BERLECON RESEARCH
information
technology
economics

FLOSS Final Report – Part 3

Free/Libre Open Source Software: Survey and Study

Basics of Open Source Software Markets and Business Models

Berlin, July 2002

BERLECON RESEARCH GmbH
Oranienburger Str. 32
10117 Berlin
Tel.: +49 30 285296-0
Fax: +49 30 285296-29
Web: http://www.berlecon.de
Email: info@berlecon.de

Acknowledgements:

This work was prepared by Dorit Spiller and Thorsten Wichmann from Berlecon Research. It is part of the final report for the project „FLOSS – Free/Libre Open Source Software: Survey and Study“, which was financed under the European Commission‘s IST programme, key action 4 as accompanying measure (IST-2000-4.1.1).

Table of contents

Figures and tables

1 Introduction

Background contribution to FLOSS project

This third part of the FLOSS final report contributes to the theoretical background of the FLOSS project by analysing the Open Source phenomenon, the market for Open Source software as well as business models for companies based on Open Source software. It also analyses best practices for the latter. The work is based on comprehensive literature and online research as well as on several expert talks and presentations on various conferences and trade fairs (such as the Wizards of OS 2 in Berlin, October 11-13, 2001 or the LinuxWorld in Frankfurt, October 30-November 1, 2001).

Organisation of the report

The report is organised as follows: Section 2.1 gives an introduction into the Open Source (OS) software phenomenon. The term OS software (OSS) is defined and compared to several other software distribution concepts. Various OSS license policies are compared and the major OSS products and development projects are introduced. Section 2.2 provides the theoretical background for a strategic market analysis of the software market in general. The software value chain for Open Source products is derived on the basis of traditional software production. For each part of the value chain, proprietary or commercial software is compared to OS software and free software.

Chapter 3 provides an overview of the software market and its relation to the IT (information and technology) market. It also segments the software market. Business dynamics for the software products market, for the software services market, and for the embedded products market are discussed. A basic understanding of market structures of the software market is necessary to evaluate the business models in Chapter 4. It is also needed to analyse the impact of OSS on the traditional software market.

Chapter 4 outlines the different OSS business models referring to the market analysis in Chapter 3. For each business model identified the basic business principle, the market and the critical success factors will be evaluated.

Results went into other parts of FLOSS project

This report formed the basic foundation for other elements within the FLOSS project. For example, the analysis of different software characteristics went into the construction of the user survey and the business models for companies that wanted to establish a business based on Open Source went into the analysis of firms' Open Source activities and the resulting policy implications.

2 Software and the Open Source phenomenon

2.1 The Open Source phenomenon

2.1.1 Definition of Open Source software

Availability of source code and price main criteria

There are several forms of software licensing and software distribution. They can mainly be distinguished with respect to two criteria: Availability of source code and price. Source code is software code written in a higher level programming language. It is different from binary code (code of ones and zeros only, which is readable for computer machines) because it reveals the structure and logic of a program. A software, which is distributed in binary code only, is also called closed source.

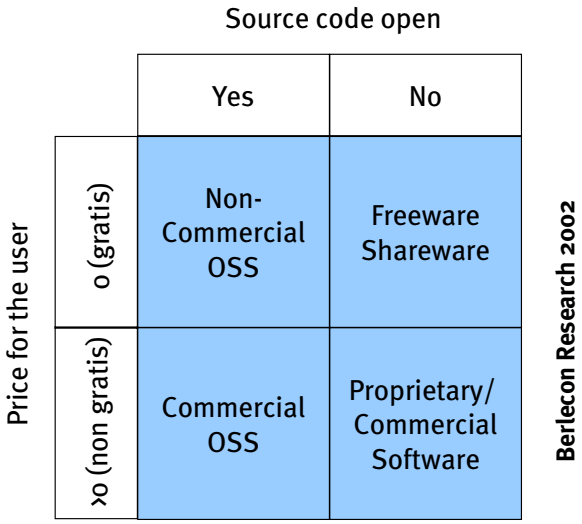


Figure 2–1 Classification of OSS versus other software

Most important software forms

The most important software forms are:

- The classical proprietary/commercial software: This software is typically distributed in binary form only. The source code is not available.
- Shareware: Software in this form is typically free for an initial period, but generally after a test period a license has to be bought. The source code is not available.
- Freeware: For this form, there is no license fee at all, at least not for the freeware product, but maybe for a complementary product. The source code is not available.
- Open Source Software (OSS): The source code of this software is available. There is commercial OSS (sometimes software that has been closed source and the source code was released) and non-commercial software, which can

normally be downloaded from a website. The main characteristic of OSS is that the users can freely use, modify and redistribute the software. However, there is a rich landscape of OSS licenses, which differ in terms of right for commercial use and other aspects. They will be examined in section 2.1.3.

Free Software and Shared Source

In addition also the terms "Free Software" and "Shared Source" can be found. „Free software" is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. It is thus much closer to „free speech" than to „free beer".[1] Shared Source is a relatively recent policy by Microsoft to give certain groups of companies access to source code of Microsoft product. This does not, however, include the right to modify the code.

2.1.2 Historic development[2]

Originally hardware as major revenue stream

From the early 1960s to the early 1980s, revenues in computer business were generated through selling and supporting hardware. For every hardware device, a special operating system was developed and deployed. The users of these systems were highly specialised IT experts. They were the ones primarily responsible for the development of additional software.

Unix a basis for the development of Internet technologies

Many efforts were dedicated to build an operating system that could be deployed on multiple hardware platforms. The most prominent example was Unix, which developed at the AT&T Laboratories and was published in 1969. Commercial users had to pay high license fees for using Unix, whereas academic institutions could use the software for a nominal charge. Consequently, Unix was the basis for the development of the Internet technologies. Many of these technologies were developed at universities and computer companies research laboratories, where Unix was deployed. Sharing the source code among software developers was commonplace. This tendency was reinforced by the emergence of computer networks like the Usenet that was started in 1979 to link the Unix community.

Turnaround in AT&T's licensing policy.

A critical event in the early 1980s for cooperative software development was the turnaround in AT&T's licensing policy. Unix became restricted to those who paid for the license to use is. Following this first step into the direction of closed source, the hardware companies IBM, HP and DEC started to develop proprietary Unix operating systems. They imposed "non-disclosure agreements" on the programmers dealing with the software and recruited many developers for commercial software development who had formerly contributed to cooperative and shared software development.

Richard Stallman and GNU

At that time, the programmer Richard Stallman worked in software development at the MIT. In 1984, he started a project to develop a free alternative of the Unix operating system. In addition, he established a special license, the GNU (named for Gnu's Not Unix) license, which was supposed to ensure that the software is indeed free and open for everyone. In order to support the GNU project, Stallman founded the Free Software Foundation (FSF) in 1985. Although linked often to the Open Source movement, Stallman is a proponent of Free Software, which goes much further in its demands.

Nevertheless, the GNU General Public License (GPL, see "Licenses") is central to the evolution of the Open Source phenomenon and has been used in many important

1. http://www.gnu.org/philosophy/free-sw.html.
2. This introduction can only be a very brief overview. To get more into the details see, e.g., Rosenberg (2000) and the literature overview.

projects. In the GPL, the principle of "Copyleft" is realised: It means that every copy of a program governed by the GPL, even if modified, must be subject to the GPL again. The licensing principles of the GPL, especially the "viral" effect,[3] are not suited for use in commercial software development as they make license fee-based revenue models impossible.

Antipathy from many businesses towards FSF

The FSF's philosophy behind software development provided great motivation for the Free Software community. But it also resulted in antipathy from many businesses which partly remains until today. The most prominent debate over the implications of Open Source Software, especially the GPL, and its effects on innovation takes place between Microsoft and Free/Open Source Software advocates, although such discussions are commonplace in more prosaic settings as well.

Linux

In the early 1990s, along with the increasing use of the Internet and the success of the World Wide Web, many new Open Source projects emerged. The most prominent example is Linux. Linux is a Unix-like operating system targeted to run on a personal computer. It was developed by the Finnish computer science student Linus Torvalds who used the GNU software tools. In 1991, he released the code of an experimental version under the GPL to a newsgroup and asked for comments and improvements. Within the last decade, Linux developed into a powerful operating system. The project shows characteristics that are typical for successful Open Source Software development over the Internet.

Cathedral vs. Bazaar

Eric Raymond, another central OSS developer and advocate, describes OSS development coordination as "Bazaar style," opposed to the "Cathedral" approach taken in classical software development, where development is organised in a more hierarchic, top-down and planned way. Linux has a modular structure, so individuals or groups of developers can focus on one part of the program. The principle of "Release often, release early" in combination with a constant peer-reviewing process ("Given a thousand eyes all bugs are shallow") is also opposed to commercial software development. (See also section 2.2.3 on software development).

OSI approach to software licensing

Linux was used increasingly in combination with the GNU tools. Because the operating system is central to IT infrastructure, it eventually became relevant for business use. In 1997, the Open Source Initiative (OSI) was founded in order to establish a more pragmatic approach to software licensing. The OSI was based on the "Debian Free Software Guidelines," which had been published in 1995. The central people for this development were Eric Raymond and Bruce Perens. Their aim was to promote OSS in commercial use because they believed that both the Free/Open Source community and the business world could benefit from wider OSS dissemination.

Open Source Definition by OSI

The OSI developed the Open Source Definition (OSD). The definition is not a license itself, but a guideline and trademark for OSS software licenses other than the GPL. Licenses according to the OSD guarantee several freedoms to software users, including commercial users. The "viral" effect of the GPL is not a requirement for OSD-approved licenses. In order to raise acceptance of OSS in the business world, the term Open Source Software instead of Free Software was established and widely accepted.

Rise in attention paid to OS projects in 1990s

The 1990s experienced a significant rise in attention paid to Open Source projects. Many companies from the IT industry began to support the projects. IBM, for example, supports a variety of Open Source projects. In 1998, Netscape was the first prominent company to release a proprietary software product as Open Source software.

3. "Viral" means that every derived software or software linked to a software published under the GPL has to be licensed under conditions that are compatible to the GPL.

Along with the Internet boom in the late 1990s, many investments were made in business models based on OSS. For example, Red Hat and VA Linux reached an incredibly high market capitalisation. Today, many of these companies are struggling or have changed their business models. But the development of OSS has not ceased. There are thousands of different ongoing projects. The most important projects will be described in section 2.1.4.

2.1.3 Open Source licensing

Most important part of the copyright is licensing

The terms "open source" or "free software" are not specific enough to describe a particular software license. In general, the most important part of the copyright in the software business regards the software's copying, distribution and preparation of derived works (Rosenberg, 2000, 89). The Open Source licenses have two things in common: The right to earn license fees is typically waived and the condition that the source code is made available is incorporated.

Copyleft

The major distinction of Open Source licenses is the principle of "Copyleft". "Copylefting" a software product means to impose the restriction on the user that derivative works have to be released under the same license (i.e., as Open Source) again. The most prominent and strongest case of "Copyleft" is the GPL. The "Copyleft" is not included in the opposing BSD-style licenses.

The restrictions and freedoms in an OSS license have to be balanced carefully when crafting a software license to both satisfy the customer's needs and the copyright owner's intentions (e.g. maximize the software's use for all users, including commercial organisations, or to ensure that the software remains "free" in the sense of the FSF's philosophy). There have been many attempts to define the ideal Open Source license, resulting in many different licenses in the open source area. In the following, we will shortly describe the most important licensing models in order to define Open Source more precisely.

Public Domain

Public domain abandons all copyrights

Releasing software to the "Public Domain" means to abandon all copyrights. The Public Domain principle is only applicable within some legal environments such as the U.S. In Germany, publishing software according to the "Public Domain" is not conform to German right (Urheberrecht). In the U.S., Public Domain software is largely developed with government support at universities or research institutes. It is available for every US citizen without any restriction. Taking the software into the commercial domain is allowed.

Shareware

Shareware make software available in binary form for small license fees

The intention behind shareware is to make a software programs available to as many people as possible. Shareware is distributed in binary form only. Most copyright owners impose a small fee for the license that normally has to be paid after a certain test phase.

Shareware evangelists argue that software producers want to be compensated for their work and a certain degree of "fair use" is needed. For freeware developers, it is supposedly even more difficult than for Microsoft to enforce their rights. Shareware products often have a built-in mechanism that dramatically reduces the comfort of usage after the trial period. This mechanism is intended to increase the willingness of the user to pay for the license.

Freeware

Freeware is freely available in binary form

Freeware is distributed in binary form without imposing license fees for usage. It is possible to grant the right to use the software exclusively to certain users, e.g. private or non-commercial users. Freeware often is part of a marketing strategy to promote complementary products. For example, Microsoft released the Internet Explorer as freeware to gain market share.

GNU Public License (GPL)

GPL most widely used Open Source Software license

The GPL is the most widely used Open Source Software license. The most prominent examples are the GNU project and Linux. The GPL was created by Richard Stallman and is representing the Free Software Foundation's philosophy.

There is no restriction on copying and distribution, but some conditions have to be met: The source code must be easily available to the user, the GNU GPL has to be enclosed with distributed software and interactive programs have to display the notices when started. Modifications are allowed as long as the author states what the change is, when the code was written and by whom. Derivative works are permitted but have again to be published under the GPL ("Copylefted"). The so-called "viral" effect arises because software incorporating GPL software must also be licensed under a GPL-compatible license. Software under another license can be "contaminated." The very strong "Copyleft" makes GPL not very business-friendly because any software company would have to reveal their software source code if they used (in the sense of included) parts of GPL software to develop it.

GNU Lesser General Public License (LGPL)

LGPL less restrictive

Because the GPL is so restrictive to commercial use, the FSF developed the LGPL. The LGPL allows commercial software to use libraries without being "contaminated" by the GPL. One main rationale was that a superior Free Software product could not be disseminated widely enough to become a standard and compete with commercial products. The first software issued under the LGPL were the GNU C libraries.

Mozilla Public License (MPL)

MPL license for Netscape browser source code

The MPL is the license under which Netscape released the source code of its Netscape browser client, Mozilla. The MPL imposes a kind of a "Copyleft" on the usage of MPL-software. The main difference to the GPL is that software under MPL can be incorporated into software products that can be licensed without "contaminating" the software. Therefore, the basics of the MPL are similar to the LGPL. Similar licenses are the IBM Public License or the Sun Public License. All of these licenses are OSI-approved.

MIT License and BSD License

MIT and BSD license

The BSD license grants the right to obtain a copy of the software including documentation materials available free of charge to everyone, as well as the right for trading the product commercially. Redistribution and use in source and binary forms is allowed. The licensing terms also hold for copies and ported versions of the software.

Part of the original BSD License was that the copyright owner's and the project's contributors names must not be used for promotion of a derived product without written agreement. But in the FreeBSD License, that condition is not mentioned anymore for practical reasons.

Table 2–1 Licenses overview

Software license	Available at no cost	Distribution allowed	No usage restrictions	Source code freely available	Source code modification allowed	Derived work must be free again	Linking with proprietary software allowed
Public Domain	X	X	X	X	X		X
Shareware	(X)1	X					
Freeware	X	X	X				
GPL	X	X	X	X	X	X	
LGPL	X	X	X	X	X	X	X
MPL	X	X	X	X	X	X	X
BSD-License	X	X	X	X	X		X

1) Shareware is gratis for a trial period only

2.1.4 Main Open Source products and development projects

Apache[4]

Development started at NCSA

Until 1995, the National Center for Supercomputer Applications (NCSA) provided the most widely used web server. The software could be copied and modified free of charge provided that NCSA would be notified about such derivative work. The users turned over software patches to the NSCA where they were incorporated into the NCSA software. In 1995, many NCSA developers changed to Netscape, so the mainly professional users were not satisfied with the software's quality anymore and put up a mailing list furthering development of the NCSA web server, then called Apache.

The Apache Software Foundation (ASF) was founded in 1999 and is now responsible for Apache development. Apache is known as a very good example for successful collegial management of the open source software development process. Brian Behlendorf as the community member responsible for public relations and involved in the project from the beginnings is strongly connected with Apache.

Apache dominates market

The Apache web server has dominated its market since 1996. Today, its market share is around 60 percent (Netcraft, 2001), leaving Microsoft IIS behind with around 30 percent. The Apache web server is included in many software solutions (e.g. IBM's WebSphere).

DNS and Bind[5]

BIND is foundation of domain name system

The Berkeley Internet Name Daemon (BIND), delivering the Domain Name System (DNS), is not very well known among general IT users. Nevertheless, as a program for turning host names into IP addresses it is a very important component of the Internet infrastructure. BIND is included in all UNIX systems and in many other systems, as well as being the de facto Internet standard for its functionality.

BIND was initially developed by Paul Mockapetris in 1984 and is now under the leadership of Paul Vixie from the Internet Software Consortium (ISC). The ISC was

4. Cf. www.apache.org; Lerner and Tirole, 2000; O'Reilly, 1999.
5. Cf. www.isc.org/products/BIND/; www.isc.org; O'Reilly, 1999.

founded in 1993 from Rick Adams with a donation from UUNET and is now supported by major players from the software industry (BIND Version 9 has been underwritten by Sun, HP, Compaq, IBM and SGI).

Free BSD[6]

FreeBSD: high-quality software without getting involved with money

In 1993, the first version of FreeBSD was released. It was initially built on the Berkeley Software Distribution (a free Unix system), and provides an Open Source Unix operating system. It is licensed to a large extent under a variant of the BSD license, imposing little restrictions on both commercial and non-profit users. NetBSD and OpenBSD are other Open Source projects built on the BSD, while FreeBSD is said to be the most popular variant. FreeBSD release engineer Jordan K. Hubbard stated that FreeBSD's goal was to produce "commercial" software in terms of functionality and quality without getting involved with money.

FreeBSD partly foundation of MacOS X

When FreeBSD was looking for a distribution channel, Walnut Creek became the major distributor and supported FreeBSD financially and through IT infrastructure equipment. One of the core developers, Jordan K. Hubbard, is now employed by Apple, which is strengthening the ties between Apple and the FreeBSD community. The Apple operating system Mac OS X is largely based on FreeBSD. WindRiver, owner of the FreeBSD trademark and distributor of the commercial BSD/OS stopped its financial engagement with FreeBSD, releasing the FreeBSD developers employed at Wind River and stopping technical support.

Gimp[7]

Gimp: OS graphics software

The Gimp (GNU Image Manipulation Program) is a graphics software often called the "free Photoshop" (a well known commercial graphics software from Adobe). It was developed by Peter Mattis and Spencer Kimball for Unix systems and is now part of most Linux distributions.

GNOME[8]

GNOME: Open Source desktop environment

GNOME (GNU's Network Object Model Environment) is competing with KDE in the marketplace of Open Source desktop environments for Linux and other Unix-like operating systems. Gnome is using the GTK (GNU GUI toolkit) and version 1.0 was released in March 1999 with Miguel de Icaza as the chief developer.

In May 2001, Eazel, a company developing a GNOME distribution that would bring more user-friendliness to GNOME, closed its doors. Nevertheless, Miguel de Icaza as the cofounder of Ximian Inc. is also pursuing the goal to bring Linux to the desktop. But differing from Eazel, Ximian is targeting business users, for example for use at point-of-sales terminals. In August 2001 Dell stopped delivery of its PCs with the Linux operating system because of its user-unfriendliness. Nevertheless, the GNOME project is supported by Sun, IBM or HP and has a strong community (as has KDE) making rapid evolution possible.

6. Cf. Yager, 2001; Michaelis, 2001; www.freebsd.org; O'Reilly 1999.
7. Cf. www.gimp.org; O'Reilly, 1999.
8. Cf. www.gnome.org; www.ximian.com; de Icaza, 2001; Hall, 2001; O'Reilly 1999.

GNU[9]

GNU project aimed at free Unix

GNU is an acronym for "Gnu is not Unix", expressing the starting point of this project very well. Richard Stallman from MIT started the GNU project in 1984 with the aim to build a freely available Unix-like operating system. When he quit MIT to focus on the GNU project, MIT continued supporting him with access to their information technology resources. The GNU project has not yet succeeded at building its own competitive OS kernel (Hurd), instead the combination of GNU and Linux is very successful. Very important determinants of the GNU project's evolution is the Free Software Foundation's philosophy and the GNU General Public License (GPL). The Free Software Foundation (FSF) was founded in 1985 and is financed by donations and fees for manuals and free software distributed on physical media.

The GNU project today is involved in many software projects related to their Unix operating system clone and the related tools that would offer a choice to commercially distributed software. Very important software products are the GNU C Compiler (GCC, the de facto standard compiler for Linux programs written in C and C++), the GNU Privacy Guard (GnuPG), GNU Emacs (a text editor) or GNOME.

KDE[10]

KDE: graphical user interface for Unix systems

KDE, the "K desktop environment", was founded in 1996. The aim was to develop a graphical user interface (GUI) for Unix systems designed to put an end to the heterogeneous and user-unfriendly Unix GUIs existing so far. It is now available in version 3.0.

KDE is GPL compatible and the KDE libraries are available under the LGPL, making commercial software development for the KDE desktop possible. All KDE applications and the important tools KDE is built on are licensed under the GPL.

A complementary product, Koffice, is developed as an open source office application suite for KDE, making KDE and Koffice one of the primary potential competitors for Windows in the desktop market.

Linux[11]

Linux today the best-know OSS project

The Linux project is tied to one person: Linus Torvalds. As a computer science student, he wanted to run a Unix-like operating system on his personal computer. As there was no satisfying software available, he started developing his own OS from "Minix", a Unix operating system for educational use created by Prof. Tanenbaum. When the software reached a certain degree of maturity, he posted the code on a discussion group asking for comments and possible improvements. Today, Linux is probably the best-known OSS project. As Linux is a Unix-like operating system kernel (providing the basic functionality of an operating system), it is deployed with the GNU tools.

The founder Linus Torvalds is still in the centre of Linux development. The Linux community consists of thousands of developers coordinated in the "bazaar style". But nevertheless, there are some hierarchic structures with Torvalds on top and so-called "Lieutenants" managing sub-projects.

---

9. Cf. www.gnu.org; O'Reilly, 1999; Freyermuth, 2001a, 2001b.
10. Cf. www.kde.org; Eduak, 2001; O'Reilly, 1999.
11. Cf. Computerwoche Spezial, 2000; O'Reilly, 1999; Rosenberg, 2000; Schmitz, 2001; Freyermuth, 2001a, 2001b.

Many organisations make up Linux ecosystem

There are many organisations involved: Mostly commercial players like the distributors (Red Hat, SuSE, Mandrake Soft) and some major hardware and software companies (IBM, HP) porting their software and hardware to Linux have a direct interest in Linux. Along with employing developers working on Linux they provide infrastructure resources (for example the Open Source Development Lab sponsored by IBM and HP among others). The Linux Standard Base (LSB) as a consortium of commercial and non-profit organisations is trying to establish common standards for Linux development (to prevent the emergence of diversity that happened to Unix in the 80/90s).

Linux released under GPL

Linux was released under the GPL. The GNU/Linux system grew in importance mainly in the server market, partly as a substitute of commercial low-end Unix systems and partly as a competitor of Windows NT. As a desktop operating system, the market share is low, but complementary developments focusing on user friendliness (e.g. GNOME, KDE, OpenOffice) may evolve into competitive Linux solutions for consumers. The main characteristic is that Linux is a very mature and in many ways technically superior operating system which has not forked as happened to Unix (with its existing commercial products like Sun's Solaris and the several free versions, namely FreeBSD, NetBSD, OpenBSD). At the moment, that is one major issue in further Linux development targeted by the LSB, as the different distributions add differing functionalities to the kernel, resulting in interoperability problems.

Mozilla[12]

Mozilla: Open Source browser project initiated by Netscape

Mozilla is the name of an Open Source browser project initiated by Netscape. When Microsoft began offering the Internet Explorer for free and improving its performance, Netscape released the code not protected by third party copyright law for its browser. The license created was the Mozilla Public License.

When releasing the code, the hope was to leverage the power of the open source development process. According to some outsiders' estimations, initially there was no large developer community and Netscape developers accounted for most of the contributions to the code. Apart from the reason that Netscape could have retained too much control over the Mozilla project, the code base is very large and the program very complex. That makes the program difficult to understand and adds to the problems of making Mozilla a successful OSS project. Ideally, there should be a modular software design (as is the case with Linux) for being able to share the work. Only in 2002 Mozilla released its 1.0 version.

MySQL[13]

Relational database MySQL

MySQL is a relational database server, initially developed in 1994. The Swedish company TcX Dataconsult AB, that later took on the name MySQL AB, published the software under the GPL (and parts of it under the LGPL) in 2000. It also offers versions under a license that enables the owner to use MySQL in commercial solutions. MySQL AB, a Swedish privately held company, which is financed with venture capital since July 2001, is the owner of the copyright. Major companies have deployed MySQL, e.g. Motorola or Yahoo!.

12. Cf. www.mozilla.org; Rosenberg, 2000; Miller, 2001.
13. Cf. www.mysql.com; www.nusphere.com; Jaeger, 2001a; Wayner, 2001; Dyck, 2001.

NuSphere is a company of Progress Software and offered a MySQL distribution along with solutions that integrate MySQL with Apache and PHP. In July 2001, MySQL AB accused NuSphere in Massachsetts/USA of violating the GPL, the first time a violation of the GPL was brought to court.

Perl[14]

Scripting language Perl

Perl was developed in 1987 by Larry Wall. First being a tool for searching, manipulating and printing texts, the scripting language Perl evolved into a network and systems administration tool. The CGI-programming functionality made it the "glue" of the Internet, especially suitable for dynamic web pages.

Wall is further developing Perl with around 100 other developers while employed at O'Reilly. ActiveState Tool, for example, is offering professional tools for Perl, and the Comprehensive Perl Archive Network (CPAN) gives access to hundreds of Perl modules.

PostgreSQL[15]

Relational database PostgreSQL

PostgreSQL is an object-relational database server. Its roots are at UC Berkeley; in 1996 a team took the existing code and developed it into an open source SQL-database. It is mainly deployed in private use and projects not involving mission-critical business operations. Other reasons keeping PostgreSQL from spreading may be its incomplete documentation and the lack of professional support, as have many OSS projects in the beginning.

In September 2001 Great Bridge, the company employing 3 of the 6 core PostgreSQL developers ceased operations. Their main goal was to offer professional services to businesses. Nevertheless, indicating PostgreSQL's quality, Red Hat offers its own database product built on PostgreSQL 7.1.

Python[16]

Scripting language Pyhthon

Python is a scripting language first released in 1991 by Guido van Rossum. It evolved quickly into a powerful object oriented and interpreted programming language, often compared to Perl or Tcl. The product is offered in a version completely integrated with Java (Jpython), enabling it to run on every computer with the Java Virtual Machine installed.

Van Rossum is working for Zope Corporation as the director of the Python Labs and is the leader of the project. The PSF (Python Software Foundation) was created in March 2001 "with the specific goal to own the intellectual property that comprises the Python programming language." Major PSF sponsors are Zope Corporation and Active State.

14. Cf. www.perl.org; O'Reilly, 1999; Lerner and Tirole, 2000.
15. Cf. www.postgresql.org; Wayner, 2001.
16. Cf. www.python.org; www.zope.org; O'Reilly, 1999.

Samba[17]

Samba for print and file server connection

In 1993, Samba was developed by Andrew Tridgell (who is still leading the project) from the Australian National University as a Windows file server and print server for Unix platforms. The current version, 2.2.x, implements SMB and CIFS (two network protocols) and adds to the comfort of replacing Windows servers with, e.g., Linux servers. Samba is included in most of the Linux distributions.

There have been some tests running Samba as a file or print server against Windows servers. SGI was said to offer the fastest Windows NT fileserver using Samba. And in November 2001, PC Magazine tested Windows 2000 against Samba/Linux as a print server, the latter outperformed Windows on both low and better quality systems.

Sendmail[18]

Sendmail handles majority of mail traffic on the Internet

Sendmail was developed in 1981 by Eric Allman as a Mail Transfer Agent (MTA). At that time he worked at UC Berkely and wrote sendmail to exchange mail between the university's network and the Arpanet. (Freyermuth, 2001a, 7). From the beginning, Sendmail was focused on openness regarding differing mail protocols, on the routing functionality (not as many other programs getting involved with the front end or mail delivery) and on a flexible configuration file.

Allman is still central to sendmail development. In 1993, Allman had to rewrite the program as there existed many forks, and succeeded in reuniting the communities. Complementing the free basic sendmail program, he and Greg Olson founded Sendmail, Inc. in 1997. Sendmail, Inc. offers commercial versions of sendmail, adding managing tools and security solutions. Sendmail dominates its market, reaching at times a market share of 75 to 80 percent of mail delivered via sendmail.

StarOffice/OpenOffice.org[19]

StarOffice suite as competitor to Microsoft Office

The StarOffice suite is a Sun Microsystems product since Sun took over Star Division two years ago. It is competing in the office suite market dominated by Microsoft Office. Corel and Lotus Smart Suite are other competitors. A year ago, Sun made the StarOffice code available as the "OpenOffice.org" project. The basic software can be used and commercially exploited by everyone, competitive advantage is gained through adding proprietary extensions, e.g. by Sun in its StarOffice product (which has to be licensed for a fee since May 2002) with a proprietary spell checker.

Tcl/Tk[20]

Tcl/Tk for developing graphical user interfaces

Tcl/Tk was developed by John Ousterhout while he was a Professor at UC Berkeley. Tcl is a scripting language comparable to Perl and Python, Tk is a toolkit for developing graphical user interfaces (GUI). Initially developed for Unix systems, Tcl/Tk now is also supporting Microsoft Windows and Apple Macintosh platforms. Ousterhout was working with Sun until he founded his own company "Scriptics" (with a similar business model as sendmail, Inc.) to further the development of Tcl/Tk.

17. Cf. www.samba.org; Kaven, 2001; O'Reilly, 1999.
18. Cf. www.sendmail.org; www.sendmail.com; O'Reilly, 1999; Lerner and Tirole, 2000.
19. Cf. www.openoffice.org; www.sun.com/staroffice/; Eduak, 2001.
20. Cf. www.scriptics.com; O'Reilly, 1999.

Zope[21]

Zope: Open Source application server

Zope is one of the projects based on initially proprietary developed software. In 1995, Digital Creations was founded by Rob Page and Paul Everitt. They were developing software for newspapers to manage advertising. Later on, they had two main software products: the free, open source toolkit Bobo and the commercial web application platform Principia. They did not have the financial means to enter the established web application server market (with competitors as IBM Websphere or BEA Weblogic), so in 1998, they adopted the typical Open Source Business model shifting their business from license revenues to a services company's business model.

By releasing Principia as OSS, they hoped to push marketing and build up a user community that would help to improve the product. Today, Zope is renowned for being a competitive alternative in the application server market, especially for content management and portals. In Germany, several companies offering Zope services founded the "Eurozope Association." And Digital Creations even changed their name to Zope Corporation to express the importance the project has for the company.

They are also engaged in the Python project as a sponsor to the Python Software Foundation and employer of Guido van Rossum, leader of the core development group working at Zope, the PythonLabs.

2.2 Software characteristics and the software value chain

2.2.1 Software characteristics

Software features in general

Following the introduction on Open Source software, we will now take a closer look at software and its features in general. Firstly, software is immaterial, which makes it different from physical goods. Secondly, users of software rarely buy only the product, instead they buy the service related to the software. Enterprise users normally buy solutions, a combination of software, hardware and services. For our analysis, we need the following basic differentiation:

Software defined to consist of product and service

For easier discussion we will define Software as consisting of two substantially different parts, which will be called product and service. The product is the license that is sold to use the software. This license can be unique (customised software product) or it can be duplicated as many times as possible (standardised software product). The services surrounding software products range from consulting, implementation, support, and training to application management. Therefore, we will be using the term software product (not software itself) and software services.

Additionally, one can differentiate between individual and standard products and services. This gives the following matrix:

21. Cf. www.zope.org; Osterberg, 2001; Rosenberg, 2000.

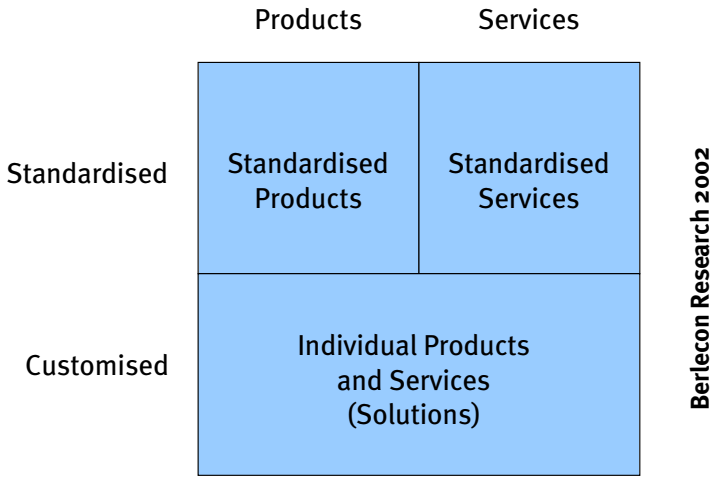


Figure 2–2 Matrix of products and services

The software product is determined by several characteristics that make it different from physical products and, therefore, influence the way the quality of software products can be determined.

Software product characteristics

The product characteristics can be summarised as follows (Balzert, 1996, 26). Software ...

- Is an non-material product with no physical limits.
- Cannot be worn out and does not physically deteriorate.
- Is a product of intellectual property. This results in high development costs and low per-item costs for standardised products. The incremental costs for additional sales are negligible. Therefore, volume is very important for sales.
- Is aging (better hardware allows better software). This aspect is becoming less important over time.
- Is difficult to measure in terms of physical product measures. Either technical or financial equivalents have to be found to measure the value of software.

2.2.2 The software value chain: overview

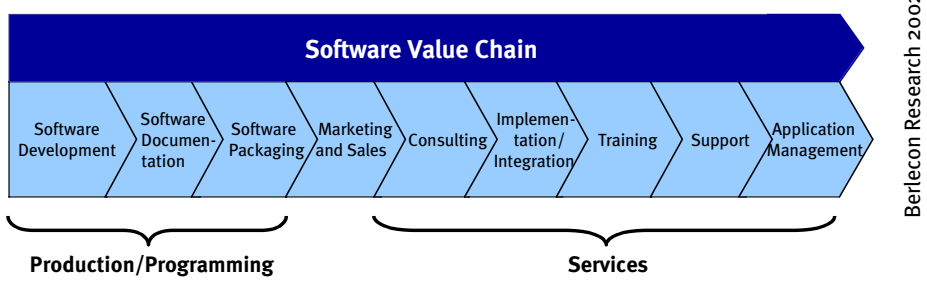


Figure 2–3 Software value chain

Value chain with product-related and service-related steps

In the software value chain, we see again product-related and service-related steps. Additionally, there are marketing and distribution steps. Here, value is created in form of information about the products.

The value chain consists of three major parts:

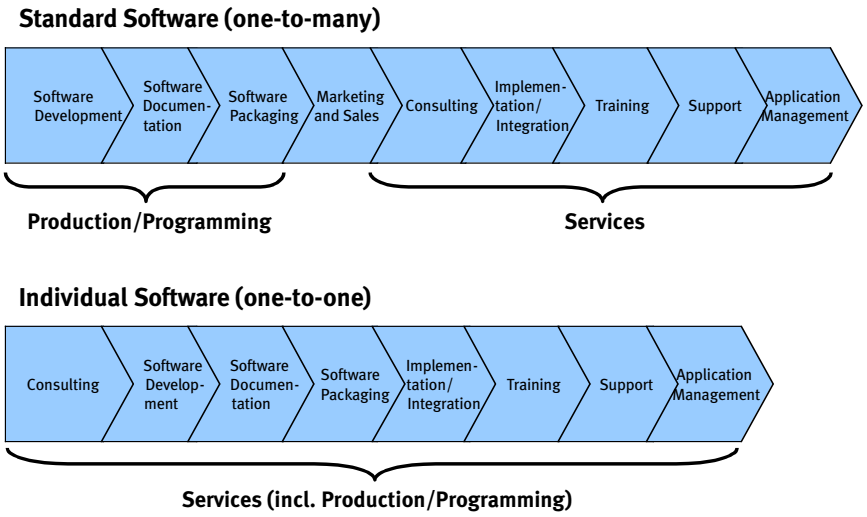
- Production/Programming,
- Marketing/Sales (Distribution), and
- Services.

Differences between standard and individual software

It is instructive to note, once again, the differences in the value chain for standard software and for individual software. Although the same steps need to be performed to create similar value in the form of a software product or solution, the order of the steps is different. It is not surprising that, for individual solutions, the first step is consulting and the definition of requirements for any individual customers. For standard solutions, the first step is the software development. (Assuming that market research has been conducted before.)

In the following section, the steps of the value chain will be analysed separately. We will describe first what the essence of each step is. Furthermore, we will point out major differences between proprietary software and OSS.

Figure 2–4 Software value chain for standard software versus individual software



2.2.3 Programming and production

Software development

Definition of software development

Software development is the writing of technical code in order to initiate and control the functionality of machines, especially computers, in a certain way and so that they meet certain requirements. Software development can result in either standard or individual software products. It includes the development of productivity tools. Productivity tools are technical codes (such as programming languages or compilers) which allow to change and control other technical codes more easily.

Trends of software development

Software development is characterised by the following trends (Balzert, 1996, 27). The indicators given in parentheses can be used to measure the relevance and importance of these trends for certain software.

- Increasing importance of software versus hardware products (indicator: relation hardware products sales vs. software products sales in the IT market).
- Increasing importance of software-related services versus software products (indicator: relation sales of software products vs. software-related services).
- Increasing complexity (indicator: number of person-years or number of source code lines, measured for consecutive releases of a software product).
- Increasing quality requirements (indicator: number of defects per 1,000 lines of source code).
- Existence of a trend from individual software products towards standard software products within a certain product line. This trend corresponds to the

classical cycle of development for technological products. At first, a special software product is developed to address an individual problem, resulting in individual software. Later, as the demand for similar problem solutions increases, the software becomes standardised. (indicator: relation of customised software sales to standard software sales).

- Increasing burden of already existing software (adaptation of old software versions to new components accounts for two thirds of software development costs, trend increasing).

Software development

There are basically two principles of developing software (Raymond, 1997):

Cathedral...

- Cathedral-principle: This principle is based on centralised planning and execution and describes the way of traditional proprietary software development. Software development is organised top-down within a strong organisational hierarchy. The name of the principle refers to the building of cathedrals in the medieval age.

...and bazaar

- Bazaar-principle: This principle is based on decentralised planning and execution and explains the way of much Open Source software development. The development is organised in a networked manner. A community of developers works on parts of the software and on improving it. The patches are brought together via the Internet on a central website. The name of the principle refers to an oriental bazaar where anyone can trade anything and anyone can participate.

Cooking-pot model

A principle which goes even further than the bazaar principle is the cooking pot model (Ghosh, 1998) with following argumentation: On a bazaar, traded goods eventually change their possessor but their value is not changing. In the cooking pot, instead, single ingredients become melted and result in something new, which is valued higher than the mere sum of the ingredients.

Development models often blended

In reality, these principles of development are sometimes blended. For example, a software product is developed in the cathedral-principle until its first release. The first release is further developed in the bazaar-principle. Users can add modules and functionalities.

Critical success factors

There are several critical success factors on this part of the value chain:

- Functionality requirements must be met, the software must perform in the desired way.
- High software quality must be achieved.
- Time-to-market must be short because of adaptability to new hardware environments. However, this factor is becoming less important because hardware development and substitution seems to slow down in some product lines.
- Development costs must be minimised.

Differences between proprietary software and OSS

There are different development models and a different developing process for proprietary software and OSS. Proprietary software allows for the centralised definition and direction of functional requirements. OSS develops in whatever direction individuals want it to. Often reality is less distinct, as also commercial entities can adopt a bazaar-like development model at least to some extent. Also Open Source development is often more hierarchical (e.g. with “Lieutenants” deciding about which elements to include) than in the pure bazaar model.

It is sometimes hypothesised that debugging seems to be faster and better with OSS because an entire community is addressing problems. In numbers, OSS should therefore be of higher quality as measured by number of defects per 1,000 lines of source code.

Another often stated difference is that both sorts of software seem to differ in the priorities of development. Proprietary software tends to emphasize usability. OSS tends to emphasize stability and functionality because it is aimed at a different audience.

Major difference between both is that the source code of proprietary software is not open for the user. Therefore, adaptations of software to special hardware conditions or functionality conditions can only be made in the form of customisation or cannot be made at all. For OSS, the source code is available and gives many more opportunities to further develop the software according to special requirements resulting in better performance.

Software documentation

Definition of software documentation

Software documentation is the aggregation and editing of information related to the application and use of software products. There are three kinds of documentation: Firstly, the software itself is a form of documentation. If the source code is available, software developers can use the code to get information about the product. Secondly, there is electronic documentation explaining the software code within the software. Thirdly, there is edited documentation, usually in paper form, explaining the functioning of software.

Differences between proprietary software and OSS

Proprietary software is usually sold with documentation. Documentation is part of the developing process at some late stage. In comparison, OSS is normally not a finished (box-)product – therefore, documentation is not always sufficient for the user. Documentation production and sales are part of the business model of some companies related to OSS, such as Red Hat (electronic documentation) and the O'Reilly publishing company (paper documentation).

Software packaging

Definition of software packaging

Software packaging is the aggregation, integration and optimisation of software products or components. (It can include the debugging of the new integrated software product.) Simple aggregation of different software components (e. g. on a CD-ROM or on a website) is not part of software packaging as it adds no value on the product level. It adds value on the distribution level and, therefore, part of the marketing/sales value chain element. Hence, software packaging assumes that there is value added to the package and the packaged software product is different from the single parts added.

Differences between proprietary software and OSS

Proprietary software usually is sold in a packaged way. Packaging is part of the developing process at some late stage. For OSS, packaging is an important part of many business models based on OSS. As the original OSS usually only exists in form of many different development projects on the web, the so-called distributors put the current versions of the software together, optimise them and then sell the package.

2.2.4 Marketing and distribution

Goal: optimise instruments of marketing mix

The general goal of marketing and sales activities is to optimise the combination of instruments within the marketing mix (product, price, distribution, advertising) in order to create an awareness within the target group (potential customers) for the unique selling proposition of the product. Eventually, this awareness is aimed to result in the purchase of the product.

Marketing and market structures for standardised products and services (see chapter 3) are very different from marketing and market structures for individual products and services (and solutions). As marketing and distribution is more important and more complex for standardised products and services, we will focus our analysis on these. There are several critical success factors on this part of the value chain:

Moore's law

An often cited success factor in the standardised products market is called Moore's Law: According to this "law", one has to reach 40 percent market share within a certain niche within 12 to 18 months. At this level of success – assuming the closest competitor is far behind – word of mouth starts spreading the message that one is the market leader. Once that happens, one's market share is expected to increase above 50 percent within the following 12 months. Moore's law is based on the assumption that successful products tend to become even more successful whereas less successful products tend to fall even further behind. Network effects can explain this.

Differences between standardised proprietary software and OSS

Difference between the two sorts of software in marketing and distribution issues are similar to the differences in documentation and packaging: Marketing of the software product itself is normally done by the company that develops the software. For OSS, however, this is different. The open source software product itself, i.e. the license to use the software, is usually not promoted separately (at least not with traditional marketing instruments like advertising). Only packaged distributions, complementary products and OSS services are promoted.

There are more differences to be found in the four instruments of the marketing mix.

Product

Product-related issues regard things like quality, time-to-market, and thus many characteristics directly resulting from the development process. The product dimension, however, also includes quality issues of the service dimension because, in the software value chain, not only the products but also the services are sold.

Price

There is a major difference between proprietary software and OSS. (See also Chapter 2.2.1 for the difference between Freeware, Shareware and OSS.) Proprietary software can be sold (or rather the right to use the software in form of a license). Therefore, different pricing strategies make sense. OSS can be sold as well, but in form of packaged OSS. The price that OSS distributors can get for their software packages is normally significantly lower than the price that competitors offering proprietary software demand. This is due to the fact that less value is being added by packaging than by software development from scratch. Pricing itself is a delicate issue for OSS because the model to sell OSS does not really comply with the philosophy of many OSS community members.

Distribution

Distribution or sales is part of the marketing mix. Its primary goal is to physically deliver the product to the right place, where it can be accessed, used, or bought by the customer. Distribution of proprietary software is normally done by the software company that develops the software product. Alternatively, distribution is done by separate distributors or VARs (value-added resellers). VARs operate either exclusively or independently, sell combinations of software and hardware and act as a sales channel for software product companies.

Moreover, aggregation is part of distribution. Aggregation is the joint distribution of software components without software product changes. No additional features are added. (Aggregation is not part of the software packaging value chain, but it is part of distribution.) Aggregation for proprietary software products is part of the development process. Aggregation for OSS, instead, is done by distributors and forms the base for a separate OSS business model.

Advertising and Branding

In the mass market, advertising has an important function. For OSS, and for operating systems (Linux) in particular, branding is even more important because many users (especially the ones without deeper software knowledge) choose their software on popularity. In addition, the OSS distributors and other OSS-related companies have a vital interest in promoting the use of OSS in general. This interest is supported by the various OSS organisations such as the Open Source Initiative or the Free Software Foundation.

2.2.5 Services

Consulting

Definition of consulting

Consulting, in the context of software business and IT-technology, is a software-related service. Consulting includes the following steps:

- Analysis of the current situation.
- Definition of requirements (Conception of functional and technical requirements).
- Selection of software solutions.

Additionally, a normal IT-consulting project would include the steps of implementation, integration, tests and training of the users. All these steps will be discussed separately in the sections below.

Differences between proprietary software and OSS

There is no difference in the consulting process itself and in the type of company that offers the consulting service. Although there are many companies that specialize in OSS products and offer special knowledge in this category, their consulting process does not substantially differ from companies which are focused on proprietary software.

Implementation and integration

Definition of implementation and integration

Implementation includes the following steps:

- Installation (to make a software run on an existing basis of software or hardware).
- Configuration (to adjust the software to customer needs within the given framework without changing the source code of the software).
- Customisation (to adjust the software to customer needs, changing of the source code is necessary, additional systems integration can be necessary).

Integration can be in form of systems integration, networking integration and application integration. Systems integration can be part of customisation if it is necessary for the functionality of the software to integrate it in existing systems. Furthermore, integration can be a separate part of the value chain if the primary goal is not to implement new software but to integrate existing systems, networks or applications.

Differences between proprietary software and OSS

There is basically no difference if the source code and the tools are available. A general argument in favour of OSS is that the open source code enables customisation, which is not possible with closed source code. Hence, OSS offers unlimited opportunities for customising a software product.

Training

Definition of training

Training on how to use software can be given either through traditional seminars or through using web-based solutions (e-learning). Training can be classified in "train the user" (normally only on applications) and "train the administrator". These are similar categories to those used in the support.

Differences between proprietary software and OSS

There is not inherent difference between both kinds of software. Training is done by two types of companies: Classical training companies offer training for all kinds of software including OSS products. For them, training is the core competence. Software distributors and service companies offer training for their software products and solutions in particular. For them, training is one service that they offer, but normally not their core competence. For OSS products, training is offered by distributors and integrators.

Support

Definition of support

Support is a form of enabling the users and administrators of software to get the software performing in the way it was supposed to perform (fixing problems). This can be done through traditional telephone hotlines, e-mail, voice over IP or through newsgroups and message boards. There is user support and administrator support.

Differences between proprietary software and OSS

Support for proprietary software is offered by special service companies or by the software producer. Typically there is a distinction between private and corporate users. While private users are being offered the basic help functions within a program and simple telephone support, corporate users can often obtain more extensive support targeted at administrators or even at helping with customisation.

Support for OSS products is first of all offered by the OSS community through various forums. However, for corporate users this kind of support is not sufficient as there is no guarantee for a problem being solved. Distributors and independent OSS service and integration companies offer support. Some of the distributors employ home-working OSS developers who are experts for certain problems or parts of software. By their contracts, these OSS developers are obliged to give expert support to the distributors' customers.

Application Management

Definition of application management?

Application management includes all activities to ensure that a software is performing in the way it was desired to perform. It includes permanent checking of the following questions:

- What are the functional and resulting technical requirements?
- Is an update of the software to a newer version available and needed?
- How is data transferred to the updated version?
- What sort of backup is needed?
- What sort of log file handling is needed?

Differences between proprietary software and OSS

There is basically no differenc between proprietary software and OSS. Application management can be done either within the company (IT department) or through an external service provider. A feature of application management can be remote administration, which is usually offered by distributors and independent OSS service and integration companies.

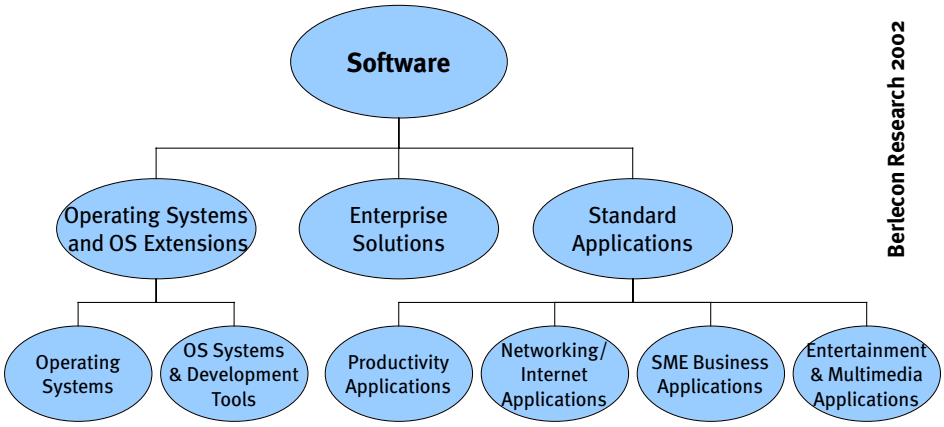
2.2.6 Software product categories

Different ways for software classification

Software can be classified in several ways. As chapter 2 provides an overview on software product related issues, we identified a technical or functional classification of software categories. The classification shown in figure 2-5 offers a better understanding of software product categories. The software products can either be standardised or individual.

This classification according to technical categories is only one possibility, although the most often used on Open Source software sites. However, the strategic market analysis in Chapter 3 will use a different segmentation according to user demands, as the users are the primary determinants of demand. The segments are slightly different from the technical classification.[22]

Figure 2–5 Technical and functional software product categories



22. This distinction illustrates a problem many of the Open Source based business models had. Coming from the technological side, the companies focused very much on product characteristics and less on the customers they were planning to address.

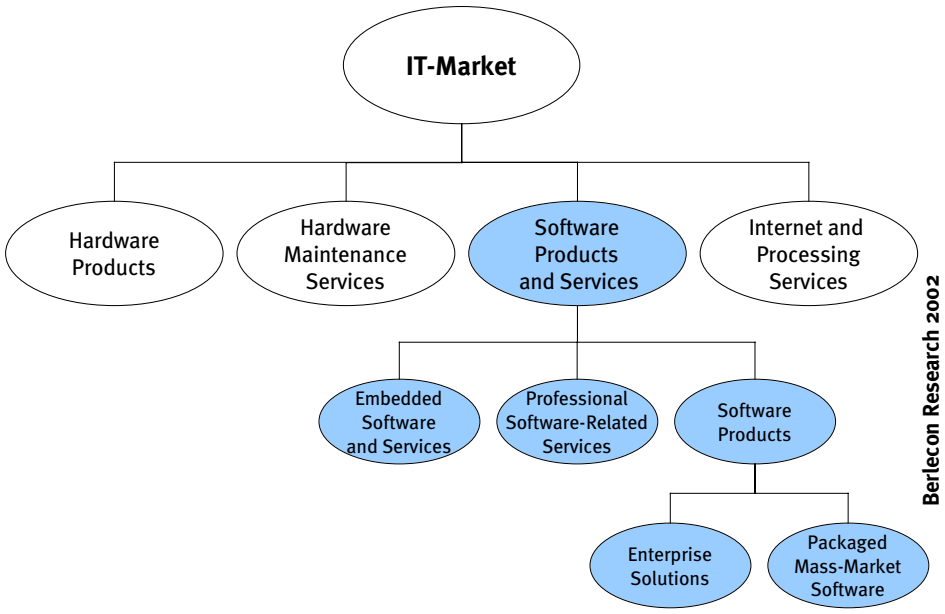
3 Open Source and the software market: market segmentation and hypotheses

3.1 The software market

The software market within the IT market

The IT market can be differentiated into four different market segments: hardware products, hardware maintenance services, software products and services, Internet and processing services.

Figure 3–1
Structure of the IT market



Source: Hoch et al., 2000.

Hardware market divided into four product segments

According to an definition from EITO (2001), the hardware market is divided into four product segments.

- Computer hardware (server systems, workstations, PCs, PC and workstation add-ons, e. g. PC printers and other PC add-ons),
- End-user communications equipment (telephone sets, mobile telephone sets, other terminal equipment),
- Office equipment (Copiers, other office equipment), and
- Datacom and network equipment.

Hardware maintenance

The hardware maintenance services market is basically supplied by hardware producers, by hardware vendors or by special service companies.

Internet and processing services

The Internet and processing services market is important for special service providers, xSPs (e.g., ASPs, payment service providers or B2B marketplaces) and similar businesses. Their main business focus is different from software development. The different business focus leads to different income models. Most income comes from service or usage fees.

These three segments do not need to be analysed in detail. We will limit the analysis on the segments and strategies, where OSS has a significant influence. It will become clear that between the four market segments several dependencies exist. The lines between the market segments blurr since, for example, some players from the hardware business shift to the services and to the software business. We will put the main focus on the analysis of the software market.

The market segments

Three software market segments

As shown in figure 3-1, the software market can be divided into three segments:

- Software products (further to be separated in enterprise solutions and packaged mass market software),
- Professional software-related services, and
- Embedded software and services.

The market segments will be analysed in detail in section 3.2. For each market segment, we will firstly explain the business dynamics in detail, and secondly apply them to the identified market segments by characterising segments, identifying the major players in the segments, and eventually generating hypotheses about the influence of OSS in the segments.

3.2 Business dynamics and influence of Open Source software

3.2.1 The software products market

Customer-driven market segmentation

Section 2.2.6 provided a technological and product-driven segmentation of software. For the market analysis, we will adopt a customer-driven segmentation. We will segment the market according to customer purchase decisions. For example, the market for operating systems on the desktop will not be regarded as a separate market because the purchase decision for an operating system for the desktop itself is usually not made separately but always related to either the hardware available or the applications available.

Distinction between enterprise solutions and packaged mass market software

As shown in figure 3-2, we will make two basic distinctions: The first one is between enterprise solutions and packaged mass market software. The market for enterprise solutions is substantially different from the market for packaged software because the revenue portion based on service fees is considerably higher. Therefore, the market for enterprise solutions can be mainly characterised as service market. But the service sales are primarily based on a special software developed.

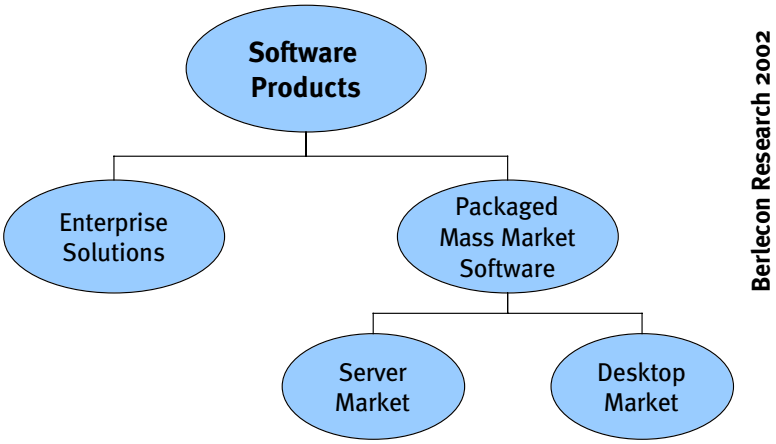


Figure 3–2
Structure of the software products market

The second distinction is within the market for packaged software. We distinguish the server and the desktop market. Within these segments, one could additionally segment between operating systems and applications. But due to the fact that the purchase decision for the operating system is dependent on the decision for applications and vice versa, we will look at the server market in general and only when necessary separate between operating systems, applications or even appliances. Similarly, we will look at the desktop market in general.

Distinction between server and desktop market for packaged software

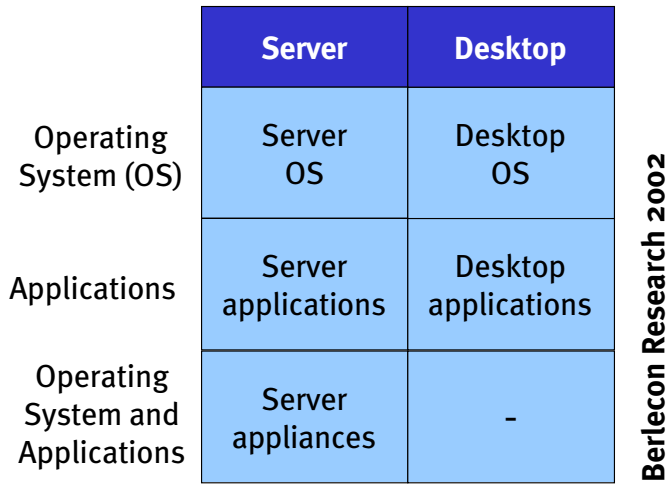


Figure 3–3
Software for the server and for the desktop/client market

Together with the segment enterprise solutions, we have three market segments:

Overall three market segments

- Server operating systems and applications,
- Desktop/client operating systems and applications, and
- Enterprise solutions.

Business dynamics in the software products market

In 1999, McKinsey analysed the software market and identified general business dynamics in the product business (Hoch et al., 1999). The analysis was focused on the traditional software business (development of proprietary software). The following conclusions were derived:

Analysis of the software business by McKinsey

In the software product business, entry barriers are low. The market is knowledge-driven, only low capital investment is needed. The low financial entry barriers boost high innovation rates (short time between releases), which, in turn, lower technical entry barriers. Therefore, firms in this market face a constant threat of new entrants.

Low entry barriers

Low marginal costs but high fixed costs

Low marginal costs result in worldwide markets, high fixed costs for research and development result in the condition that many copies must sell. The latter statement becomes blurred when considering OSS development. While the fixed costs are high in terms of hours devoted to a specific project, there are not necessarily monetary costs involved. Thus, an Open Source software could – at least in principle – survive despite a combination of large efforts and insufficient usage. Whether the programmers find this appealing in the long-run and will not switch to other, more successful projects, is another question.

Tendency towards concentration

There is a race for leadership and a tendency towards concentration because of the following aspects: Firstly, there exists a network effect because of the interoperability of programs. The ability of programs to operate and communicate with each other is crucial. Hence, the more users a program has, the more the users can benefit from it. Secondly, there is a barrier for people to switch once they are trained to work with a program. Thirdly, popularity of a program is a major factor for the purchase decision (again a network effect).

The tendency for concentration seems to be stronger in consumer markets, where people only execute programs. In markets, where the purchase decisions are made by people with special IT knowledge, the training barrier and the popularity barrier are usually not as high. People with IT knowledge base their purchase decisions to a lesser extent on popularity. They also get used to a new program more easily.

Unstable market position

But any market position of a player in the software products market is never stable because of technology switches.

Server operating systems and applications

High-end...

The market for server operating systems and their applications can be divided into different segments. We have to look at the high-end server market with different players, where the interdependence with hardware is inevitable because new hardware developments (such as the Itanium processor) consequently lead to various projects porting various operating systems to the new hardware.

... and low-end server market

The other segment is the low-end server market with a variety of applications. In this market, we have to look at the operating systems on one hand and on the server applications on the other hand. The applications include:

- Intranet servers (database servers, file servers, print servers),
- Multi usage servers (web servers, mail servers, streaming media servers, chat servers), and
- Connectivity servers (firewalls, gateways/routers, dialup servers).

Customers differ from desktop market

The customers in the server market are different from the customers in the desktop market. The decision makers have in general a basic IT knowledge and are corporate customers. Purchase decisions are made by management (small companies), IT managers (small and medium companies) or by IT departments (medium and large companies). The purchase decisions are made on criteria different from the desktop market. Therefore, the server market is driven by different critical success factors.

Most corporate customers buy solutions

Most corporate customers buy solutions rather than mere products. Hence, service companies, ISVs (independent software vendors) and VARs (value added resellers) are important distribution channels. Partnerships and co-operations with these groups are one critical success factor.

There is a discussion about the criteria that determine the purchase decision in the server market. Basically, they relate to the technical and functional quality of the software, but there are also other aspects like the number of skilled developers available.

Products and players in market for server operating systems

Important products and players in the high-end server operating systems market are: IBM (AIX, AS400, OS/390, OS/400), Fujitsu-Siemens (BS2000, BS2000/OSD), Sun (Solaris), HP (different Unix versions), Silicon Graphics (Irix), Compaq/Digital (Unix, VMS). The main products and players in the low-end server operating systems market are Microsoft (Windows plus server applications) and Unix in different variations and derivatives (one of which is Linux plus applications).

OSS influences on the server software market

Linux is supposed to offer several advantages and to constitute a serious alternative as server operating system due to the following advantages.

- High stability and high reliability,
- Low/zero license costs,
- Freely modifiable features, and
- Relatively small kernel.

Unix applications relatively easy to transfer

Furthermore, many applications for Unix systems are relatively easily transferable to Linux. Also many software application vendors are interested in having their Unix applications to run on Linux and vice versa as support of a single product is much cheaper than supporting different versions.

Thus, there are a number of factors and developments that support the hypothesis that Linux market share in the server market is going to rise further.

No restriction to Linux

In the server applications market, we are not only talking about Linux. It is the web server software Apache that leads the market of web servers on the public Internet with a market share in September 2001 of about 61 percent. Microsoft had a market share of 29 percent and iPlanet (Netscape) 2 percent (Netcraft, 2001). Besides the web server combination of Linux and Apache also the file server combination Linux and Sambahas significantmarket share.

Desktop/client operating systems and applications

Classical mass market

The market for desktop/client operating systems and their applications is the classical mass market. It is also the market that is mostly referred to when talking about software markets. The market is a worldwide market. It has a high tendency towards concentration and is dominated by Microsoft. Customers base their purchase decision on popularity, on interoperability with other systems and applications, and on number and quality of applications available for the operating system. In addition, they face switching costs due to training. Therefore, the market leader has a strong position.

Service companies and VARs (value added resellers) are an important sales channel for private and SME sales because many customers buy computer equipment with the operating system and the applications pre-installed. Hardware companies and their sales channels are important because of the integration of software and hardware (pre-installation) and the bundled sales. For software producers, the partnerships and co-operations with hardware producers and with service companies and VARs are critical success factor.

Major actors on the market

The main players and products with regard to desktop/client operating systems are (market shares according to IDC, 2000a):

- Microsoft (Windows) dominates with a 88 percent market share in 1999.
- Apple (Mac OS, Mac OS X) comes second with a significantly lower market share of 5 percent in 1999.
- Linux on the desktop has a market share of 4 percent in 1999.

Since the time this data was generated, the market share of Microsoft has even increased to well above 90 percent.

Segmentation of application markets

The applications in the desktop/client market can be segmented along several criteria. The most simple differentiation is into horizontal (the general office programs) and vertical (functionally specific or industry-specific) applications. As already mentioned, the market for applications is closely linked to the market for operating systems. Therefore, the players are basically the same. The market leadership for applications is held by Microsoft with its Office Suite, which also runs on many MacOS desktops. There are various OSS applications, which run on Linux or other OSS operating systems, but also on Windows.

OSS influence on the desktop/client software market

There are different possibilities of competition between the market leader Microsoft with its Windows and Office bundle and Open Source software. First of all, there is direct competition of bundles. For a long time Linux in combination with Open Source programs (e.g., OpenOffice) has been positioned as competitor. As these applications do, however, not provide the same level of user-friendliness and familiarity as the Microsoft family, strong competition is unlikely. The unequal positions are even strengthened by the fact that Microsoft products are sold pre-installed with most Intel-based computers and many (private and SME) users buy a bundle of hardware, operating system and applications.

This has changed with MacOS X which, although not purely Open Source, is based (in parts) on an Open Source Unix. Together with the available Microsoft Office suite, this product bundle of hardware, operating system and applications provides a stronger competition to the Wintel bundle. Although not purely Open Source, the advantages of the Open Source development model might make the Apple alternative more powerful and might increase Apple's chances to compete more strongly with Microsoft. The outcome of this stronger competition has to be awaited.

A further influence of Open Source can be in the field of desktop applications. Several larger desktop application projects, e.g. Mozilla or OpenOffice, have reached a stage, where it is expected that they can become strong competitors to the respective Microsoft products Internet Explorer and Office within one or two years.

Enterprise solutions

ERP systems and other enterprise applications

The major products in the enterprise solutions segment are ERP (enterprise resource planning) systems. But several other products belong to this category as well, e. g., CRM (customer relationship management) software, SCM (supply chain management) software, KM (knowledge management) software, groupware, e-learning software, etc. Worldwide, the main players are according to Hoch et al. (1999): IBM, Oracle, Computer Associates, SAP, HP, Fujitsu-Siemens, Hitachi, Parametric Technology, Peoplesoft, and Baan.

Customisation needed

Enterprise solutions almost always need customisation. Hence, the market is characterised by revenues that are based on product licences on one hand and service fees on the other hand. According to an internal McKinsey study, 30 percent of the installation costs for an ERP system account for the software product licence, 70 percent account for professional services fees to implement the product. (Hoch et al., 1999, 36). Some software companies offer the service themselves, some through partnerships. Customisation and installation projects typically take several months to be completed.

As this market is somehow between the products and the services market, we will include the market for enterprise solutions in the analysis of the services market.

3.2.2 The market for software-related services

Business dynamics in the market for software-related services

As (Hoch et al., 1999, 42) have found out in their analysis, the business dynamics in the market for software-related services differ in several ways from those for the software products market, mainly due to its service character.

Low entry barriers, constant threat of new entrants

The capital to be invested to start a service company is low. Instead, knowledge is the major foundation and entry barrier. Therefore, it is easy for a new company to set up if its founders have the technical knowledge to offer software services. Thus entry barriers are low, and incumbents constantly have to be aware of new entrants.

High pace of innovation

A high pace of innovation in the software products market leads to a similarly high pace in the services business. New technologies arise and with them new companies, which are able to offer the service. Some players in the services business are offering products as well.

Constant and significant marginal costs

Software services are a classical people-selling business with constant and significant marginal costs. In this sense, the services market is different from the product market. The cost of a second project are quite the same as the cost for the first project, even when a similar solution is implemented.

Human resources

Human resources are the most important asset in the services market. Therefore, it is important to attract capable human capital and to invest in human resources.

High fragmentation

Contrary to the product business, the services market is not ruled by the law of increasing returns. Therefore, the market faces much higher fragmentation and one can find:

- Small companies,
- Regionally focused companies, and
- Very few truly global players in service firms (e.g., Accenture, IBM Global Services, EDS, CSC, Cap Gemini Ernst & Young, KPMG, PricewaterhouseCoopers, Deloitte & Touche, Cambridge Technology Partners).

Market segments for software-related services

Two basic market segments

The services market consists of two basic market segments:

- IT consulting and systems integration as well as
- IT services.

The firms in the IT consulting and systems integration segment provide mainly customised services (solutions), including customised software products.

The IT services include support and maintenance, training and application management (including outsourcing). The firms in this field provide mainly these services, but they can sometimes also offer consulting and systems integration. The IT services seem to be stronger related to a product business. Many of the larger firms in this field are service units of hardware producers, sometimes even separate entities, such as Siemens Business Services or IBM Global Services. Others are primarily software producers where the focus on services becomes more important and generates a higher portion of revenues than the software sales.

There is no clear line between the two market segments, and many companies are active in both segments. The two segments are represented in the business structure of some firms in the service market. They usually have a business unit for “corporate solutions” (which includes consulting, implementation, and integration). The business

is project-related – normally a project is finished once a new solution is implemented and the customer is trained to work with it.

Major companies

Major firms active in these services markets are:

- The global service companies that usually offer services to large (global) enterprises.
  - Worldwide: Accenture, IBM Global Services, EDS, CSC, Science Applications, Cap Gemini Ernst & Young, Hewlett Packard, DEC, BSO Origin, Fujitsu-Siemens (Hoch et al., 1999, 27).
  - Germany: CSC Ploenzke, CAP Gemini Ernst&Young, Accenture, gedas, PricewaterhouseCoopers Unternehmensberatung, KPMG Consulting, Plaut Gruppe, Atos Origin, IBM Deutschland, Siemens Business Services, T-Systems, Hewlett-Packard, Lufthansa Systems Group, GE CompuNet Computer (Lünendonk, 2000)
- Thousands of small service firms, either focused on specific solutions and technology or on their home region. The latters' main customers are often SMEs.

OSS influence on the services market

With regard to the global services companies and their large integration projects – such as enterprise solutions – OSS will not yet have a significant influence. The OSS-related discussion is mainly focused on the question whether the main firms in the market of enterprise solutions will make their solutions portable on Linux. (SAP, IBM, Oracle, Software AG, Sybase all have already done so with major ERP applications.)

With regard to small service companies, however, OSS has a significant influence. Firstly, there are service companies that focus exclusively on Linux and other OSS. Secondly, there are distributors of OSS that offer services and support for their products. Thirdly, there are specialised service companies that shift to OSS or include Linux and other OSS into their spectrum. Strengths and weaknesses as well as the threats and opportunities of the three groups in relation to each other will be analysed in section 4.2.3.

3.2.3 The market for embedded software products and services

Business dynamics in the embedded software market

Customers are producers of special devices

The major difference between the software products and embedded software is the customer. In the market for embedded software, the buyers of software are not the end-users, but the producers of different kinds of devices. Hence, the direct customers of embedded software are software developers who need to work with embedded software products and develop their customised version of software to ensure functionality in any kind of device. Therefore, the market for embedded software products is a classical B2B market. The software companies that supply embedded software are the suppliers for the device producers.

Major business dynamics

We will limit the analysis to embedded products and, in this first step, to the embedded operating systems in particular. The business dynamics in the embedded software products market are similar to the dynamics in the non-embedded market. The major business dynamics can be summarised as follows:

- The indirect network effect is strong because of the interoperability and portability of software and because of the skills of developers. The less common any embedded system tool is, the fewer common components can be used and the fewer people are able to handle the embedded software. The network effect is especially strong for operating systems, which form a basis for many other software applications and software development tools.
- The business is knowledge-driven and the capital investment is low, which results in high innovation rates. However, the network effect establishes a high entry barrier.
- The marginal costs are low and result in worldwide markets. With non-OSS, the fixed costs for development are high and result in the condition that many copies must sell in order to have a profitable business.
- In the long term, the market position is never stable because of technology switches.

Analysis of the embedded software market

Home-grown operation systems with traditionally large market share

Traditionally, the largest share of the embedded operating systems and kernel market is occupied by so-called "home-grown" operating systems. These are operating systems, which are privately developed and maintained. They used to account for between one half and two thirds of all embedded systems' operating systems (Evans Data Corporation, 2001). According to a survey among 500 developers in 2001, home grown systems are still the most widely used systems, followed by WindRiver's operating systems and DOS. Embedded Linux, as an open operating system, is already number four.

Challenges for market leaders

The home-grown systems face several difficulties, which will become even stronger in the near future. Basically, applications in the embedded market are getting much more complicated, and networking is becoming more important, which makes developing and maintaining these private systems much more expensive. Therefore, it is predicted that many of the home-grown operating systems will switch to Linux in the near future. (Evans Data Corporation, 2001) Half of the home-grown systems operate real-time, half of them are smaller, non real-time systems. A real-time operating systems needs to guarantee that a given operation will be performed within a given time window. Hence, the critical factor for real-time is determinacy.

Advantages for Linux ...

Linux is supposed to have several advantages compared to the home-grown systems (Cook, 2000):

- It is standardised and Open Source. It has a modular structure and can be trimmed down for several purposes. Hence, the features of Linux are freely modifiable for various reasons.
- It has a low price.
- There are many skilled programmers available on the labour market.
- Compared to private operating systems, there are many drivers available. (Compared to Windows, of course, not so many.)

... and disadvantages

However, there are also several disadvantages:

- Linux is not designed to be a real-time system. Although there are various projects to develop components to make Linux real-time for various situations, the results are not comparable to operating systems that have been designed for real-time performance.
- Compared to other proprietary embedded operating systems, there is still a lack of drivers for Linux.
- Linux is relatively large by embedded standards. However, it can be trimmed

down and there is a trend for hardware products becoming cheaper. And the kernels of its major competitors offering proprietary software are also quite large.

- Embedded systems are often mission-critical for business processes. Here, the perceived lack of professional 24h support is a disadvantage for Linux.

OSS influence on the services market

Linux constitutes an alternative for home-grown systems when their operations not necessarily have to be real-time and when their operations are not mission-critical. A number of Linux embedded products is expected, such as set-top boxes, cable TV, toys, car devices, manufacturing devices, in the long-term also house-hold devices. (IDC, 2000a) Here, Linux has potential to gain market share.

The major suppliers of embedded Linux are seen to be Lineo, MontaVista and Red Hat. (VDC, 2001)

4 Business models based on Open Source software

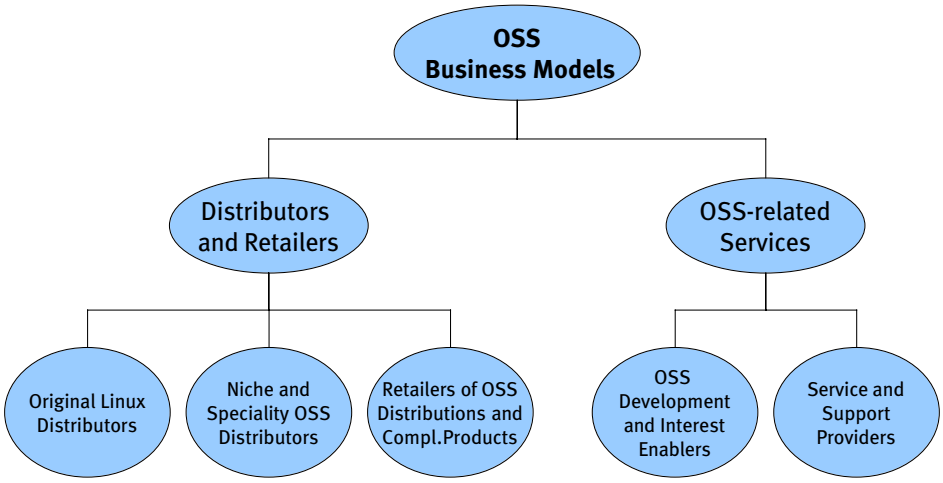
4.1 Overview

In our analysis, we identified several types of business models that are based on the OSS technology. Of course, in reality, many companies operate in more than one field, for example the distributors. Although their main business is marketing and distribution of products, a major portion of their income is based on additional services such as consulting and support.

Focus on business models that are purely based on OSS

The focus of the subsequent analysis is on business models that are purely based on OSS. This means they would not exist without the occurrence of the OSS phenomenon. Similar to the software market analysis, we distinguish between product-related businesses and service-related businesses. Figure 4-1 provides an overview of the existing business models. Service and support providers offer consulting, system implementation and integration, support, training, recruiting and staffing services.

Figure 4–1
OSS business models



Berlecon Research 2002

Questions for analysis

The business models will be analysed in the following section with regard to the following questions:

- What do the companies do (main product or service offer)?
- What is the market (customer and competitive analysis)?
- Why do they earn money (Why not)?

4.2 Distributors and retailers

4.2.1 Original Linux distributors

Product and service offerings

Examples

The Linux distributors package and sell their own version of the Linux operating system. Examples for original Linux distributors are Red Hat, SuSE, MandrakeSoft, Caldera, Turbolinux, and Slackware.

To end users (consumer or corporate), they sell the Linux operating system in various software packages and bundles such as server applications, desktop applications, e-commerce suites, often in variations for various hardware. To IT administrators, they sell appropriate administration tools for their applications. To developers, they sell development tools and usually also various Linux versions for OEMs (original equipment manufacturers), which use the operating system as a basis for their hardware-software combination packages or embedded systems.

Elements in Linux distributions

A Linux distribution consists of the Linux kernel and several hundred additional files that together form the Linux operating system. In order to develop their own versions, the distributors need to collect the newest Linux release and all the related files. The second step is testing, tuning and optimising the existing software fragments working together with the aim to achieve a good performance and reliability. These efforts are normally returned to the OSS community. In a third effort, means for smooth installation, good documentation, efficient management and productivity tools are created. For these steps, the Linux distributors support the Linux community by providing development laboratories. Additionally, they employ several developers or work with freelance developers.

Pros and cons of being distributor

On one hand, Linux distributors save an enormous amount of software development costs because they do not have to develop their operating system from scratch. For example, the Red Hat 7.1 operating system is estimated to have cost $1 billion (Wheeler, 2001a). However, there remains a significant investment for the development of their optimised Linux versions. On the other hand, the Linux distributors cannot price their products as freely as proprietary software producers can. The several components of Linux can be downloaded freely and as many times as possible from several project websites. Therefore, the value added by distributors on the product level is mainly the packaging.

The software products based on the Linux version are eventually offered on a website for download or on a CD-ROM for physical distribution. The Linux distributors use several sales channels – the most important ones are the VARs (value added resellers) and retail chains (in particular bookstores).

Critical success factor

A critical success factor in the Linux distribution business is brand building. Hence, the distributors heavily invest in marketing (advertising, trade fairs, public relations). Distribution and Marketing are the core competences of the Linux distributors.

Additional services

Despite this fact, most of the distributors provide additionally Linux-related services such as consulting, integration, support and training. The services generate an additional income stream. Additionally, the distributors generate a small income stream through merchandising – the selling of T-shirts, mugs etc.

The market for Linux distributors

Two market segments covered by distributors

Basically all of the Linux distributors cover two market segments: First, the mass market with standardised packages offered to SMEs and private consumers. Second, the market for individual solutions, which are offered to medium to large corporate customers.

The mass market for operating systems separates into the server and the desktop market. In the server market, OSS is supposed to offer several advantages and to constitute a serious alternative as server operating system. The major competitors are Windows NT and the various other Unix systems. In the desktop market, the Linux market share is very small. Here, Linux has one major competitor: Microsoft with its various Windows versions. The question for the coming years is whether the Linux operating system will be successful on the desktop or not.

The solutions market (service-related) is completely different from the mass market (product-related). Most of the distributors consider the solution business as increasing and profitable income stream. Some of them cover the solutions market through partnerships with consulting companies. On the other hand, companies like SuSE have build up their own profit centre for corporate users.

Advantages and disadvantages of business model

Low margins for distributors

In the mass market, Linux itself can be regarded as a commodity because its components can be freely downloaded or copied. The packaging, which is done by every distributor and results in the different Linux versions, constitutes the added value. However, the margins per unit sold are not very high. The product business with Linux distributions is a mass market and the active companies will have to increase their product sales (and eventually their market share) to become profitable.

Differentiation by branding...

Therefore, the distributors are forced to develop other means to differentiate themselves from each other and from proprietary competitors. So far, this differentiation has been mainly achieved by branding, a critical success factor in the mass market. Linux distributors such as Red Hat or SuSE put much effort into marketing (advertising, trade fairs, even certification of their Linux trainings can be regarded as a brand building effort).

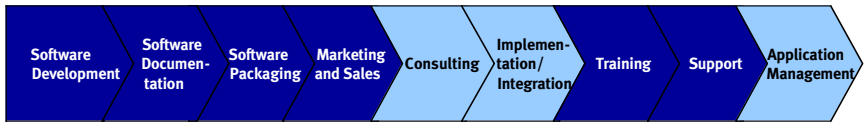
...and exclusive sales channels

The second success factor is gaining access to sales channels, which are bookstores and VARs. Mandrakesoft, for example, exclusively cooperates with Macmillan bookstores in the US. In Germany, the company is trying to intensify partnerships with small consulting companies and integrators to gain access to the SME segment.

Distributors shifting towards solutions and consulting

As it is difficult to survive on the product business alone, the distributors build up a second business by shifting towards the solutions and consulting market. This might be interpreted as a move towards a more lucrative business as pure software retailing provides only low margins because the marketing costs are high and, in the desktop market, the number of potential buyers of Linux box products is still low compared to the buyers of Microsoft box products. Due to their software knowledge from packaging and optimising the Linux parts, the distributors certainly have the OSS as well as technical competence to build up consulting and service business. But it is in question whether they already have the know-how in consulting and business processes to become serious competitors for existing service or consulting firms.

Figure 4–2 Primary focus of distributors in the software value chain



Berlecon Research 2002

4.2.2 Niche and specialty OSS distributors

Product and service offerings

Specialised on OSS other than operating systems

The niche or specialty distributors develop and distribute different OSS but no operating systems. Their products include applications, development and administrative tools. Normally, their software is developed to run on Linux, but some products also run on Windows or other operating systems. Examples are Zope (formerly Digital Creations), Sendmail.com, Covalent Technologies, Cygnus (acquired by Red Hat), Precision Insight (acquired by VA Linux), MySQL, ActiveState, and CollabNet.

Companies live symbiotically off an OSS project

In this business model, companies live symbiotically off an OSS project. OSS is collected, maintained and/or developed. The main function of those companies is to coordinate the scheduling and make a commitment to delivery and support of a dedicated product. Normally, they employ some of the core developers of the specific project and rely heavily on their relations to the developer community.

Some examples in detail: Zope offers an OSS applications server, a platform to develop CMS (content management systems) among others, and several additional tools to add task-specific or industry-specific features. ActiveState offers proprietary development tools for Linux development, CollabNet proprietary software development management tools. Precision Insight offers OSS server tools to support graphics hardware. Covalent Technologies offers an optimised version of the Apache web server. Sendmail.com offers a message server and a version of its software to be embedded in various products. Similarly, MySQL offers its database software in a version for commercial use and to be embedded in other software products.

The market for niche and speciality OSS distributors

Market share often high

The market share for some OSS products such as Samba or the Apache web server is rather high. For example, a web server survey conducted by Security Space counting the web servers across all domains, reports an Apache market share of 63 percent in October 2001. Similarly, a web server survey conducted by Netcraft counting active server across all domains reports an Apache market share of 61 percent in October 2001. However, this does not say much about the success of the companies that are trying to profit from this development.

Customers often VARs or OEMs

The market access for the niche and specialty distributors is very different from the access for the Linux distributors. Normally, they do not directly target private consumers or SMEs with their products. The major Linux distributors can easily incorporate specialty software components into their out-of-the-box packages and, because of their stronger brands, sell it to a broad customer base. Nevertheless, some specialty distributors offer a limited number of packages that can be ordered on their websites and are distributed through the same channels as the Linux distributions (e.g. VARs and retail chains).

Hence, the major customers of specialty distributors must be either VARs (Value Added Resellers) or OEMs (Original Equipment Manufacturers), which sell optimised hardware-software bundles or develop and sell embedded products.

Advantages and disadvantages of business model

Because the companies in this category develop and distribute OSS, which can normally just be downloaded and copied, it is hard to imagine what kind of direct business model they could use. Indeed, simply selling software does not propose a unique selling proposition because access to OSS is largely unrestricted, at least in most license models.

Variety of income streams based on complimentary products

As a result, there are a variety of income streams summarised in this model. Most of the companies sell additional services for their product (consulting and support). Some of them, for example MySQL, generate income from license fees for commercial licenses for the normally GPLed MySQL. Others, for example Precision Insight, decide not to disclose the source code for the newest version of their products, but only for previous versions. Another way, used by Sendmail.com, is to develop proprietary commercial software on top of the basic sendmail functionality. Most of these ideas are critical because the companies leave the field of pure OSS players and become a player in the traditional software business. They have to be able to live in both worlds.

4.2.3 Retailers of OSS distributions and complementary products

Product and service offerings

Sales channels for software and documentation

The retailers are major sales channels for the distributors. They either sell the distributors' software products or they provide and sell additional documentation and information on OSS products or merchandise. The retailers are not involved in the software development process. Instead, their core competence is distribution or, in some cases, publishing. The retailers are not solely focused on OSS. For Lehmanns Fachbuchhandlung or mitp, for example, OSS training and documentation books are just one part of their retailing or publishing business. O'Reilly is the only known company, whose business is almost solely based on OSS. O'Reilly is selling documentation and training books.

Other examples for publishers and retail chains include CNET, ZDNet, and Fona (Denmark). In addition, there are several specialised Linux shops like CrazyPenguin (UK), Linuxland (Netherlands), Linux Central (U.S., part of INT Media Group), and Linuxbutikken (Sweden, Norway).

The market for retailers and publishers

Firms target the mass market

The retailers and specialised Linux shops target the mass market only. Their customers are private or corporate users, developers or IT-administrators. For OSS, the market is slowly shifting to users that are not "software freaks" or developers, but instead use the software.

Advantages and disadvantages of business model

Access to customers and known brand

The advantage of established retailers is their access to customers via their retail stores. Another advantage is the known brand of retail chains. Web-based resellers and specialised Linux shops will find it difficult and expensive to create a brand awareness, especially when they compete directly with the distributors and publishers.

Merchandising, in general, is not a business model itself but only an additional income stream whenever a strong brand has been established. Therefore, income from merchandising is primarily interesting for the Linux distributors.

Need for documentation

There was a need for documentation as OSS became more popular and was applied outside the OSS community. For commercially developed software, the documentation is normally done by the software producer or in cooperation with a publisher. O'Reilly combined its OSS knowledge with publishing knowledge and succeeded in establishing a brand for OSS books. Because the company covers many OSS projects, it is not dependent on a single software development.

4.3 OSS-related services

4.3.1 OSS development and community enablers

Product and service offerings

This category includes primarily two different sorts of actors. These are first of all marketplaces like SourceXchange, Cosource.com, intraDAT (vshop.org) and secondly conference and trade fair organisers like LogOn Technology Transfer or Linux New Media.

Marketplaces

The function of exchanges or marketplaces is to match potential buyers (organisations or individuals looking for "needed improvement") and sellers (OSS developer community). The software produced would be customised or build-to-order OSS. The main argument for the potential of these exchanges is the assumption that many software developers want to decide themselves what project to work on (which is not possible as an employee). Additionally, the global reach of the Internet could leverage the developer potential all around the world and possibly even drive the prices down.

Improving development process with additional services

The marketplaces for software development offer the matching service and improve the development process through provision of a project manager and productivity tools. Even demand aggregation can be a service: Multiple buyers with the same problem aggregate their funds via the marketplace to get a software solution. As far as known, no company in this business has become profitable so far. SourceXchange closed in April 2001.

Conference organisers – the other business model in this group – enhance interest in OSS projects and provide opportunities for the OSS community and business partners to meet. The organisers either are specialised in OSS and Linux or are general conference organisers that generate part of their income through their focus on OSS. There is some relation to the publishing business and consequently publishers like O'Reilly are also active in organising Open Source conferences.

The market for OSS marketplaces and conference organisers

Chicken-egg problem for marketplaces

Marketplaces address two groups, the buyers and the suppliers of software. The "supplier" group is the OSS developer community, the "buyer" group are corporate users or developers, who look for a specific solution. Normally, the buyers would be willing to pay for a service like those provided by the marketplaces. For the buyers, however, any marketplace is only interesting if there is a sufficient number of interested sellers, e.g. developers. So OSS marketplaces are facing a chicken-egg-problem like many marketplaces in other areas.

Potential lack of sponsors for conference organisers

Conference and trade fair organisers address mainly the OSS community. This can become a problem because the business model does not really conform to the OSS philosophy. Organisers in the traditional conference business address a group of buyers and sellers who are interested in product or technology information and know-how. They generate most income from the sell-side because sellers use the events for marketing their products. Additionally, buyers or people interested in gaining information pay entrance fees for conferences (high fees) or trade fairs (lower fees). It is questionable whether the market for conference organisers in the OSS business is attractive.

Advantages and disadvantages of business model

Marketplaces

So far, the pure marketplaces and exchange models have failed. Probably the value added is not sufficient to carry a business model purely on the matching function. Revenues can only be generated from the demand side as the developers are probably not willing to pay for the service.

On the demand side the "buyers" of the software might not have trust in the completion of the projects. A normal software development project is usually done with a subcontractor according to a contract with exact specifications on what needs to be done. These projects often exceed the cost and time in the contracts. With regard to these experiences, it seems unlikely that a company would trust a developer community with no or vague responsibilities and certainly no guarantees for completion of projects. On the supply side, the main competition of these business models is the OSS community itself and all the projects that are managed by volunteers.

Potentially useful as complimentary service

The matching function could rather be used as an additional service in the spectrum of a service company. Some distributors, for example MandrakeSoft, apply part of this model in their support offerings. They pay OSS developers and specialists a certain amount of money to solve specific problems.

Conference organisers

So far, OSS conferences and fairs have usually had comparatively low prices and it is questionable whether the organisers can operate profitably. They cannot demand high entrance fees for conferences because most people interested are OSS community members. Nor can they demand high fees from software suppliers because many suppliers are community projects or small and regionally focused service companies. At least, due to constantly rising interest in OSS, the trade fairs can become a more attractive field because of rising visitor numbers.

Conference organisers are very dependent on the interest in a special subject. Their core competence, however, is expert knowledge for the evaluation of interesting topics, finding interested people, and the project management of a conference or trade fair. With regard to those competencies, however, conference organisers do not need to restrict their business model to OSS. Although their knowledge can be OSS-limited – in order to evaluate OSS topics, they need to have knowledge about the software market in general.

Reputation as entry barrier

An entry barrier for potential entrants in the OSS conference market can be reputation within the OSS community. Some OSS conference organisations have already gained significant reputation in the community. However, as mentioned before, the critical factor in the OSS conference business is that a significant part of interested visitors is not willing to pay the corporate entrance fees that are paid on most B2B events.

4.3.2 OSS-related services and support

Whole variety of services offered by companies with different backgrounds

OSS-related services and support include several services such as consulting, systems integration, support, maintenance, remote administration, training, and application management. The companies in the OSS-related services market differ according to their background. First of all, there are companies that have a background in Linux or other OSS products. They are trying to establish services that build on their product knowledge. Hence, their core competence is the technological and product knowledge. Most of them offer a full range of services. Linux distributors, niche and specialty distributors as well as independent OSS service companies belong to this category.

And secondly, there are companies that have special process knowledge in how to provide a service related to IT in general. This can be knowledge in IT consulting, systems integration, IT-training or IT-recruiting, sometimes even with a vertical functional or industry-specific specialisation. They can extend their offerings to OSS-related services.

Examples

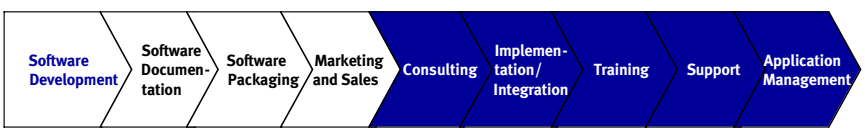
Examples for full service companies offering various services based on OSS knowledge are:

- Linux distributors (e.g., Red Hat, SuSE, Caldera, MandrakeSoft, Turbolinux).
- Niche and specialty distributors (e.g., Zope, MySQL, Sendmail.com, Covalent Technologies),
- Independent OSS service companies (e.g., Linuxcare) and many small service and integration companies with special technological OSS and particular Linux knowledge (e.g., Linux Information Systems, B-connected).

Examples for special services based on integration and service knowledge extended to OSS are:

- For consulting and systems integration: (Global System Integrators; Accenture, KPMG, PricewaterhouseCoopers etc.), various small consulting and integration companies.
- For training: various training and e-learning companies (e.g., Microconsult)
- Recruiting and Staffing services: various IT-specialised recruiting companies (StepStone-IT, Monster.de, JobUniverse.de)

Figure 4–3 Primary focus of service companies in the software value chain



Berlecon Research 2002

Product and service offerings

Critical factor often Linux experience

Consulting companies and system integrators help their customers realise IT strategies based on business needs as described in section 2.2. The critical factor is often Linux expertise: The small integrators and service companies do usually have a background in OSS development and are trying to establish a business based on services. The large integrators can activate their Unix people to acquire Linux expertise or hire OSS developers.

Support companies offer their services in various models, as the "classical" support model of OSS (send a bug report to the project's community) is not accepted by many business customers. Commercial support gives businesses the possibility to have their OSS products supported without getting involved with the developer community culture (even if the support staff are developers, e.g. at MandrakeSoft). As shown by IDA/Unisys (2001), support contracts vary in terms of...

Variation of support contracts

- Contract categories: installation support (for a limited time after installation), support packages (price per call), annual support contracts.
- Way of support: telephone hotline, e-mail hotline.
- Level of support: 1st level (smaller problems, end-user targeted), 2nd level (administrator), 3rd level (developer targeted, sometimes including source code changes) support definition and coverage.
- Coverage hours and days: 10x5 to 24x5 or 24x7, working days only or all 365 days/year.
- Reaction time: from 1 hour in the best case to 8 or 16 hours (or "next business day").
- List of supported products: hardware and/or software.
- Personalisation – inclusion of individual consultancy and auditing.
- Patch- and update management (in case of new version).
- Supported infrastructure type, from desktop-PC to mainframe.

OSS focused training companies have two choices offering their courses: They can offer classical seminars with physical attendance (often after a certification process by the software vendor) or e-learning solutions (e.g. Red Hat).

The market for OSS-related services and support

Customers for systems integration,...

The customers for systems integration range from small to large corporations, which pay for a solution instead of paying for a product. Hence, the service is project-related.

...support, ...

Support is needed in any market and on any user level. For example, OEMs (original equipment manufacturers) and ISVs (independent software vendors) can be customers of Sendmail.com support when incorporating Sendmail into their product. System administrators do usually need support when a new product is implemented. But also private and business users need support with their product (which is normally offered in a standardised way).

...and training

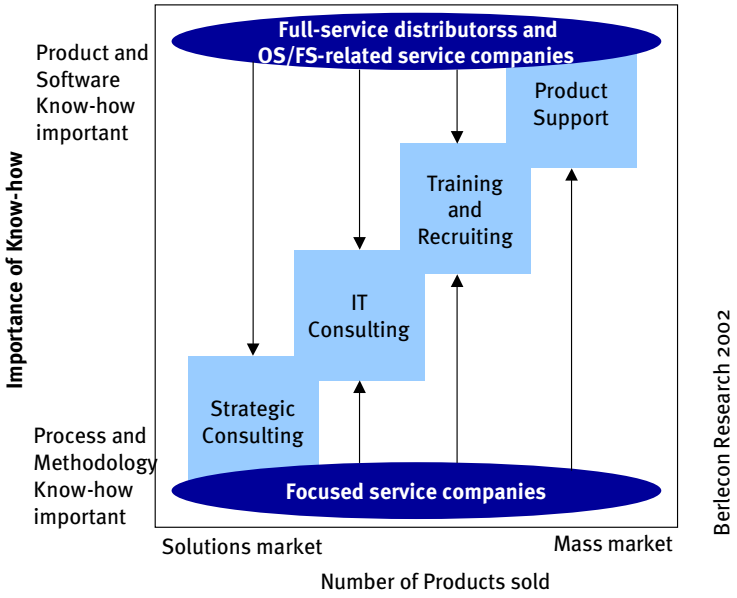
Customers for OSS training are users on various levels: Red Hat, for example, is offering courses for users, systems administrators and developers in classical seminars as well as in e-learning courses. Customers typically are business-related users. Training products at Red Hat have one focus on the Red Hat Linux distribution and related software; under "E-Business" they offer a course for SAP-Red Hat integration. (In their e-learning courses, they also offer C/C++/Java programming and general Unix/networking courses.)

Advantages and disadvantages of business model

Success depends on importance of product know-how vs. process know-how

As already mentioned, there are two fundamentally different groups of firms active in the OSS-related services market. Firms with OSS background have substantial product and technology knowledge, which they use to build up their services business. Businesses solely based on OSS products are dependent on the acceptance and development of OSS. Companies without OSS background have substantial process know-how in the services sector. They attempt to extend their offering to OSS-related services. Whether firms from one group or from the other group will succeed, depends on the importance of product know-how vs. process know-how in the separate service fields. Figure 4-4 illustrates this relation.

Figure 4–4 Areas of success in the OSS-related service business



Strategic consulting vs. product support

Strategic consulting is a service field where methodology and process know-how are extremely important, product know-how is less important or, for a special project, can easily be acquired. Product support, on the other hand, requires primarily product know-how whereas knowledge about the support process can as easily be acquired as process know-how in the consulting business.

The firms with OSS background will be mainly successful in areas where product know-how is important and process know-how can be easily acquired. This is the case with support and training offerings. The players without special OSS know-how will be mainly successful in areas where this know-how plays only a minor role or can easily be acquired.

5 References

5.1 Literature

Balzert, Helmut (1996): Lehrbuch der Software-Technik: Software-Entwicklung, Heidelberg.

Balzert, Helmut (1998): Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung, Heidelberg.

Bezroukov, Nikolai (1999a): A second look at the Cathedral and the Bazaar, in: First Monday, Vol. 4, No. 12, 1999, http://www.firstmonday.dk/issues/issue4_12/bezroukov/index.html.

Bezroukov, Nikolai (1999b): OSS Development as a special type of academic research (a critique of vulgar Raymondism), in: First Monday, Vol. 4, No. 10, 1999, http://www.firstmonday.dk/issues/issue4_10/bezroukov/index.html.

Blase, Paul (2000): Open Source White Paper – A Software Company's Dilemma. DiamondCluster, http://www.diamondcluster.com/work/Wpapers/WPSoftware.asp.

BMWi (2001): Open-Source-Software – Ein Leitfaden für kleinere und mittlere Unternehmen, http://www.bmwi.de/Homepage/download/infogesellschaft/Open-Source-Software.pdf.

Claybrook, Bill (2001): AberdeenGroup InSight: Linux is on the Move Up!. Aberdeen Research, July 24, 2001, http://www.ibm.com/linux/LinuxInSight.pdf.

Cook, Rick (2000): Embedded Linux Basics. LinuxWorld.com, May 2000, http://www.linuxworld.com/linuxworld/lw-2000-05/lw-05-embedded.html.

Dafermos, George N. (2001): Management and Virtual Decentralised Networks: The Linux Project, in: First Monday, Vol. 6, No. 11, 2001, http://www.firstmonday.dk/issues/issue6_11/dafermos/index.html

de Icaza, Miguel (2001): Is Linux ready for the corporate desktop?, in: Computerworld, August 9, 2001, http://www.computerworld.com/storyba/0,4125,NAV47_STO62924,00.html.

Delio, Michelle (2000): It'll Be an Open-Source World, in: wired.com, August 15, 2000, http://www.wired.com/news/print/0,1294,38240,00.html.

Dyck, Timothy (2001): MySQL, in: CNET.com, March 5, 2001, http://enterprise.cnet.com/enterprise/0-9513-707-6618957.html?tag=st.it.9500-704-0.sr.9513-717-6716336-7053536.

Eduak, Thomas (2001): Nur Fenster nach Windows-Norm? In: Computerwoche Nr. 44/ 2001, p. 38, October 19, 2001.

EITO (2001): European Information Technology Observatory 2001.

Evans Data Corporation (2001): Embedded Systems Developer Survey 2001, August 2001, http://www.evansdata.com.

Farbey, Barbara; Finkelstein, Anthony (2001): Evaluation in Software Engineering: ROI, but more than ROI". Paper submitted for "Third International Workshop on Economics-Driven Software Engineering Research" (EDSER-3 2001), http://www.cs.virginia.edu/~sullivan/edser3/finkelstein.pdf.

Forschungsstelle für Verwaltungsinformatik, Universität Koblenz-Landau (2000): Protokoll der Veranstaltung 'Open Source Software (OSS) in der Bundesverwaltung' der Koordinierungs- und Beratungsstelle für Informationstechnik in der Bundesverwaltung, 26.-28. September, http://linux.kbst.bund.de/auftakt/ergebnis/oss-protokoll.pdf.

Freyermuth, Gundolf S. (2001a): Offene Geheimnisse. Aus der Open-Source-Geschichte lernen, Teil I, in: c't 20/2001, p. 176, http://www.heise.de/ct/01/20/176/default.shtml.

Freyermuth, Gundolf S. (2001b): Offene Geheimnisse. Aus der Open-Source-Geschichte lernen, Teil II, in: c't 21/2001, p. 270, http://www.heise.de/ct/01/21/270/default.shtml.

FSF/ GNU: Categories of Free and Non-Free Software, http://www.gnu.org/philosophy/categories.html.

Gertz, Winfried (2001): Das Consulting-Geschäft um Open Source, in: Computerwoche, April 27, 2001.

Ghosh, Rishab (1998): Cooking pot markets: an economic model for the trade in free goods and services on the Internet, in: First Monday, Vol. 3, No. 3, March 1998, http://www.firstmonday.org/issues/issue3_3/ghosh/index.html.

Ghosh, Rishab; Prakash, Vipul Ved (2000): The Orbiten Free Software Survey, in: First Monday, Vol. 5, No. 7, July 2000, http://www.firstmonday.dk/issues/issue5_7/ghosh/index.html.

Goder, Andrea (2001a): Entdeckung der Linux-Welt, in: Computerwoche Special, 02/2001.

Goder, Andrea (2001b): Pinguine und Pleitegeier: Open-Source-Firmen in der Konsolidierungsphase. Computerwoche Special, 02/2001.

Hall, Mark (2000): Start-up's Apps Make Desktop Linux Easier, in: Computerworld, September 9, 2000, <http://www.computerworld.com/cwi/story/0%2c1199%2cNAV47_STO63615%2c00.html.

Hecker, Frank (1999): Setting up Shop, http://www.hecker.org/writings/setting-up-shop.html.

Hoch, Detlev J. et al. (1999): Secrets of Software Success: Management Insights from 100 Software Firms around the World. Boston, Massachusetts, 1999.

Hoffmann, Naomi (1999): Open Source Software, http://public.kitware.com/VTK/pdf/oss.pdf.

IDC (2000a): Linux: What's the use? Western Europe 1999-2004.

IDC (2000b): IDC IT Forecaster: Computer Operating Environments Coevolve, August 8th, 2000, http://www.idc.com/itforecaster/itf20000808.stm.

Jaeger, Till (2001): Klage wegen GPL-Verletzung, August 13, 2001, http://www.ifross.de.

Jaeger, Till (2000): Copyright oder Copyleft, in: Computerwoche Spezial, 4/2000, p. 36, , http://www.ifross.de/ifross_html/art6.html.

Johnson, Justin Pappas (2000): Some Economics of Open Source Software. December 2000, http://www.idei.asso.fr/English/EPresent/.

Kaven, Oliver (2001): Performance Tests: File Server Throughput and Response Times, in: PC Magazine, November 13, 2001, http://www.pcmag.com/print_article/0,3048,a%253D16554,00.asp.

KBSt (2000): KBSt-Brief Nr. 2/2000: Open Source Software in der Bundesverwaltung, http://linux.kbst.bund.de/02-2000/.

Köppen, A.; Nüttgens, M. (2000): Open Source: Strategien für die Beratung. In: Scheer, A.-W.; Köppen, A.: Consulting – Wissen für die Strategie-, Prozess- und IT-Beratung, Saarbrücken, pp. 231-242, http://www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/Consulting/OpenSourceStrategien.pdf.

Lerner, Josh; Tirole, Jean (2000): The simple Economics of Open Source, December 2000, http://www.univ-tlse1.fr/idei/Commun/Articles/Tirole/Simple-Economics.pdf.

Lünendonk (2000): IT-Beratungs- und Systemintegrations-Unternehmen in Deutschland, http://www.luenendonk.de/html/marktanalysen_listen.html.

Miller, Robin (2001): Learning from Mozilla's mistakes. NewsForge, October 23, 2001, http://www.newsforge.com/article.pl?sid=01/10/20/1841215&mode=thread.

N.N. (2001a): Star Office 6 löst sich vom Microsoft-Vorbild, in: Computerwoche Nr. 42/ 2001, pp. 26-27, October 5, 2001.

N.N. (2001b): Sprung auf den Desktop für Linux zu hoch?, heise online News, May 15, 2001, http://www.heise.de/newsticker/result.xhtml?url=/newsticker/data/odi-25.05.01-002/default.shtml&words=Sprung%20Desktop.

Netcraft (2001): Netcraft Web Server Survey, October 2001, http://www.netcraft.com/survey.

Nüttgens, M.; Tesei, E. (2000a): Open Source – Konzept, Communities und Institutionen, in: Scheer, A.-W. (Editor): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 156, Saarbrücken, http://www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/heft156/heft156.pdf.

Nüttgens, M.; Tesei, E. (2000b): Open Source – Marktmodelle und Netzwerke, in: Scheer, A.-W. (Editor): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 158, Saarbrücken, http://www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/heft158/heft158.pdf.

Nüttgens, M.; Tesei, E. (2000c): Open Source – Produktion, Organisation und Lizenzen, in: Scheer, A.-W. (Editor): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 157, Saarbrücken, http://www.iwi.uni-sb.de/nuettgens/Veroef/Artikel/heft157/heft157.pdf.

O'Reilly (1999): Open Source; kurz & gut, Köln.

Open Source Initiative: The Halloween Documents, http://www.opensource.org/halloween/.

Osterberg, Jürgen (2001): Flotter Auftritt mit Zope, in: Computerwoche Nr. 44/ 2001, p.26, October 19.

Rasch, Chris (2001): The Wall Street Performer Protocol: Using Software Completion Bonds to Fund Open Source Software Development, in: First Monday, Vol. 6, No. 6, June, http://www.firstmonday.dk/issues/issue6_6/rasch/.

Raymond, Eric (1997): The Cathedral and the Bazaar, http://www.tuxedo.org/~esr/writings/cathedral-bazaar/.

Raymond, Eric (1998): Homesteading the Noosphere, http://www.tuxedo.org/~esr/writings/cathedral-bazaar/.

Raymond, Eric (1999): The Magic Cauldron, http://www.tuxedo.org/~esr/writings/cathedral-bazaar/.

Rosenberg, Donald K. (2000): Open Source: The Unauthorized White Papers, Foster City.

Sandred, Jan (2001): Managing Open Source Projects: A Wiley Tech Brief.

Schenk, Thomas: Linux: Its history and current distributions, http://www.developer.ibm.com/library/articles/schenk1.html.

Schmitz, Ludger (2001): The use of open source software in the public sector, http://europa.eu.int/ISPO/ida/?http&&&ag.idaprog.org/Indis35prod/doc/333.

Security Space (2001): Security Space Web Server Survey, October 2001, http://www.securityspace.com/s_survey/data/200110/index.html.

Succi, Giancarlo; Paulson, James; Eberlein, Armin (2001): Preliminary Results from an Empirical Study on the Growth of Open Source and Commercial Software

Products. Paper submitted for "Third International Workshop on Economics-Driven Software Engineering Research" (EDSER-3 2001), http://www.cs.virginia.edu/~sullivan/edser3/paulson.pdf.

Symons, Charles (2001): Controlling software contracts, http://www.gifpa.co.uk/library/Papers/Symons/esepg97.html.

TechConsult (2001): Fast jedes 5. Unternehmen setzt 2003 auf Linux. Press Release, http://www.techconsult.de/de/Services/studiendetail.cfm?id=82.

Tuomi, Ilkka (2001): Internet, Innovation and Open Source: Actors in the Network, in: First Monday, Vol. 6, No. 1, January 2001, http://www.firstmonday.dk/issues/issue6_1/tuomi/index.html.

Venture Development Corporation (2001): Linux's Future in the Embedded Systems Market (Summary for the Embedded Linux Community), June, http://www.linuxdevices.com/articles.

Wayner, Peter (2001): Open source databases bloom, in: Computerworld, September 9, 2001, http://www.computerworld.com/storyba/0,4125,NAV47_STO63629,00.html.

Weiss, George (2001): The Future of Linux and Open Source. Gartner, Strategy & Tactics/Trends & Direction, Note Number AV-13-9850, June 20, 2001.

Wheeler, David (2001): More Than a Gigabuck: Estimating GNU/Linux's Size, Version 1.06, 30.6.2001, updated 8.11.2001, http://www.dwheeler.com/sloc/redhat71-v1/redhat71sloc.html.

Wheeler, David (2001): Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!, http://www.dwheeler.com/oss_fs_why.html.

Whitlock, Nathalie Walker (2001): The security implications of OSS, March 2001, http://www-106.ibm.com/developerworks/linux/library/l-oss.html?open&l=252,t=gr,p=SecImpOS.

Wiehr, Hartmut; Wild, Martin (2001): Linux-Die Alternative, in: Informationweek, July 7, 2001, http://www.informationweek.de.

Wysong, Thom (2000): Introduction to Open Source and Free Software, http://www.technodemocracy.org/papers/ossfs.html.

Yager, Tom (2000): BSD's strength lies in devilish details. InfoWorld, November 2, 2001, http://www.infoworld.com/articles/tc/xml/01/11/05/011105tcbsd.xml

5.2 Web resources

5.2.1 Publications on OSS and Linux

Daemonnews.org (monthly, BSD)	http://www.daemonnews.org/
Enterprise Linux today	http://eltoday.com/
Linux Community.de	http://www.linux-community.de/
Linux Weekly News	http://lwn.net
Linux World	http://www.linuxworld.com/
Linux.com	http://www.linux.com/
LinuxBusiness.com	http://www.linuxbusiness.com/
LinuxInsider.com	http://www.LinuxInsider.com/
Linuxjournal	http://www2.linuxjournal.com/index.html
Linux-Magazin	http://www.linux-magazin.de/
LinuxNews.com	http://www.linuxnews.com/
Linuxtoday.com	http://linuxtoday.com/

LinuxUser	http://www.linux-user.de/
Newsforge	http://www.newsforge.com/?578
Pro-Linux	http://www.pro-linux.de/
Slashdot	http://slashdot.org/
Computerwoche	http://www.computerwoche.de
ct, iX	http://www.heise.de

5.2.2 Publications on IT in general

ZDNet	http://www.zdnet.com
Infoworld	http://www.infoworld.com
CIO.COM	http://www.cio.com
Darwin	http://www.darwinmag.com/
Computerworld	http://www.computerworld.com
IDG.net	http://idg.net/
InformationWeek	http://www.informationweek.com
IT world	http://www.itworld.com/
WSTA Cyberlibrary	http://cyberlibrary.wsta.org
OS opinion	http://www.osopinion.com/
NetworkWorld	http://www.networkworld.com
IT papers	http://www.itpapers.com
Bitpipe	http://www.bitpipe.com/

5.2.3 Software aggregation

Freshmeat	http://freshmeat.net/
Linux Center	http://www.portalux.com/
Linux.org	http://www.linux.org
LinuxApps.com	http://www.linuxapps.com
LinuxISO	http://www.linuxiso.org/
Tuxfinder	http://tuxfinder.com/

5.2.4 Organisations/OSS community

BerliOS	http://www.berlios.de
Free Software Foundation	http://www.fsf.org
GNU.org/ FSF	http://www.gnu.org
Linux International	http://www.li.org/
Linux Professional Institute	http://www.lpi.org
Linux Verband e.V.	http://www.linux-verband.de/
Linux.org	http://www.linux.org
Open Source Development Lab	http://www.osdlab.org/
Open Source Initiative (OSI)	http://www.opensource.org
OSDN.org	http://www.osdn.com/
Silicon Valley Linux User Group	http://www.svlug.org/
Sourceforge	http://sf.net

5.2.5 Commercial OSS companies

ActiveState	http://www.activestate.com
Linux Global Partners	http://www.linuxglobalpartners.com/
Linux Information Systems AG	http://www.linux-ag.de
LinuxCare	http://www.linuxcare.com
LinuxMall.com	http://www.linuxmall.com/
LinuxTshirts.com	http://www.linuxtshirts.com/
MandrakeSoft	http://www.linux-mandrake.com
MySQL AB	http://www.mysql.com
NuSphere	http://www.nusphere.com/
O´Reilly	http://www.oreillynet.com/
Penguin Computing	http://www.penguincomputing.com
PostgreSQL, Inc.	http://www.pgsql.com/
QliTech	http://www.qlilinux.com
Red Hat	http://www.redhat.com
Sendmail, Inc.	http://www.sendmail.com
SuSE	http://www.suse.de
Trolltech	http://www.trolltech.com
VA Linux	http://www.valinux.com
Zope Corporation (formerly Digital Creations)	http://www.zope.com

5.2.6 OSS projects and products

AbiWord	http://www.abiword.org
Apache	http://www.apache.org
Bind	http://www.isc.org/products/BIND/
CVS	http://www.cvshome.org
Enhydra	http://www.enhydra.org
FreeBSD	http://www.freebsd.org
Gimp	http://www.gimp.org
GNOME	http://www.gnome.org
GNU Project	http://www.gnu.org/software/software.html
Gnu.org	http://www.gnu.org
GPG	http://www.gnupg.org
Interbase	http://www.borland.com/interbase
KDE	http://www.kde.org
Koffice	http://www.koffice.org
Linux	http://www.linuxhq.org
Mailman	http://www.list.org
Mesa	http://www.mesa3d.org
Mozilla	http://www.mozilla.org
MySQL	http://www.mysql.com
OpenOffice	http://www.openoffice.org,
Perl	http://www.perl.com
PHP	http://www.php.net
PostgreSQL	http://www.postgresql.org
Python	http://www.python.org
Qt	http://www.trolltech.com
Samba	http://www.samba.org

Sendmail	http://www.sendmail.org
Staroffice	http://www.sun.com/staroffice
Tcl	http://www.scriptics.com/
Wine	http://www.winehq.org
Zope	http://www.zope.org