to freely distribute and use the software, therefore it cannot count as FOSS. Nosko et al. give BitKeeper as a perfect example: this OSS is free only to developers of open source code. Commercial (closed source) developers need to pay for a license. This led to a fierce debate in the open source community, especially after Linus Torvalds has decided to use BitKeeper for the development of the Linux kernel itself ([NGLF04]).

2.4. It’s All About the License

Chosing the (right) license is crucial to every open source project. The license governs if and how the resulting software may be used, distributed and altered.

It can be differentiated between the GNU General Public License (GPL), which is the most common and influential of all OSS licenses, and any other license. Krishnamurthy even goes as far as to differ between GPL-based and non-GPL-based open source business models ([KRIS03]).

This is because the GPL forces companies, which use GPL-licensed code, to give away the source code of the resulting software to any third party interested in it. Therefore, this behaviour clearly has effects on the resulting business models, on the way companies behave ([KRIS03]): a GPL company needs to worry about competitors ripping know-how from the freely available source code. Lee calls the GPL "copyleft", or a "viral" license: "The terms are viral because they preclude proprietary software companies from building software based on code falling under the GPL because any enhancement will be 'infected' by the GPL requirements." ([LEEJ06]).

FOSS basically demands for a GPL or at least a GPL-like license, as the freedom to alter and distribute the source code must not be redeemed ([OSI10]).
Yet licenses which are non-viral, but still FOSS-compliant, do exist. Lee names the Apache and the BSD license as examples ([LEEJ06]). The BSD license even allows companies to include its source into their products and to distribute those products in binary form, keeping the source code secret, as long as they publicly acknowledge the fact that they used BSD licensed source code. Thus being a rather market-friendly licensing model, BSD code has been incorporated for example into Microsoft Windows XP or, in large parts, in Apple's OSX operating system ([KRIS03] and [LEEJ06]).

Richard Stallman, free software evangelist, provides an overview of current, most important licenses and comments about them ([STAL07]).

2.5. The Evolvement of Open Source: From Hacker Culture to Business Model

Traditionally, open source was "typically viewed as a cooperative approach to product development and hence, more of a technology model. It is typically not viewed as a business approach. However, increasingly we find that entire companies are being formed around the open source concept." [KRIS03]

Companies need to earn money, either by increasing revenues or by cutting costs. If there are more and more companies engaging in OSS, there must be a business model to support it! So, open source clearly has outgrown the hobbyist's niche it was usually put in.

For once, some companies worked as early adopters of the new idea and managed to grow significantly in revenue streams using open source business models. Good examples are the publishing house O'Reilly or the Linux distributor RedHat ([KRIS03]). Another important factor was the evolution of the internet, which functions as a crossing point for almost all open source communities. It allowed open source developers, shattered all over the planet, to join their forces in a collaborative manner ([HEME08]).

The led, over time, larger corporations to use open source: Microsoft used parts of the BSD code for its Windows XP operating system ([KRIS03]), Novell aquired the major open source distributor SuSE ([HEME08]).

Today, open source is almost everywhere. This development was mainly carried on by the success of Linux. The broad adoption of Linux created demand for Linux-based applications and services. Due to its generous open-source license, Linux firmwares can even be found in mobile phones, TV sets and many other consumer products.

The movement towards business models can be seen as a two-sided evolution ([NGLF04]). On the one hand, modern moderate FOSS practitioners strive to make money off their efforts by building up business models around what started as a hobbyist's experiment. Examples for this are successful open source enterprises like MySQL, SuSE, Bacula Systems or Vienna's Linbit.

On the other hand, traditional commercial corporations embrace the open source idea and try to set foot on this new territory - either by buying out open source enterprises or by releasing (some of) their products as open source themselves. Good examples are Sun, which released the StarOffice suite of programs and the Solaris operating system as open source. Another perfect example is SAP, which released its database to the open source community (which was later taken on by MySQL) or Netscape, which open sourced its browser (now known as Mozilla Firefox).

This development, however, isn't without controversy within the OSS community. Hardliners continue to warn of a mixup in capitalistic models and demand for OSS to remain pure. More moderate OSS proponents, on the other hand, welcome an economic point of view and embrace companies which invest in open source.

FOSS entirely drawn up in commercial companies, without resembling today's Microsoft? Nosko et al. see a middle ground: "Open source and proprietary software models remain distinct, but [...] less software will be licensed at either extreme in the future" ([NGLF04]). Yet the outcome of this struggle remains to be seen ([HEME08).

3. Software Industry in Vienna, an Overview

This paper deals with the implications of FOSS specifically on Vienna (though the findings apply to any other similarly sized and shaped location as well). It is therefore necessary to lay out a few facts of Vienna's IT industry.

The facts and figures presented in this paper are taken from official sources (Statistik Austria), which can be considered as reliable and accurate, however lack a level of detail. Additionally, data from studies and polls, done by market research institutes and sponsored by local communities and bodies are taken in concern. This data already comes at a rather high level of detail, but is probably less accurate. Regarding statistics and studies from UBIT, the local chamber of commerce for IT companies, it is safe to assume a reasonable amount of accuracy, as Austrian companies are required by law to join their respective body of the chamber of commerce.

Generally, Vienna lacks large multinational software corporations like Microsoft, SAP, Nokia or Apple. Instead, small-to-medium enterprises (SME) and single-person enterprises (SPE) are predominant: 62% of UBIT member companies are SPE, while only 4% have 20 or more employees ([RAML07]).

Nevertheless, Vienna's IT sector plays an important role in the local economy. 8% (5200) of Vienna's companies belong to the IT sector, yet this sector generates 15% (or seven billion Euros) of Vienna's gross added value ([RADA08]).

According to Radauer's study, main problems of Vienna's IT sector are a lack of cooperation between local IT companies, a too product-oriented development of products and a low rate of companies which engange in research and development (R&D) activies (only 36% of companies). Moreover, many start-up companies don't survive the critical phase between their third and fifth year.

Regarding open source specifically, for some 50% of IT companies OSS makes up more than 80% of their revenues. Another 20 to 25% make up to 40% of their revenues with OSS. Two thirds of these OSS companies are either SPE or have less than five employees. Main fields of business are servers, operating systems and databases ([RADA08]).

Radauer further defined barriers of OSS revenue growth: most importantly, a lack of customer acceptance (more than 25%), a lack of customer-side know-how (more than 30%), but also a feeling of insecurity regarding liability and support (more than 20%). He goes on to conclude that in order to further OSS in Vienna's IT sector, awareness training should be considered.

Consequently, both the City of Vienna and the chamber of commerce already pursue a strategy of promoting open source to companies within the IT sector as well as to end users (potential customers). For example, a platform (www.opensource.co.at) has been launched to list open source companies and encourage cooperation between them. Moreover, there are a number of events focused on open source and its economical aspects.

At the end of this chapter, a comparison between the IT sector of the City of Vienna and the City of Munich is to be made, because the City of Munich is in the process of adopting open source, which has lead to several studies. The data found in these studies can help to deduct conclusions which apply to Vienna equally well - provided that the cities are comparable.

Munich has 1.4 million residents, Vienna has 1.7 million. In Munich, 64% of all information and communication techniques (ICT) companies were SME. Average annual revenue per company in Vienna is 2.58 million Euros, which compares to 2.71 million in Munich ([MUNI07]). We can conclude that, on the whole, Vienna and Munich are comparable as far as the structure of the IT sector is concerned.

It must be noted, however, that statistics aren't exact, as there are some differences. For example, in Munich media, advertising and journalism companies are counted as ICT as well. In Vienna, this isn't the case. Nevertheless, on the whole, the cities are comparable enough.

Furthermore, there are no statistics available specifically for software producing companies.

4. Business Models for Open Source Software

As already laid out in the previous chapters, OSS is an expanding market and companies continue to venture into open source and generate lasting revenues from it.

Common to all existing open source business models is the goal to deliver a superior product, at a fast time-to-market and at lower costs by harvesting the power of the supporting community beneath the open source model ([RIEH09]).

But how does this actually work? How can these new open source business models be categorized? Several authors have already tried to accomplish this task and a few field studies have been undertaken.

Many authors name "distribution of open source packages" as one of several possible business models ([KRIS03], [WATS08]). Here, a company takes existing FOSS - often developed entirely by third parties - and creates and distributes this as a new package. The added value mainly stems from the coherent packaging and single source of distribution. An often given example is a Linux distributor like RedHat, any company which uses the Linux kernel and several GNU programs to package an entire operating system including a desktop and applications.

Watson continues to name two further open source business models. "Sponsored open source" are projects, where financial donors don't receive revenues from their contribution, for example the well-known Apache project. "Second generation open source" encompasses all activities where a company makes money off supporting services. This model includes the selling of licenses to corporate users, like MySQL does ([WATS08]).

Fitzgerald develops a more modern approach. He defines "value-added service-enabling" as a first category. Similar to Watson's "second generation open source", in this model revenue is generated from services built around successful, popular open source projects. Furthermore, "loss-leader market-creating": a product is released as open source to "create a market" (loss-leader); when this product gained in popularity, it becomes possible to sell licenses of a slightly more capable enterprise version to corporate users. Fitzgerald continues to introduce the new category "leveraging community software development", which basically means to cut costs by having the community contribute code. Lastly, "leveraging the open source brand", which means to benefit from existing, well-respected open source brands like Linux or Apache ([FITZ06]).

Finally, the ongoing FLOSSmetrics, a project sponsored by the European Union, deserves to be mentioned. FLOSSmetrics analyzed several thousand FOSS projects and more than one hundred open source companies to provide statistical information and derive a list of business models found. This list currently comprises ten categories. The by far most often found category is "product specialist", which means that a company created or maintains a FOSS projects and now lives on being the best source of knowlege regarding this very product ([FLOS10]).

Of course, one should be aware of the fact that however you may categorize open source business models, in real life there will be a mixture of all sorts of ways of generating revenue.

Because of this fact, and because of constantly, rapidly changing markets, the author of this paper attempts a categorization of open source business models at a relatively high level of abstraction. It is therefore suggested to divide all business models into "service and support" on the one hand and "software sales" on the other. These rough categories may further embrace more finely-grained subcategories, like "distributor", "loss-leader models", and others.

4.1. Service and Support

The first open source business model is "service and support". It encompasses all ways of generating revenue by providing either special services and/or additional support for a FOSS package.

Services include trainings, certifications for technical and sales staff, consulting services, software development services (developing new features on request), integration services (making FOSS and legacy software work together seemlessly), upgrade services (which means providing automatic, centrally organized software upgrades) and others.

Support includes installation and maintenance support, troubleshooting and "bug fixing", often combined with a service level agreement (SLA). This provides a level of guarantee and liability which is otherwise not available for OSS, but certainly crucial to any IT manager.

4.2. Software Sale

This second category consists of strategies to make money from selling software or licenses.

As said before, it is perfectly acceptable to charge a fee for a FOSS package. Many companies make money from selling FOSS packages physically, on a media, like a CD or DVD. Although FOSS usually can be downloaded easily and free of charge from the internet, some decision makers still prefer or require software packages from a reliable source, on a physical media they can lay hands on, which can be stored away in the company's archives ([KRIS03]). Canonical's Ubuntu Linux may serve as an example: while the distribution can be downloaded from Canonical's website free of charge, the company also sells CDs for 5 British Pounds each.

One of the most common approaches, however, is a loss-leader type system. Here, the FOSS is given away freely, while a slightly altered version of the product is being sold under a commercial license. The differentiation between the FOSS and the commercial product is often some added features which are required by enterprise customers. In other cases, the commercial product is stable and well-tested, supported and guaranteed by the company standing behind it, while the FOSS version is community-supported only and contains bleeding-edge code.

A very interesting variant of the loss-leader model is the so-called "dual licensing" scheme. The point here is that the same software product is sold under a commercial license as well as given away as FOSS.
The best known example of dual licensing is the maker of the most popular free database system, MySQL (now owned by Sun).

To conclude this chapter, a final business model which evolved around selling software is to be introduced: the "distributor". Typically, distributors of Linux are being mentioned, for example Canonical (Ubuntu), etc. However, any other type of FOSS could be "distributed" as well. In the case of a Linux distribution, the distributors job is to bundle a Linux kernel together with GNU software and other FOSS (desktop) applications into a single-package, complete operating system.
Many authors mention this type of open source entrepreneur as "Providing support services to enterprise customers" ([KRIS03]) - but that hardly qualifies as a "software sale". Yet RedHat, arguably the best known commercial Linux distributor, charges money per copy of its distribution. So what is it that sets RedHat apart from all the other Linux distribution which are available free of charge?
In RedHat's case, it's a tested and certified piece of software with a guaranteed life cycle. In other words, RedHat adds value which FOSS otherwise doesn't have: guarantees ([YOUN99]).

5. Factors of Success

Is open source per se the better business model? Is it superior to traditional commercial models? Of course not. Scholars have even shown that proprietary platforms mostly are more profitable than their open source competition ([ECKA05]). So, the answer is: it depends. So far, it was proved that small companies can gain relative advantages over larger companies when using open source.

But open source models are influenced by a number of factors which are critical to success. Failure to observe these common pitfalls may easily be fatal for a company. These factors can also serve as a baseline to determine wether or not an open source business model makes sense in a given economical situation.

5.1. FOSS and Monopoly

One of the foremost business principles of traditional commercial business models was, and is, to build up and protect company knowledge. Source code, as the container of a considerable amount of a software developing company's know-how, is a closely kept secret.

Unsurprisingly, this is also one of the harshest arguments open source critics bring forth against open source business models: the knowledge you build up, the time and money you put in research and development, all that should be made available to the public free of charge, enabling competition to rip all that without paying a fee? "[...] it is clear that FOSS firms face significant challenges, as they are not only working in a clearly distributed system, but in one where the traditional means of control are of limited use." ([DAMA08])

Matthias Bärwolff shows that this scenario doesn't have to be true. On the contrary, even with FOSS it is possible to exert a certain amount of monopoly by controlling the FOSS project. He further states that this is even necessary in order to sustain profitability, that competition demands for a certain degree of monopoly.

Bärwolff names two important points about a FOSS project where monopoly and market power can be exerted: the rights to the software (brand) name, which can be a trademark and is often hold by the head developer or the developing company, and the control over the project's organisation itself ([BAER08]).

As an example serves the web browser Mozilla Firefox. Although this is FOSS, it exerts market power and discriminates competition: first, Mozilla partnered with Google to make Google the default search engine and received financial consideration in return. Second, Mozilla executes tight control over its rights to the "Firefox" brand name. While the browser's source code is GPL licensed, it is strictly forbidden to modify the software *and* distribute it under the name "Firefox". This means that even for the slightest change in the source code by a third party, be it a bug fix, a fork of the project and a release under a different name would be required.

Usually, however, monopoly power based on FOSS is weaker than that of proprietary software, as the latter keeps the source code a secret and, in addition, can hold patent rights on some parts of the software ([BAER08]).

5.2. Managing the Community

FOSS is tightly bound to communities. Open source communities develop, support and promote FOSS projects. Also companies which rely on open source business models need to interact with open source communities, at least to some degree.

Communities can have various positive effects on an open source company. "Fruitful relationships with communities can therefore be an important mechanism for creating revenues and gaining protection from competitors" ([DAMA08]). Communities are mainly used to import additional knowledge into a company, which can be later transformed into innovative products: "Using communities is a way for firms to increase the total amount of resources they can draw upon in the innovation processes" ([DAMA08]).
Companies leverage a community's knowledge for development, for example to use community-developed code in a commercial product, for support by having the community support and troubleshoot customers or supply patches for software bugs, and for creative input, ideas or simply the propagation of the brand name ([KRIS03]).

This effects, however, don't come automatically: "firms can benefit from the creative ideas of individuals outside the company. However, the inflow of ideas and innovations from external parties does not happen spontaneously; new strategies and ways of working are required to create a good fit between what the firm does and the resources and capabilities available in its external environment." ([DAMA08]).

"The main problem with seeding and growing a user community is the support cost. [...] open source firms address this problem by leading the community to become self-supporting." ([RIEH09]). The management's goal should therefore be to enable the community to work as much as possible without requiring efforts from the company's part. Riehle suggests that the best way to achieve that is to create a "win-win situation". A community member's contribution to the product should be rewarded by increasing his reputation within the community.

5.3. When to Use Open Source, When Not to

Open source isn't the perfect strategy for every economical situation an IT company might find itself in. While it certainly is a good business model for some, others failed at trying to set foot on open source grounds. So, when is it favourable to use open source, and when is it better to abstain?

According to Krishnamurthy, the potential of an open source business model can be analyzed along two dimensions: "customer applicability" and "relative product importance" ([KRIS03]).
"Customer applicability" defines the possible market share, or in other words, the number of potential users who are interested in the product.
"Relative product importance" means how important the product would be to the user. For example, an operating system certainly is important to everyone, while a funny screensaver is only considered "nice-to-have".
He concludes that the highest potential lies with products having both a high relative importance and customer applicability ("stars" quadrant), although open source makes sense in the remaining three quadrants as well.
This theory, however, is a bit vague. It may serve as an indicator wether a certain software has market potential, but not as an indicator wether an open source business model should be chosen.

Regarding the aforementioned business models, it makes sense to introduce three criterias where open source is most advisable. For once, small companies can benefit the most from open source. Second, a company having tight ties to and a good standing with communities should pursue this strategy, as they have the best chances for a successful integration. And last, a company lacking market power (be it because of its size, competition, etc.) could go for open source as an alternative way of creating market power.

In the end, the decision pro or contra open source will probably be handled on a case by case basis. Further research into this issue would be desirable.

6. Structured Interviews

For this paper, several structured interviews with local OSS companies were conducted, posing questions concerning the open source business model and the relation to the local economic environment (see Appendix A for a list of questions and detailed explanations).

The companies interviewed were:

- Linbit, the creator of high-reliability software
- Proxmox, a distributor of virtualization packages
- Herbert Pötzl, creator of a virtualization software for linux
- Mutti Medien OG, a web development company
- Toscom, a SPE focusing on OSS consulting, integration and development

In short, the outcome is that different economic situations exist for each company. Linbit is operating globally, in a highly technological market. Toscom, on the other hand, is focused almost entirely on Vienna.

Another interesting fact is that most interviewed companies not aware of or not interested in OSS promotions undertaken by UBIT and the local government.

Astonishingly, 3 out of 5 persons interviewed couldn't even name 3 good examples of leading Viennese open source companies! Yet 5 out of 5 persons interviewed named Linbit. This proves two things. Firstly, Linbit is a perfect example for a leading local OSS company, as they receive broad peer recognition. Sadly, it also proves either that there are extremely few local companies with a global outreach or that networking between local OSS companies is inefficient to a degree where one company doesn't even have knowledge of its local competition.

In the following two chapters, real-life examples, small case studies, of well-known Vienna-based software development companies are presented. These companies are amongst Vienna’s open source elite and influence others to follow their lead.

The companies listed here are selected based on international reputation (news items, mailing list discussions, etc.), achievements (code produced, awards) and a peer vote (see Appendix A, question 11).

Given Vienna’s small structured IT sector (see chapter 3), it comes as no surprise that these companies are very small, with employee numbers between 5 and 15. Yet they do have a global influence and recognition in their respective fields.

6.1. Case-Study: Linbit

The small company from Vienna’s 12th district developed a high-availability solution for storing enterprise data. DRBD (the Distributed Replicated Block Device) works on top of Linux and makes sure that data is always available to read from and write to, even if all but one cluster nodes fail.

With a total of just 10-15 employees, Linbit achieved world-wide success: DRBD has been widely accepted all over the world with key customers like Siemens, Ericsson or Deutsche Telekom.

Since 2009, DRBD is also part of the Linux kernel, which means that it’s available out of the box on every linux system. Linbit has recently opened a subsidiary in the USA and a widely spread partner program.

Right from the start, Linbit focused on an open source business model. DRBD was released under the GPL license as FOSS, while commercial support was sold through consulting contracts. Today, a side product (DRBD Proxy) is available under a commercial license.

Linbit is therefore deploying a mixed business model, selling service and support as well as commercial licenses.
Yet the main product, DRBD, can't be seen as a loss-leader, because most of the company's revenue is generated from service and support contracts for DRBD.

Apart from personal sympathisation for the open source movement, one of the main reasons was that Linbit, being a small, underfunded start-up, needed a freely available, GPL-licensed product to build a userbase and thus attract customers.

Phil Reisner, CEO, further states that the open source business model isn't any better or worse than traditional commercial models. It just fits his company, and that's the main reason it's in use. The only major drawback seems to be competition ripping of know-how, source code, and so on; according to Reisner, it happens relatively often that Linbit's products are being simply rebranded and commercially sold under a new name, violating the GPL.

Although Linbit undoubtedly is the prime example for Vienna's open source sector, it isn't embedded into local networks and open source initiatives. According to Phil Reisner, Linbit's CEO, the Austrian market is way too small to deserve special attention. Linbit's products are highly specialized solutions, which can only be marketed on a global scale.

Interestingly, Linbit doesn't complain about a lack of support from the public sector. Reisner claims that management bases its procurement related decisions usually on simple facts, like TCO. Either you're in, or you're out. License or business model used doesn't affect customers such a great deal.

6.2. Case-Study: Proxmox

The 2-5 employees company, situated in Vienna’s 5th district, develops open source solutions mainly for virtualization platforms. Its goal is to enable a simple, useable and stable environment for virtual platforms, integrating technology like OpenVZ. Virtualization, in this context, means running various instances of (different) operating systems on top of one hardware platform.

Licensed under GPL (FOSS), Proxmox competes against commercial solutions from Microsoft, Citrix, VMware and others. In this competitive market, Proxmox has already managed to gain a significant userbase. Active management of partners (partner program) is currently under contruction. The Company is earning money by selling (paid) support contracts for their products.

The company's main business model is being a distributor: existing FOSS projects are thoroughly tested and bundled into a new package. Most of the employees' working time is spent on making the various FOSS work together smoothly in a single package.

According to Martin Maurer, director of Proxmox, the company chose an open source model for personal beliefs, but also because it allows a start-up to survive a highly competitive market. The free base product attracts many smaller companies. On the other hand, it turns out to be rather hard to convince large organisations with correspondingly larger budgets to buy Proxmox.

Martin Maurer is aware of open source initiatives and networks in Vienna, but isn’t very well integrated. He believes the public sector should spend more money on successful open source projects, if not switching to using open source products at all.

Such a shift in technology usage, like the City of Munich did, would have positive effects on small and medium sized companies in the whole IT sector, he claims.

7. Conclusion: Implications on Software Industry

So far, we have seen that open source is still on the rise. Small open source start-ups prove themselves with lasting success, large corporations embrace the open source idea publicly. Both practitioners in the economy and scholars develop, describe, categorize open source business models.

But wether a company actually engages in open source or not, are there any implications of FOSS on the software developing industry (as a whole)? What are the implications specifically for the Vienna region?
National and local governments alike openly think about switching to using FOSS internally - or are already in the process of doing so. What are the implications of such activities? What would that mean for Vienna? Would it be a good idea for (local) government to actively promote FOSS?

The most obvious aspect about FOSS, which certainly comes to mind, is zero costs for licenses. That should affect any organisation adopting open source. But there is more.

Recent studies and scientific papers have clearly shown that there are considerable implications of FOSS on the software industry. In this chapter, the most important ones are analysed.

7.1. Easier Market Access for SME

FOSS comes without any costs for licenses, upgrades or mandatory support contracts. This means that SME and even SPE can get an enterprise infrastructure up and running yet avoid high initial costs. Accessing a market becomes easier for such companies.

Basically, this can be seen as a problem of marginal costs. For example, almost every IT company nowadays needs a relational database management system, like the commercially available (and extremely costly) Oracle. A company can calculate the costs for Oracle licenses by dividing by the number of employees using the system, the number of products sold, or any other meaningful factor. This means that the larger a company gets, the cheaper the calculatory price per factor unit becomes. The reason is that marginal costs of software is often close to zero: Oracle costs the same for one employee as for a hundred. A small company would incur a larger burden.

But it's not only SME which benefit from zero license costs. Larger corporations may leverage freely available code to cut on their software production costs. Today, developing an entirely new operating system would be economically unfeasible for any organisation. That is why even market leaders in the field use open source code in their operating systems. Think about Microsoft using BSD code in Windows XP or Apple using the whole BSD platform as the basis for its OSX line of operating systems ([NGLF04]).

7.2. Competitive Advantage for SME

Marginal costs also come into play when developing software. No matter how many copies of a software package are sold, the costs to develop the software are (practically) always the same. In other words: starting from the second copy sold, marginal costs are close to zero.

This is, of course, an advantage for larger companies: the more copies of a software package are sold, the lower calculatory development costs get!
But there is more to it: large companies benefit from a well-known brand name, a reputation built up from key accounts and readily available software solutions.

A small company, on the other hand, has to prove itself and often doesn't have a solution available, it has to develop one in the first place. This would lead to a considerable advantage of larger companies when it comes to pricing.

Here is where open source comes in: using FOSS as a basis is a way that allows small companies to quickly develop competitive solutions in a cost-effective way. These cost savings transform into lower price offerings to customers. Thus even very small companies can compete with stable, fairly priced products ([GUEN08]).

What is more, small companies can benefit from the established brand name of the FOSS package. For example, the Linux distribution "Debian" is known and established as a stable and secure server operating system. A small company which sells solutions based on Debian, may reference this brand name and leverage the positive effects ([FITZ06]).

7.3. Potential for Innovation

Another important implication of FOSS is the larger potential for innovation stemming from open source ([GUEN08] and [VASH03]).

According to Günther, 71% of IT companies claim that the usage of OSS directly leads to new product innovations ([GUEN08]). The reasons for this high potential for innovation, which is specific for OSS, are a "quick diffusion of knowledge", "no exclusivity of knowledge" and "complementary offers" ([GUEN08]).

"Quick diffusion" means that by the very nature of open source communities, that is the cooperation of developers over the internet, knowledge quickly spreads over aligning communities and companies.

"No exclusivity of knowledge" refers to open access to source code and related knowledge for everyone.
"Complementary offers", finally, means that by using open standards, it becomes easier for interested third parties to to develop additional programs.

Günther concludes that open source leads to an increase in added value.
But there is more to it. The potential for innovation, as laid out in the study, allows primarily small companies to benefit from open source. Traditionally, companies try to accumulate and protect knowledge internally. Therefore it is logical that a large company with a relatively longer history has accumulated a much larger database of knowledge in the past than a small (start-up) company can have. A potential for rapid innovation allows small companies to bypass this assembling of knowledge and enter the market with innovative products quicker and easier.

7.4. Larger Variety of Applications

Furthermore, open source leads to a larger variety of available applications, as Economides and Katsamakas proved ([ECKA05]), and a stimulation of the local software market ([VASH03]).
In their study, the authors compared proprietary versus open source platforms, where "platform" refers to, for example, an operating system, a central processing unit or a game console.

The authors found that, in the case of a proprietary platform, a two-sided pricing strategy takes place. For example, a gaming console would be a platform, while games for this specific console would be applications. The producer of a proprietary gaming console now makes pricing decisions towards two directions: first, he charges the end user who buys the console. Second, he charges third party companies, competitors, which develop applications (games) for this console - these development companies have to pay royalties in order to get access to a software development kit and documentation, for instance.

As a result, there are two kinds of applications (games) for this console on the market: those that were developed by the platform's producer himself and those that were developed by third parties.

The platform producer may now decide to make its applications loss-leaders. He could drop prices and make more money from selling platforms instead. The application doesn't need to be profitable on its own, it can be sold below marginal costs.

An open source platform, on the other hand, employs open standards and allows every interested company to develop and sell applications. In this scenario, each application has to be profitable, because it can't be subsidised by platform sales.

The authors however conclude that this fact alone doesn't constitute that proprietary software per se has negative influences on the market and that open source should be propagated: "It follows that it is important to analyze *the whole system of prices* to determine the appropriate public policy in technology industries". Yet the authors "... also found that the variety of applications is larger when the platform is open source." ([ECKA05]).

Another conclusion from the study is that "When a proprietary system competes with a system based on an open source platform*, then the proprietary system most likely dominates both in terms of market share and profitability*" ([ECKA05]). It can be concluded that all this is positive especially for small companies, as they can get "a foot in the door" and participate in a more variable market more easily.

7.5. Bringing It All Together: Implications on Vienna

In chapters 7.1. to 7.4. global, general implications of FOSS on software developing companies were examined. But what about implications specific to the Vienna area?
Unfortunately, research on this topic is scarce. Papers and reliable studies are missing, not only for Vienna, but also for comparable regions. A few papers outline the situation for Munich (and some other German cities). Although these are also pretty basic, from a scientific point of view, this thesis needs to rely on them.
Apart from that, the conclusions presented in this chapter are drawn directly from the outcome of the previous chapters.

As shown in the previous chapters, any company can benefit from FOSS. It is clear, however, that SME can benefit by far the most. So, Vienna with its small structured IT sector, where SME and SPE are predominant, should be able to gain from FOSS.

Another argument brought forth by FOSS proponents was that FOSS directly stimulates innovation. This should also be an advantage for Vienna, because the local IT sector is low on research and development (see chapter 3).

FOSS is unthinkable without a well-organized, motivated community. Statistics show that networking in Vienna is still lacking behind (chapter 3). FOSS could stimulate local networking and cooperations as companies come to realize the need for and the potential of communities.

8. Support for Open Source in the Public Sector

Open source is a big issue also for the administration of local governments. Being often the largest consumer of IT goods and services in their jurisdiction, government decisions to use or even actively enforce FOSS has huge effects on the local software developing industry as well ([GUEN08]).
Vienna already promotes open source. But what effects would a decision to switch the administration to open source bring? Is it advisable to support open source in other ways?

Many governments already support open source one way or another, and many studies have been recently made on this topic. According to scholars, the trend to promote FOSS is strongest in Europe ([RAJU07]). On municipal level, several German cities, including Munich, are early birds in this field ([CASS08]).

For a government, several ways to promote OSS exist: it can create policies or even laws to enforce open source in its administration or beyond. It can directly subsidize open source projects (note that open source is largely indirectly subsidized as a lot of FOSS is developed at public universities) or it can simply use open source in its administration internally ([LEEJ06]).

Wether a government actively enforcing or subsidizing open source has positive effects on competition and the market as a whole is fiercely debated ([LEEJ06], [RAJU07], [GUEN08]). The majority of scholars, however, see positive outcomes by and large.

8.1. Network Effect

An administration adopting open source leads to a strong network effect. This is not only because a government administration usually is the largest consumer of IT software, but also because government contractors tend to follow (indirect) standards set by the administration, so they tend to adopt open source applications as well ([LEEJ06]). More open source users also means a larger community, therefore better support and more active development. In the end, everyone benefits from this network effect.

8.2. Added Value

A government adopting open source also increases local added value. The reason is that instead of paying considerable license fees to a foreign corporation, more small contracts are awarded to local contractors: open source applications need service and support, integration, adaption of legacy software and training of employees. All of these tasks can be given to local contractors, thereby increasing local revenues and added value ([GUEN08]).

8.3. Underproduction of Public Goods

Lee states that an underproduction of public goods may occur. He argues that few private organizations engage in developing open source code, as they can't directly and exclusively capture the benefits. Open source, in this perspective, is basically a public good. It is a government's duty to ensure that enough is produced. Without a government subsidizing and supporting open source, an underproduction would take place, which would have negative effects on the economy ([LEEJ06]).

9. Summary

Open source is still one of the hot issues in the IT sector worldwide and in Vienna. Open source companies continue to develop successful business models built upon the idea of FOSS. At the same time, traditional commercial companies try to embrace the idea of open source and venture into this new field as well. To some degree, a convergence is taking place.

Successfully established open source business models are numerous. They can be roughly categorized firstly into “service and support”, which means giving away FOSS for free while charging for anything that is related to consulting, integration, guaranteed support, and the like. Second, “software sale”, which means charging a license fee on a per-copy basis. Among the many subcategories here, the most important one are various “loss-leader” models, where the base software package is given away free of charge as FOSS, while a slightly altered version is commercially sold.

Generally, the recently developed open source business models – and open source on the whole – bring several advantages, most of all for small companies. For them, it becomes easier to access the market, as entry hurdles like huge license fees become obsolete. They also gain a competitive advantage over traditional commercial enterprises, as they leverage freely available open source code as the basis for self-developed products. Furthermore, it has been shown that open source is a driver for innovation and also leads to a greater variety of applications on the market.

Open source, however, doesn't work "out of the box". A few stepping stones deserve special attention. Open source companies should integrate with communities and carefully define what they expect from the community and how they can get it. Furthermore, there are strategies to maintain a monopoly on the FOSS project, which is crucial to prevent competition from ripping a company's work.

All of this links to Vienna’s IT sector, which is dominated by SME and SPE and already has a strong presence of open source oriented companies. Statistics also show that only a very small percentage of companies actively engage in research and development. As small companies benefit the most from OSS, institutions like the local government and the local chamber of commerce are well-advised to promote the idea and usage of open source to companies within the IT sector and end users.

Studies also show that a government using OSS has considerable influences on the local IT sector and the added value. Instead of paying costly software licenses to large corporations, a higher number of small contracts are given to local specialist companies. It also has positive effects on knowledge dissemination and local research and development.

Scholars however argue over wether it is advisable for governments to actively enforce or subsidize open source or not.

10. Appendix A: Questions for Structured Interviews

For the structured interviews which have been conducted fort his thesis, the following questions have been asked. For coherency's sake, the questions listed below were translated from German to English.

Comments and explanations are printed in *italic letters*.

1. Please explain your open source business model?
Various open source business models exist (see thesis chapter 4). This question wanted the interviewee to explain, in his own words, "how he actually makes money".

2. Do you run open source business models only or rather a mixture between open source and commercial models? Which licenses are in use?
Some companies define themselves as being "open source". They believe in what they do and produce open source only. For others, open source can be useful in a number of situations, but is still regarded as an addition to, not a replacement of traditional commercial models.
In FOSS, the license plays an important role. It restricts the kind of business models which can be used.

3. Why OSS?
Why does the interviewee use OSS? Why not traditional commercial models, which maybe are easier to setup? Is it a matter of taste, of personal abilities or simply enforced by market conditions?

4. Did you start out producing OSS in the first place or did you switch to OSS business models later?

5. What are, in your opinion, advantages and disadvantages of OSS models?

In academic literature, many (dis-)advantages of OSS are layed out; but how do practitioners in the field see their own situation?

5.1. Do you have problems with competitors ripping of your products, know-how, either legally or illegally (e.g. “GPL violations”)?
In open source, everyone can read, modify, and therefore make use of “your” product. Commercial competitors, who rip off open source products by just rebranding them, are an often mentioned threat to open source companies.

6. Well-known, large enterprises successfully use OSS business models. In your opinion, is it possible to employ their strategies for SMEs as well?
Vienna’s IT sector is a comparatively small-structured one, mainly missing out on large software producers.

7. Do you know about the efforts of the Austrian Economic Chamber (WKO) and the City of Vienna (Magistrat der Stadt Wien) to advocate and promote OSS?
The two organizations mentioned in this question try to actively promote OSS models, for example by launching the website opensource.co.at or by hosting a number of open source focused events. In this question, the interviewer tries to find out wether these activities actually reach the intended recipients or not.

8. Are you part of these newly established networks?
A major focus of the two organization from question 7 is to form (informal) networks of SMEs to cooperate on projects and exchange know-how. The interviewer tries to find out if target companies actually use these.

9. Do you believe that the usage of OSS in the public sector has positive effects, either directly or indirectly, on the local IT economy, compared to traditional commercial software?
The public sector is one of the most important players on the IT market, not only because of its size, but also because it influences many companies depending on public contracts. It has been debated in Vienna to switch to

OSS (almost entirely), like Munich did. Given the structure of Vienna's IT sector, what would the outcome of this policy look like?

10. Does OSS bring any (direct or indirect) advantage specifically for SMEs and SPEs (single-person enterprises)?
Vienna has a very small-structured IT sector, lacking large key corporations while being dominated by SMEs (5-10 employees) and SPEs. The interviewer tries to find out if OSS has any specific advantages for small companies, which large corporations maybe don't have?

11. Please name 3 good examples of Vienna-based OSS developing companies, of course not including your own.
The interviewer is trying to find out if there are a number of companies, which are well-known in the IT sector and stick out of the crowd.

10. Appendix B: List of Literature

[BAER08] Bärwolff, Matthias: Monopolelemente bei freier Software. In: Lutterbeck, Bernd, Bärwolff, Matthias and Gehring, Robert (Hrsg.): Open Source Jahrbuch 2008. Lehmanns, 2008.

[CASS08] Cassel, Mark: Umstieg auf Open source-Lösungen in der Stadtverwaltung: Ein Vergleich der Städte Treuchtlingen, München, Wien und Schwäbisch Hall. In: Lutterbeck, Bernd, Bärwolff, Matthias and Gehring, Robert (Hrsg.): Open Source Jahrbuch 2008. Lehmanns, 2008.

[DAMA08] Dahlander, Linus and Magnusson, Matts: How do firms make use of open source communities? Long Range Planning 41, 2008, pages 629-641.

[ECKA05] Economides, Nicholas and Katsamakas, Evangelos: Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. Law & Economics Research Paper Series Working Paper No. 05-20, New York, 2005.

[FITZ06] Fitzgerald, Brian: The Transformation of Open Source Software. MIS Quarterly, Vol. 30, No. 3, 2006.

[FLOS10] The FLOSSmetrics Project, http://www.flossmetrics.org, 2010, opened 10.1.2010.

[GUEN08] Günther, Jochen: Die Bedeutung von Open Source in der öffentlichen Verwaltung und der IT-Branche. In: Lutterbeck, Bernd, Bärwolff, Matthias and Gehring, Robert (Hrsg.): Open Source Jahrbuch 2008. Lehmanns, 2008.

[HEME08] Hemetsberger, Andrea: Vom Revolutionär zum Unternehmer - Die F/OSS-Bewegung im Wandel. In: Lutterbeck, Bernd, Bärwolff, Matthias and Gehring, Robert (Hrsg.): Open Source Jahrbuch 2008. Lehmanns, 2008.

[KRIS03] Krishnamurthy, Sandeep: An Analysis of Open Source Business Models. University of Washington, 2003. http://faculty.washington.edu/sandeep, opened 10.1.2010.

[LEEJ06] Lee, Jyh-An: New Perspectives on Public Goods Production: Policy Implications of Open Source Software. Vanderbilt Journal of Entertainment and Technology Law, Vol. 9, No. 1, 2006. Stanford, 2006.

[NGLF04] Nosko, Chris, Garcia-Swartz, Daniel D. and Layne-Farrar, Anne: Open Source and Proprietary Software: The Search for a Profitable Middle-Ground. Law and Economics Consulting Group LECG, Chicago, 2004.

[MUNI07] The Chamber of Industry and Commerce for Munich and Upper Bavaria: Location for Information, Communication and Media. Munich, 2007.

[OSI10] The Open Source Initiative (OSI), http://www.opensource.org/docs/oss, opened 2.1.2010.

[RADA08] Radauer, Alfred: IKT Standort Wien im Vergleich. KMU Forschung Austria, Wien 2008.

[RAJU07] Raju, K. D.: Is the Future of Software Development in Open Source? Proprietary vs Open Source Software: A Cross Country Analysis. *Journal of Intellectual Property Rights,* Vol. 12, No. 2, 2007.

[RAML07] Raml, Reinhard: Arbeiten am IT-Standort Wien. IFES, Wien 2007.

[RAYM00] Raymond, Eric S.: The Cathedral and the Bazaar. http://catb.org/~esr/writings/homesteading/cathedral-bazaar/, 2000, opened 10.1.2010.

[RIEH09] Riehle, Dirk: The Commercial Open Source Business Model. In: Nelson, M. L.: Value Creation in e-Business Management, LNBIP 36. Springer, 2009, p. 18-30.

[STAL07] Stallman, Richard: Various Licenses and Comments about Them. Free Software Foundation, 2007, http://www.gnu.org/philosophy/license-list.html, opened 10.1.2010.

[STEB05] Stebler, Beat: Open Source in der Wertschöpfungskette. High Value Consulting. Aesch, 2005.

[STMO07] Singh, Param Vir, Tan, Yong and Mookerjee, Vijay S.: Social Capital, Structural Holes and Team Composition: Collaborative Networks of the Open Source Community. Proceedings of the

International Conference of Information Systems, ICIS, Montreal, 2007.

[VASH03] Varian, Hal and Shapiro, Richard: Linux adoption in the public sector: An economic analysis. Technical report, UC Berkeley, 2003.

[WATS08] Watson, Richard T.: The Business of Open Source. In: Communications of the ACM Vol. 51, No. 4, 2008, page 41pp.

[WEST08] West, Joel: Unternehmen zwischen Offenheit und Profitstreben. In: Lutterbeck, Bernd, Bärwolff, Matthias and Gehring, Robert (Hrsg.): Open Source Jahrbuch 2008. Lehmanns, 2008.

[YOUN99] Young, Robert: Giving it Away: How Red Hat Software Stumbled Across a New Economic Model and Helped Improve an Industry. The Journal of Electronic Publishing. University of Michigan, 1999.