Open Source Business Model
– Balancing Customers and Community

Thomas Rosén

Department of Management and Engineering
Linköping University, SE-581 83 Linköping
Abstract

Free and Open Source Software has not only increased researchers’ interest about community-driven software development, but lately, interest from commercial actors increased as well. In addition, some scientists have claimed that Open Source Software has entered a new phase: OSS 2.0. Even so, a coherent way of analyzing commercial Open Source ventures is still missing.

Commercial Open Source firms’ strategies are often described using the term “business models”. However, these models often lack stringent structures and have been used primarily to describe the firms’ offerings and methods to earn revenue.

Through the adaptation of an existing, firmly theoretically-based analytical business model framework, this thesis suggests a new analysis model for studying for-profit Open Source companies. In addition, the framework is generically constructed, ensuring its usability for other industries as well. The model consists of three elements: market positions, operational platform and offering.

This particular study concerned four software product vendors, all of which base their products on Open Source Software. When analyzing their business, insights were made about how these firms operated. The result show that there are certain key elements and factors that determine if a company has a sustainable business or not. From the analysis framework, three elements were refined. The main Open Source Software project connects the market positions and the operational platform; and from the offering, the product and service and the revenue model were very important.

The study identified eight key factors which influenced the elements: brand for the product, the company and the Open Source Software project; community, that is the sum of the non-paying users and developers connected to Open Source Software projects; resources, which are community-based resources such as development and testing; legitimacy, the perceived legitimacy regarding licenses and the revenue models; control, i.e. the control the firm has of the software; ability to charge, or how the company can charge for its services; customers, the paying users; and finally volume, which is the number of paying customers.

The findings also indicate that companies interested in working with the open-source community have to be able to balance the demands from both their customers and the community in order to benefit and gain competitive advantage.
Acknowledgements

This has been one of the greatest challenges in my life, not only mentally, but also physically. During the time spent working with this thesis, I have managed to go from part to full-time PhD candidate. I have had a major surgery and had to recoup from that. Without my family (Hi mom!), this would not have been possible. Therefore, I am glad this ordeal is over and I am ready for the next...

I would like to thank my supervisors, Staffan and Anna, for their patience, perseverance, advice and support that enabled me to get me through this. I would like to thank Daniel for the excellent help with the business model framework and interviews and his overall support. I would like to thank Peter for our crazy trip to the United States for the first round of interviews. I would also like to extend my thanks to all the nice people in my department for all the support I they have given me through these years.

A comforting thought is sent to poor Mica who graciously agreed to correct the language I so brutishly have mangled in this thesis. He may not have realized the magnitude of the task he accepted. I am forever grateful for your work!

I am especially grateful to Peo, the CEO at Cendio. Not only did you let me start this endeavor part-time, but you also supported me when I decided to quit at Cendio in order to become a full-time PhD candidate.

Heartfelt thanks go out to my other “victims” in the case companies: Inge and Daniel at Cendio; Jeremy and Jon at Codeweavers; Mårten and David at MySQL; and Michael at Red Hat. Without your help, this thesis would never have happened.

Finally, I would like to express my gratitude to Lundbergsstiftelsen, which financed this thesis.

Linköping, 2008

Thomas Rosén
Content

1 Introduction ............................................................................................................. 1
  1.1 Purpose ............................................................................................................... 2
  1.2 The Study ......................................................................................................... 2
  1.3 Outline of the Thesis ........................................................................................ 2
2 Free and Open Source Software ............................................................................. 5
  2.1 Hardware and software .................................................................................... 6
  2.2 Licenses ............................................................................................................ 7
  2.3 Hackers and crackers ...................................................................................... 8
2.2 Free Software or Open Source Software .......................................................... 8
  2.2.1 Free Software, Free Society ........................................................................ 9
  2.2.2 The Cathedral and the Bazaar .................................................................... 11
  2.2.3 Beyond software: OSS in political science ............................................... 12
  2.2.4 User innovation .......................................................................................... 14
  2.2.5 The motivations of the developers ............................................................. 15
  2.2.6 Economics and business of OSS ............................................................... 16
  2.2.6.1 Free or open source software? ............................................................... 17
  2.2.6.2 FLOSS study – Use of OSS ................................................................. 20
  2.2.6.3 FLOSS study – Firms OSS activities ..................................................... 21
  2.2.6.4 FLOSS study – OSS Markets and Business Models .......................... 21
  2.2.6.5 Gaming theory and other methods ....................................................... 22
  2.2.6.6 Balancing closed and open source ....................................................... 22
  2.2.6.7 OSS 2.0? ............................................................................................. 23
  2.2.6.8 OSS licensing ....................................................................................... 23
  2.2.6.9 Licensing schemes ............................................................................... 25
  2.2.6.10 Enforcing the copyleft ...................................................................... 28
2.2.6 Licensing schemes ....................................................................................... 26
  2.2.6.10 Enforcing the copyleft ...................................................................... 28
3 Strategy, management and business models ....................................................... 29
  3.1 Long-term strategy ........................................................................................ 29
  3.1.1 Structure and Strategy ............................................................................... 29
  3.1.2 Planning ...................................................................................................... 30
  3.1.3 Positioning ................................................................................................. 30
  3.1.4 Resource-based view (RBV) ....................................................................... 31
  3.1.5 Dynamic capabilities .................................................................................. 32
3.2 Short-term strategies and typologies ................................................................. 33
  3.2.1 Domains and uncertainty .......................................................................... 33
  3.2.2 Configurations ......................................................................................... 34
  3.2.3 The Adaptive Cycle .................................................................................. 35
3.3 Vision and culture ............................................................................................. 36
  3.3.1 Creative management ............................................................................... 36
  3.3.2 Vertical Market Systems .......................................................................... 37
  3.3.3 Reframing business ................................................................................. 38
3.4 The business model .......................................................................................... 39
  3.4.1 Definitions ................................................................................................. 39
  3.4.2 Value Creation ......................................................................................... 39
  3.4.3 Business model as mediator between technology and business .............. 40
  3.4.4 Business models for IT ............................................................................ 41
  3.4.5 Strategic management approach to business models ............................... 42
  3.4.6 Analytical tool ........................................................................................ 43
  3.4.7 Refined model and tool ....................................................................... 44
5.3.1 Key marketing position factors ........................................................................ 74
  5.3.1.1 Business idea, vision and goals ................................................................. 74
  5.3.1.2 Competitors ........................................................................................... 74
  5.3.1.3 Competitive strategies ............................................................................ 75
  5.3.1.4 Partners .................................................................................................. 76
  5.3.1.5 Owners ................................................................................................... 76
  5.3.1.6 Community and customers ................................................................. 76
  5.3.1.7 Brand .................................................................................................... 76
  5.3.1.8 Promotion ............................................................................................. 77
  5.3.1.9 Culture .................................................................................................. 78
5.3.2 Key operating platform factors ........................................................................ 78
  5.3.2.1 History and paths ................................................................................... 78
  5.3.2.2 Processes ............................................................................................... 79
  5.3.2.3 Experience and competencies ............................................................... 79
  5.3.2.4 Intellectual property ............................................................................. 80
  5.3.2.5 Agreements ........................................................................................... 80
  5.3.2.6 Economy ............................................................................................... 80
5.3.3 Key offering factors ......................................................................................... 80
  5.3.3.1 Offer model ........................................................................................... 80
  5.3.3.2 Product/service mix .............................................................................. 81
  5.3.3.3 Revenue models .................................................................................... 81
  5.3.3.4 Distribution model ................................................................................ 81
    5.3.3.4.1 Linux ............................................................................................. 81
    5.3.3.4.2 Mac OS .......................................................................................... 81
5.4 MySQL .............................................................................................................. 82
  5.4.1 Key marketing position factors ....................................................................... 82
    5.4.1.1 Business idea, visions and goals .......................................................... 82
    5.4.1.2 Competitors ....................................................................................... 83
    5.4.1.3 Competitive strategies ....................................................................... 83
    5.4.1.4 Partners ............................................................................................. 84
    5.4.1.5 Owners ............................................................................................... 84
    5.4.1.6 Community and customers ............................................................... 84
    5.4.1.7 Brand ................................................................................................. 84
    5.4.1.8 Promotion .......................................................................................... 84
    5.4.1.9 Culture ............................................................................................... 86
  5.4.2 Key operating platform factors ......................................................................... 86
    5.4.2.1 History and paths ................................................................................. 86
    5.4.2.2 Processes .............................................................................................. 86
    5.4.2.3 Experience and competencies ............................................................. 87
    5.4.2.4 Intellectual property ......................................................................... 87
    5.4.2.5 Agreements ......................................................................................... 87
    5.4.2.6 Economy ............................................................................................. 87
  5.4.3 Key offering factors ....................................................................................... 87
    5.4.3.1 Price model ......................................................................................... 87
    5.4.3.2 Product/service mix ........................................................................... 88
    5.4.3.3 Revenue models ............................................................................... 88
    5.4.3.4 Distribution model ............................................................................. 88
5.5 Red Hat .............................................................................................................. 88
  5.5.1 Key marketing position factors ......................................................................... 89
    5.5.1.1 Business idea, vision and goals ............................................................ 89
    5.5.1.2 Competitors ...................................................................................... 90
    5.5.1.3 Competitive strategies ....................................................................... 91
    5.5.1.4 Partners ............................................................................................ 91
    5.5.1.5 Owners ............................................................................................. 92
    5.5.1.6 Community and customers ............................................................... 92
    5.5.1.7 Brand ................................................................................................. 92
    5.5.1.8 Promotion .......................................................................................... 93
    5.5.1.9 Culture ............................................................................................... 94
  5.5.2 Key operating platform factors ........................................................................ 94
5.5.2.1 History and paths ................................................................. 94
5.5.2.2 Experience and competencies ............................................... 95
5.5.2.3 Processes ........................................................................... 96
5.5.2.4 Intellectual property ............................................................ 96
5.5.2.5 Agreements ....................................................................... 96
5.5.2.6 Economy .......................................................................... 96
5.5.3 Key offering factors ............................................................... 96
5.5.3.1 Price model ................................................................. 96
5.5.3.2 Product/service mix ..................................................... 97
5.5.3.3 Revenue models ........................................................... 97
5.5.3.4 Distribution model ....................................................... 97

6 Cross-case analysis ................................................................. 99

6.1 Features ................................................................................ 99
6.1.1 Summary of the case companies and their offerings ............. 99

6.2 Business model analysis ....................................................... 100

6.2.1 Market positions ................................................................ 100
6.2.1.1 Business ideas, visions and goals ........................................ 100
6.2.1.2 Competitors ................................................................. 101
6.2.1.3 Competitive strategies .................................................. 101
6.2.1.4 Partners ........................................................................ 102
6.2.1.5 Owners ........................................................................ 102
6.2.1.6 Branding ........................................................................ 103
6.2.1.7 Corporate branding benefitting from fame ......................... 104
6.2.1.8 Positioning without fame ................................................ 105
6.2.1.9 OSS effects for marketing positioning ................................. 106
6.2.1.10 Promotion ................................................................. 107
6.2.1.11 Cultural findings ........................................................ 107

6.2.2 Operational platforms ....................................................... 108
6.2.2.1 History and paths ........................................................ 110
6.2.2.2 Experience and competencies .......................................... 111
6.2.2.3 Processes ................................................................. 112
6.2.2.4 Intellectual property rights (IPR) ...................................... 112
6.2.2.5 Economy ................................................................. 113

6.2.3 The offerings ................................................................. 113
6.2.3.1 Product and service mix ............................................. 114
6.2.3.1.1 Packaging service .................................................. 114
6.2.3.1.2 Development service ............................................. 114
6.2.3.1.3 Consultancy service ............................................... 114
6.2.3.1.4 Training ............................................................... 114
6.2.3.2 Revenue models ........................................................ 115
6.2.3.2.1 Subscriptions ....................................................... 115
6.2.3.2.2 Fees ..................................................................... 115
6.2.3.2.3 Proprietary licenses ............................................... 116
6.2.3.2.4 Value-added hardware .............................................. 116
6.2.3.2.5 Value-added software ............................................. 117
6.2.3.2.6 Hybrid licensing .................................................... 117
6.2.3.3 Price model findings .................................................. 119
6.2.3.4 Distribution models .................................................... 119

7 Conclusions ............................................................................. 121

7.1 Theoretical implications ............................................................ 121
7.1.1 The extended business model .............................................. 121
7.1.2 “Business models” for OSS companies ................................. 122

7.2 Analytical results ................................................................. 124
7.2.1 Key factors for OSS software product companies ................ 125
7.2.1.1 The most important factors .......................................... 128
7.2.2 Closing discussion .......................................................... 128
7.2.3 Future challenges ............................................................ 129

7.3 Related research ...................................................................... 129
7.4 Recommendations for further research .................................. 129
8 References .................................................................................................................. 131
  Literature................................................................................................................... 131
  Miscellaneous........................................................................................................... 136
  Interviews.................................................................................................................. 141
Appendix 1 – Swedish interview guide (1)................................................................. 143
Appendix 2 – English interview guide (2005)......................................................... 146
Appendix 3 – Interview guide 2007 ............................................................................ 147

1 Introduction

While some consider Free and Open Source software as something that is gratis, business is still being made from it.\(^1\) Many firms worldwide make their living selling services to customers who use the supposedly free software; some of them are huge multinational corporations like International Business Machines (IBM), Hewlett-Packard (HP) and Sun Microsystems.

Nevertheless, some companies have gone even further. Not only do they offer services based on OSS software; they also create software products and offer them to customers in the same way as more traditional and proprietary software vendors do. This could potentially mean clashes between different philosophies, such as sharing versus for-profit and best effort versus “good enough”. In the end, this means that companies that would want to pursue this sort of business have to balance the often conflicting demands from the market and community.\(^2\) This thesis investigates such OSS software producing companies.

The research regarding OSS, from an economic and management point of view, has been focused on three main veins: innovation through the participation of users in the development of software (cf von Hippel and von Krogh, 2003; Klinecweicz, 2005); the motivation of developers to create the software (cf Lerner and Tirole, 2002; Ghosh et al. 2002a; Bonaccorsi and Rossi, 2003); and the economics and motivations for companies to make use of it (cf Lerner and Tirole, 2002; Mustonen, 2003). Aside from that, researchers have studied the licensing (cf Väälimäki, 2003, Comino and Manenti, 2007) and in some cases, the business models for (cf Raymond, 2001; Wichmann and Stiller, 2002) firms exploiting free and open source software.

However, the research regarding business models for OSS uses the term “business model” to connote the offerings and revenues, with an emphasis on service-based revenue models as opposed to other, more proprietary models. Most of the earlier-studied models are heavily indebted to Raymond (2001). In this thesis, this view of a firm’s business model is considered too narrow; other dimensions should influence the workings of a company as well.

In fact, there exist much wider definitions of a business model (cf Timmers, 

---

\(^1\) This thesis will use the abbreviation OSS to signify Open Source Software. Other examples of terms are F/OSS (Free/Open Source Software) and FLOSS (Free and Libre Open Source Software). The former term, aside from sending dental health signals, has been used in European research, since Libre Software is commonly used in French and Spanish speaking countries as a translation from Free Software. As shown below, Richard M. Stallman, the creator of Free Software, connotes the term as a moral or ideological standpoint that has normative implications on how the software is used. Since the use of the software in some of our cases has more of a pragmatic view, which, if not opposed then disagrees with those of Free Software, this term is not used.

\(^2\) When the term “community” is mentioned, it connotes the OSS community unless stated other wise.
1998; Amit and Zott, 2001; Chesbrough and Rosenbloom, 2002; Hedman and Kalling, 2002; Afuah, 2003; Kindström, 2003 and 2005). However, these frameworks range from the too complex, for example Hedman and Kalling (2002), to the excessively generic, thus lacking crucial elements that would make them viable tools for analyzing OSS companies, such as Kindström (2003). Therefore, the goal here was to create a solid, simple but effective business model framework, and then utilize this framework as a tool for the research analysis.

Finally, an interesting issue would be to distinguish if there are key factors that indicate successful OSS business, and if the business model framework can be an analysis tool for that purpose. By using the new business model framework, and testing it on a number of case companies, this research will attempt to answer that question.

1.1 Purpose

The purpose of this thesis is to develop a framework for describing and analyzing software product OSS companies from a business model perspective. More specifically, this thesis aims to:

- develop a business model framework;
- analyze the revenue models of the case companies and compare them to established research; and
- determine the key factors that are important for a sustainable business strategy for software product OSS companies.

1.2 The Study

This is a multiple case study of four companies, two American and two Swedish. Each country is represented by a larger company: Red Hat Inc. in the USA and MySQL AB in Sweden, and a smaller one, Codeweavers Inc. in the USA and Cendio AB in Sweden. All four companies develop and package software products based on open source software.

Each of the cases has been analyzed through the extended business model framework that was developed during the study.

1.3 Outline of the Thesis

The thesis begins with a theoretical framework consisting of two parts. The first is a discussion of Open Source Software, the second, a description of the strategic and managerial research for the business model framework, which ends with the new adapted business model framework. The next part of the thesis consists of an explanation and discussion of the background and methodologies used in the research.

Following this, the analysis begins with a structural presentation of the cases, followed by a cross-case analysis. The final analytical portion of the thesis presents the theoretical implications of the new business model
framework and the resultant findings.

Finally, the research's findings are presented and discussed, findings with the potential to aid the companies involved and others like them, and suggestions for further research are made.
2 Free and Open Source Software

This chapter will present the concept of Open Source Software (OSS), and is part of the theoretical framework for this thesis. However, in order to best present the concept, the chapter first takes a step back to look at the computers, hardware, software and other things associated with it.

In order to explain how OSS works in the computer industry, a brief and basic explanation of the technology and terminology in the industry will be provided.

The computer industry is widely known as one of the fastest-moving industries in the world, rivaled perhaps only by the emerging biotechnology industry (Wichmann, 2002a; Hedman and Kalling, 2002). The changes are rapid and often profound. This is because in reality, the computer industry consists of not one, but two fast moving sub-industries: those dealing with hardware and software. Even if they are considered conjoined, each of these sub-industries has had its own history of evolution and revolution, each which eventually affects the other (Levy, 2002).

The changes in the hardware industry have been somewhat “regulated” by the now famous Moore’s Law from 1965 that states, “The number of transistors on a chip doubles about every two years” (Intel, 2007). Instead of revolutions, the progress can be seen as an evolution, and this ongoing process is projected to last for many years (Kurzweil, 2001).

For software, the progress has taken more of a revolutionary path, where new technology, processes or even paradigm shifts have altered the industry in a fundamental way. The software industry has lived through events like “the software crisis” in the 1970s, when companies and universities realized that the complexity of software had to be addressed. This created new research fields and “software engineers”, and turned software into something that could carry value. Another major event was the open source software revolution at the end of the last millennium, in which major multinational corporations like IBM and HP openly embraced and even praised the power of software created through voluntary actions (Randell, 1996).

One of the key issues is how to develop software, and there are different paradigms for this. Microsoft and other commercial players use the concept of “good enough” software, and release their products with “known bugs”; there are known errors in the code, but it should be “good enough” to ship (Bach, 1996).

However, the philosophy of free or open source software differs. In general, deadlines are not considered important and the software is released (as stable) when most of the (known) errors are corrected. The way that this is done is also discussed between the main “thinkers” behind the movement, Richard M. Stallman and Eric Raymond (Stallman, 2001; Raymond, 2001).

Despite the fact that it has its roots in the dawn of computing (Raymond,
most people seem to consider Open Source Software (OSS) as something relatively new. But prior to that and even today, most people using the Internet are not aware that a major part of their traffic to web pages and through name servers is made possible thanks to OSS (Netcraft, 2007).

Another important philosophy considers the rights of the producer of software and those of the users of that software. The commercial producers protect their rights through their licenses, only giving the users the mere right to use the software under certain conditions and for a fee. This type of software license will be referred as to a “proprietary license” henceforth.

The free and open source software producers often do the opposite, giving away the rights of the software to the users. Interestingly enough, both of these practices use the same basic law, the copyright law. Where the proprietary licenses make full use of the law, terminating most of the rights of the user, the OSS license known as the General Public License (GPL) does the opposite thing, enforcing the rights of the user; therefore it is often referred to as “copyleft” (Stallman, 2001).

2.1.1 Hardware and software

Most modern electronic devices could be called “computers”; regardless if they are washing machines, refrigerators, mainframes, servers, PCs, laptops or mobile phones, they all have both hardware and software.

The hardware is the physical machine consisting of electronic components like the processor, memory, screen etc. In order to do anything, and to enable us to control the device, the hardware needs software to run.

The software contains instructions for what the different electronic devices will do in a certain situation. In order to describe the difference, Figure 1 below shows the “computer stack”:

![Figure 1: Simplified computer "stack".](image)

Software can be compared to the recipe of a dish. Software has two different "states", the source code and the binary code. The source code is readable for humans and consists of text and numbers, while the binary code only consists of binary data, which is executable and interpreted by the hardware. To make the analogy of the recipe, the source code is the recipe, which we read
or create in order to make a dish. It is made in a certain language, just as
recipes can be made in English, Swedish or French.
Examples of the languages used are Java, C, C++, XML etc. In order to
make the hardware understand, the source code needs to be converted into
binary code. Some languages use certain software that interprets the source
code, for example Java, while others like C and C++ use special software
called **compilers** that generate the binary code. Compiling could be seen as
making the dish, i.e. blending the ingredients and cooking them, which
when done is served.

### 2.1.2 Licenses

Almost all software comes with a *license*. This is because all software is pro-
tected by copyright legislation. This means that a user normally has no
rights to use the software. A license usually grants the user certain rights,
but mostly they also contain a *disclaimer*, making the copyright owner free of
damages incurred by the software.

There are many different types of licenses. At one end of the spectrum are
the *commercial* licenses, often called **End User License Agreements (EULA)**.
These are normally a specific license per software or software company; Mi-
crosoft, for example, has a large number of licenses. At the other end of the
spectrum are more generalized licenses like **freeware**, **shareware**, **public do-
main**, **free software** (FS) and **Open-source Software** (OSS). To make the mat-
ters more confusing, some of the public domain licenses are considered as
OSS.

The terms of these licenses ranges from more or less free (public domain),
which give the user the right to even change the license to a commercial one,
to very limited terms (commercial), which even charge the user a fee for the
rights entitled. The way the software is distributed also differs; some comes
only as executable code (binary code), while others are distributed with both
the source code and the binary code. Table 1 shows the differences:

<table>
<thead>
<tr>
<th><strong>Table 1: Different types of software licenses.</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of license</strong></td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>Public domain</td>
</tr>
</tbody>
</table>
| Open source Soft-
| ware               | $\times^5$                       | x                       | x                       | $\times^5$                    |
| Free Software       | -                                | x                       | x                       | -                             |

3 In some countries such as the USA, some software can also be patented. In general, how-
ever, software is copyrighted.

4 There are of course other kinds of license terms, like adware, crippleware, vaporware etc.
The ones listed are the most used.

5 Some of the licenses allow licensing fees.
The term OSS refers to how a piece of software is licensed in a certain way to ensure that the user gets extra benefits, the rights to not only use the software, but also to read the source code, change it and redistribute it. By giving the users this right, the idea is that the software evolves at great speed (Open Source Initiative, 2007).

2.1.3 Hackers and crackers

Important to note is that the programmers tied to free and open source software call themselves “hackers” (Levy, 2002, Raymond 2001, Stallman, 2002). This is not to be confused with the term “hackers” used by media to denote criminals who exploit computer systems or try to gain access to systems without proper authorization. The original hackers call them “black hats”, “crackers” or the derogatory term “script kiddies”.

It is important to understand the difference in this matter, because most of the prominent figures of the software history are called “great hackers” (Wikipedia, 2007a).

2.2 Free Software or Open Source Software

As an ironic twist, even if Free Software (FS) already had existed for several years, the academic community did not seem to realize it existed before the creation of the term Open Source Software (OSS), and the Initial Public Offering (IPO) of Red Hat and other OSS companies. The first reaction from academics was a lack of understanding, mainly of the driving forces for the developers but later concerning why and how firms could work with OSS.

FS and OSS are different terms describing the same movement, but from different standpoints. In Europe, a third term, Software Libre (SL), has been created, principally out of language needs. From the scientific point of view, the research has often concerned all three terms despite the differences between them, which in turn has created abbreviations like FOSS, F/OSS and FLOSS. In this thesis, the term OSS will be the most used; in this section, however, the same terminology as used by the particular authors is used.

What in essence is all this about? It is about software used on computers. This particular kind of software is developed by a large number of developers, and it is free for anyone to use, develop and distribute. All these rights are protected through a license. The license differs from a “proprietary” license, for example for Microsoft Office, which only allows the user to use the

<table>
<thead>
<tr>
<th>Freeware</th>
<th>-</th>
<th>-</th>
<th>x</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shareware</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>x⁶</td>
</tr>
<tr>
<td>Commercial licenses</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

⁶ Shareware usually charges the user for a full version of the software, or after a period of time.
software under certain conditions. The concept of free or shared software has been around since the early days of the computer, and has been transformed by companies that have created business opportunities using it. Since the millennium shift, the interest in this kind of software and the business it brings has been increasing in the academic world.

The concept as a whole, no matter what part of the movement you mean, has been described in books and articles, and usually books are the best way to start to understand how an idea works. The two most prominent authors, both of whom are the founders of the two parts of the community, are Richard Stallman and Eric Raymond. It is therefore fitting to let them start to describe the driving forces and economics behind the movement; the difference between FS and OSS will be described by their creators. In fact, the books of Stallman (2002) and Raymond (2001) have been the natural starting point for most researchers investigating free and open source software.7

2.2.1 Free Software, Free Society

It is safe to call Richard M. Stallman “the founder of free software” (but also, incidentally, open source software). He came to the Artificial Intelligence Lab at MIT in 1971 and embraced the software-sharing culture. In his book *Free Software, Free Society* (2002), he describes, in the form of essays, his time at MIT, how his GNU (GNU’s Not Unix) project was started, and his view on information, FS and OSS. In 1984, he left MIT and started to work on GNU software, wrote the GNU Manifesto, and in 1985 created the Free Software Foundation. The most-used license for free software (or OSS) is GNU General Public License (GPL), which Stallman first released in 1989.8

2.2.1.1 Free or open source software?

The collection of essays was collected after Eric Raymond proposed the term Open Source Software, so many of the essays contain criticism and point out differences between FS and OSS. The following quote describes his view perfectly:

For the Open Source movement, non-free software is a suboptimal solution. For the Free Software movement, non-free software is a social problem and free software is the solution (Stallman, 2002, p. 55).

Free software is described as being the moral and ideological point of view regarding the creation and dispersion of the source code. In a way, the two movements can be seen as political opponents, just as the in American democracy with its republicans and democrats. Stallman’s ideological stance is based on the “Free as in Freedom” (opposed to free as gratis) in which he de-

---

7 Both books contain shorter essays that have also been published on the Internet. Raymond’s essays have also been publicly revisited continuously. The content presented here is mainly based on the published books.

8 The second release of GPL was released in 1991 and the forthcoming third version is currently being formalized (2006).
scribes the current system of copyright and patents as a way of trapping and even limiting the creativeness of software developers (or even every person). Companies that are limiting individual user’s rights in these matters are considered unethical, and even anti-social, by Stallman.

His message of freedom has been mistaken to be that no business can be made from software, a notion he is distinctly opposed to. He describes how he survived, after leaving MIT, on fees collected from copying tapes of his tools, and how the Free Software Foundation was formed as a tax-exempt charity which mainly received its income from copies (on CD-ROM), books, t-shirts, etc. He also tells the story on how he made money from being a consultant and the story regarding Cygnus Solutions, the first company solely working with FS, and their business concept.

2.2.1.2 Software patents

Stallman is also opposed to software patents and the current use of copyrights, and goes to great lengths to describe how they have are being misused now and have been in the past. To illustrate this, Stallman also shows us how he used the copyright laws to create the copyleft licenses that are the foundation of FS. He describes the reason for creating them like this:

My work on free software is motivated by an idealistic goal: spreading freedom and co-operation. I want to encourage free software to spread, replacing proprietary software that forbids cooperation, and thus make our society better. That’s the basic reason why the GNU General Public License is written as it is – as a copyleft (Stallman, 2002, p. 91)

Stallman has set his sights high, since he really is concerned about the current society and the use of computers and software. In an essay written in 2002 he gives a number of examples on how the freedom of using computers is being undercut by large corporations today (Stallman, 2002, p. 115-117). Stallman then argues why software should be free and how this would affect our society.

2.2.1.3 Understanding Stallman’s point of view

The key of understanding Stallman is to see him as a programmer first and a philosopher second. His goal is to enable other programmers to adapt the software after their personal need, and at the same time, protect them from companies stealing their code and making money from it. He is not a businessperson – his description on how to do business is close to impossible for a company larger than just a small number of employees. Moreover, selling CDs when most people can download the same software for free will be less profitable given time. Succeeding with a company like Cygnus Solutions, with a business model of selling general support for FS or OSS when such software numbers tens of thousands in the free domain, is probably close to impossible to achieve.
2.2.2 The Cathedral and the Bazaar
Stallman is the founder and the foremost philosopher of the FS movement. Nevertheless, his foundation is ideological with moral first and practical second. Eric Raymond, who was also a part of the movement, saw other important issues, ones based on a more practical sense (Raymond, 2001).

What Raymond saw was the new development process that Linus Torvalds introduced for the Linux Kernel project:

Linus Torvalds’s style of development - release early and often, delegate everything you can, be open to the point of promiscuity - came as a surprise. No quiet, reverent cathedral-building here - rather, the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches (aptly symbolized by the Linux archive sites, who’d take submissions from anyone) out of which a coherent and stable system could seemingly emerge only by a succession of miracles. The fact that this bazaar style seemed to work, and work well, came as a distinct shock (Raymond, 2001, p. 21)

Raymond also envisioned a new way of developing software – where the input from a multitude of both users and developers was a powerful and effective way of creating high quality open programs. He also coined a new “law” for software development, affectionately dubbed “Linus’ Law”: “Given enough eyeballs, all bugs are shallow” (Raymond, 2001 p. 30).

2.2.2.1 Commercializing “free software”
In 1998, Netscape decided to open up the source code for their web browser, and Raymond (and others) felt that the label “free software” was troublesome to use when talking to commercial actors:

Specifically, we have a problem with the term “free software”, itself, not the concept. I’ve become convinced that the term has to go. The problem with it is twofold. First, it’s confusing; the term ‘free’ is very ambiguous (something the Free Software Foundation’s propaganda has to wrestle with constantly). Does “free” mean “no money charged?” or does it mean ‘free to be modified by anyone’, or something else? Second, the term makes a lot of corporate types nervous. While this does not intrinsically bother me in the least, we now have a pragmatic interest in converting these people rather than thumbing our noses at them. There’s now a chance we can make serious gains in the mainstream business world without compromising our ideals and commitment to technical excellence - so it’s time to reposition. We need a new and better label (Raymond, 1998).

Many prominent community leaders at the time lauded the new suggested label, “open source”, even if founder Richard Stallman was against it (Raymond, 1998).

2.2.2.2 Business models and OSS
In order to promote the commercial use of OSS, Raymond also presented a number of different business models, those that he thought might apply to companies using OSS. At the time when he wrote it, GPL was the mostly prominent license used, so he foresaw troubles using a direct sale-value model and suggested a number of alternative business models (2001). Table 2 below shows and summarizes them:
Table 2: Open Source Business Models by Raymond (2001).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use-Value Funding Models</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost-Sharing</td>
<td>Pool programming resources into shared project like Apache web server.</td>
<td>While paying your programmer(s), you will get (the) others free.</td>
</tr>
<tr>
<td>Risk-Spreading</td>
<td>If you have created utility software that may not be maintained within your</td>
<td>If/when your programmers quit, the software will still be alive and</td>
</tr>
<tr>
<td></td>
<td>organization, make it free and let others use it and maintain it. Example:</td>
<td>maintained, without costs for you.</td>
</tr>
<tr>
<td></td>
<td>CiscoPrint.</td>
<td></td>
</tr>
<tr>
<td><strong>Indirect Sale-Value Models</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loss-Leader/ Market Positioner</td>
<td>Use of OSS to create or maintain a market position for proprietary software.</td>
<td>Marketing and PR, lower cost for creating software. Faster evolution of</td>
</tr>
<tr>
<td></td>
<td>Example: Netscape Communications.</td>
<td>the software. Drives server software sales.</td>
</tr>
<tr>
<td>Widget Frosting</td>
<td>Hardware manufacturers make their drivers, configuration and utility tools</td>
<td>Marketing and PR, lower cost for creating software. Faster evolution of</td>
</tr>
<tr>
<td></td>
<td>free. Example: Apple Computers.</td>
<td>the software. Drives hardware sales.</td>
</tr>
<tr>
<td>Give Away the Recipe, Open a Restaurant</td>
<td>Packaging and distribution of software and creates a market position for</td>
<td>Marketing and PR. Drives sales of service contracts and subscription</td>
</tr>
<tr>
<td></td>
<td>services. Examples: Cygnus Solutions, Red Hat, Digital Creations.</td>
<td>services.</td>
</tr>
<tr>
<td>Accessorizing</td>
<td>Sell accessorizes. Low end: t-shirts, mugs etc. High end: manuals and</td>
<td>Riding the hype wave.</td>
</tr>
<tr>
<td></td>
<td>other documentation. Example: O'Reilly and Associates.</td>
<td></td>
</tr>
<tr>
<td>Free the Future, Sell the Present</td>
<td>The proprietary software will be opened under certain conditions. However,</td>
<td>The software will have an extended lifecycle. Customers will benefit and</td>
</tr>
<tr>
<td></td>
<td>the peer-review will be inhibiting in the early stages, when it is needed</td>
<td>might be more positive about the product.</td>
</tr>
<tr>
<td></td>
<td>the most. Example: Aladdin Enterprises.</td>
<td></td>
</tr>
<tr>
<td>Free the Software, Sell the Brand</td>
<td>Open a software product; retain a test suite and/or brand name and add</td>
<td>Marketing and PR, lower cost for creating software. Faster evolution of</td>
</tr>
<tr>
<td></td>
<td>services or third part software. Example: Sun Microsystems (StarOffice/</td>
<td>the software. Drives software and hardware sales.</td>
</tr>
<tr>
<td></td>
<td>OpenOffice).</td>
<td></td>
</tr>
<tr>
<td>Free the Software, Sell the Content</td>
<td>Brokerage. The client and server software are open, but the content is</td>
<td>Marketing and PR, lower cost for creating software. Drives content sales.</td>
</tr>
<tr>
<td></td>
<td>proprietary. No example, but stock-ticker or Internet access mentioned.</td>
<td></td>
</tr>
</tbody>
</table>

These models will be revisited later in the thesis, during the presentation of the framework for business models developed for this research.

The following sections present the material that has been essential for this work. However, there may be another dimension to OSS. Is it possible to take some of the key components of OSS development and make use of them in other areas? The next author definitely believes so.

2.2.3 Beyond software: OSS in political science

OSS has had some impact in other scientific areas as well. The almost revolutionary concept of working for a common good, rather than solely for personal enrichment, may of course affect other parts of our society. In “The Success of Open Source” (2004), Weber focused on the way the OSS community reframed some of the basic problems of governance. He examines property, and how the production and distribution of goods, mostly digitalized goods (such as software, music, movies and other “property”), has been affected by the Internet and OSS. He states:
Open source is an experiment in building a political economy – that is, a system of sustainable value creation and a set of governance mechanisms. In this case it is a governance system that holds together a community of producers around this counterintuitive notion of property rights as distribution, it is also a political economy that taps into the broad range of human motivations and relies on a creative and evolving set of organizational structures to coordinate behavior (Weber, 2004 p. 1).

Some of the questions he asked are in line with what business management researchers has studied, but he is still focused on political and organizational issues, rather than the business aspects. Some of his discussion regarding generic business models for OSS will be examined in the following chapters.

Weber makes a reasonable explanation for the success of OSS by looking into micro foundations and macro organization. In the first, he explores the myths and facts regarding the individual motivations of the developers and the economic logic of the collective good; in the latter, coordination of the contributions and the complexity of the technology, involved (software), and how it is governed (Weber, 2004).

In conclusion, Weber presents two generic hypotheses for when the open source process is more likely to be successive: one concerning the task, the second the motivations of the people involved. For tasks to be successful using the open source process, these characteristics must be present:

- Contributions must be allowed to be derived from sources that are not proprietary, closed or discriminating.
- The product is considered important for a critical mass of users.
- The product is subjected to peer-review and continuous error correction.
- There is a positive network effect when using the product.
- Anyone, individually or in a small group, can lead the project and generate a substantive, evolving and useful core.
- A community exists that interacts and develops the process surrounding the product development (Weber, 2004 p. 271).

In order to make this work, the people involved in the process need the following characteristics:

- Potential contributors can easily judge the quality and viability of the product.
- Everyone involved has all the information needed to make informed guesses about the future quality and viability of the product.
- The contributors have other motives than economical gain, and may settle for other types of future rewards.
- Contributors gain new knowledge through “learning by doing”, which may be valuable later in the process.
- The contributors are inclined to be positive, normative and ethical about the process (Weber, 2004 p. 272).
The theme of Weber’s book lies at the outskirts of this research’s focus, but there are some aspects that still affect the commercialization of OSS. In addition, Weber’s work is a great introduction to OSS itself, and opens the mind for the larger implications the OSS concept may attain in the political world itself.

However, the following sections describe the research areas that have some impact on this work. In this thesis, an attempt has been made to present them in such a way that they reflect on the impact or importance they have had for the research; this means that some articles are given space, while others are presented in short.

Even if OSS was primarily a movement within software development, give the creative way of using (American) copyright law and the new development processes, it has made its imprint on sciences other than computer science. As stated above, the research has primarily been conducted in three veins: user innovation, developer motivation and the business economics of OSS. In this thesis, the focus has naturally been on the articles concerning business management and economics, but issues regarding law and even political science are also addressed since they affect the business model concept.

The concept of giving intellectual property (which software is considered) away, and the implications of this on the software business, has been studied by a multitude of researchers. When examining articles, dissertations, papers and books, the business-related research has focused on the areas presented above. In addition, in order to investigate the business logic of OSS in full, articles regarding the legal incitements behind OSS licenses and some political science have also been studied.

Even if user innovation has not been of primary interest for this research, it is believed to be one of the key elements explaining why some companies have entered OSS. For this reason, some of the most important articles and authors are presented below.

2.2.4 User innovation

One of the most potent research streams regarding OSS was the hypothesis regarding user innovation. Researchers had long realized that extensive in-house research and development was a massive barrier of entry to many markets, and very few corporations had the resources to be able to compete, but suddenly upstarts who got their ideas to the market in a different way beset these giants. Chesbrough (2003) calls this “open innovation”.

One natural context for open innovation is of course OSS. The study of the innovativeness of the community sprouted a number of articles. One of the hypotheses, the private-collective innovation model, where the freely shared information was used for the common good was presented as opposed to invention created from private investment (von Hippel and von Krogh, 2003). Similarly, the researchers studied how users joined and specialized themselves in community projects (von Krogh et al., 2003). Some also investigated
how users profited by freely revealing their innovations (Harhoff et al., 2003), while others looked into how old and new ventures benefited from revealing their source code and thus gaining advantages and benefits from open innovation (Henkel, 2003; Gruber and Henkel, 2004).

However, not everyone shares the all-shiny view. For instance, a Klincewicz (2005) study shows that the number of innovative OSS projects is rare; the majority of projects are so-called “me too” projects which mimic others. OSS commoditizes software very quickly, and expensive research projects may suffer from this. However, there are indications that the commoditization pressure from OSS would stimulate innovation in proprietary software rather than OSS.

Even if this type of research has been very active, it has not been interesting for this work. Whether OSS is innovative or not does not matter in the end as long as the OSS concept persists. Since this matter rarely matters for the individual developer or user, it may be a purely academic matter. Nevertheless, what matters for the developers, or at least should matter, is the reason(s) for them to enter into OSS and develop software. This issue has been one of the major questions for the economic researchers.

It may look like the motivations of the developers have little to do with the commercialization of OSS, but nothing could be closer to the truth. In reality, the developers and users make up OSS, so companies interested in utilizing OSS should be very interested in how they can motivate developers to work on their projects.

2.2.5 The motivations of the developers
In their article, Lerner and Tirole (2002) studied what motivates programmers to contribute to OSS projects. Aside from those who are paid to work in OSS projects, the developers who spend their free time programming with OSS are motivated by both their status among other developers and the prospects of getting a well-paid job in the future. This concept inspired other researchers into looking beyond Raymond’s prime motivator: “an itch to scratch” (2001).

2.2.5.1 Floss report - developer survey
The perhaps largest and most comprehensive study on OSS is the European FLOSS study. The project, funded by the European Commission, made its final report in 2002, consisting of five parts. The part of the report that received the most attention was part four, which contained the online survey with answers from 2,784 OSS developers.

The findings both confirmed and rescinded the results from earlier surveys. The questionnaire consisted of topics ranging from personal features, work and project experiences, motivations and expectations regarding the OSS community and the questions regarding Free Software (FS) and OSS,  

---

9 There are actually seven reports – but the reports are in four distinctive parts.
proprietary software and monetary or non-monetary rewards.

The first part of the report covers the personal questions of the questionnaire. The findings confirmed some of the preconceptions of an OSS developer, e.g. that they are predominately male, but one of the assumptions regarding OSS developers as singles proved to be false. In fact, the majority of the developers had relationships. In addition, most of the developers were highly educated with some university degree, mostly bachelor or master degrees. One of the more interesting findings was that a majority of the developers (of the sample) were from Europe (Ghosh et al., 2002a).

OSS development was rather a hobby than a profession. Most of the developers did not spend more than 10 hours per week developing OSS. However, developing OSS may not be considered for “leisure”, since a majority of the sample claimed to use OSS when at work or school/university. Another part investigated was the number of projects in which the respondents were working in. The vast majority of the developers were active in up to five projects and, interestingly enough, at least one-third indicated that they have led one project (Ghosh et al., 2002a).

Why do developers develop OSS and continue to do so? A majority of the developers joined OSS in order to learn and develop new skills, and stay for the same reason. Another important aspect is to share knowledge and skills. The report concludes that the initial motivation for joining was aimed at individual skills and exchange of information, and that commercial and political aspects grow more important when the community matures (Ghosh et al., 2002a).

2.2.5.2 Floss study – Source code authors

The final part of the report tried to establish the authorship in the source code. In order to do this, the research team used specialized scanning software with which they scanned OSS source code. The results were not any surprise; most OSS projects were small, with a small number of active developers, and the contribution from the initial author was the most crucial during the whole life cycle of the project (Ghosh et al., 2002b).

Even if the result was somewhat expected, it does have implications for at least one of the cases. It is not as easy as to start an OSS project and then think that developers will flock to it; rather, the opposite is the rule of thumb. This means that in order to gain any interest in a project, many resources must be put into it. This is something that has been discussed in other work (Goldman and Gabriel, 2005).

The result shows that when it comes to many authors, OSS is quite “top-heavy”: a small number of developers are responsible for large parts of the total amount of code. The second result shows that a typical OSS project is often quite small; the vast majority having has one or two developers working in them. Not surprisingly, the rule of thumb says the larger the project is the more developers are working in it. The conclusion is that projects are of-
Goldman and Gabriel (2005) list a number of reasons for why developers volunteer their skills for OSS:

- need for the product ["an itch to scratch" (Raymond, 2001)]
- enjoyment, creativity
- reputation and status in the community
- affiliation, spending time with like-minded
- identity, the self-image
- ideology, values, for example the FS philosophy
- training, learning to improve the skills in order to boost the career
- fairness a feeling of debt for using a software
- making things better
- feedback, connected to the training of one’s skills

What Goldman and Gabriel do not mention is that the individual may be motivated by one or several of the reasons listed above, and the reasons may change during the time in the community, but this was made clear in the FLOSS study. The motivation of the developer is individual, and perhaps even more complex than anticipated.

Reality is often, and regardless of all the good intentions or strategies in the world, not every company succeeds. This is something with which most articles agree. However, there are some “best practices” when it comes to creating an OSS company, which is the subject of the next section.

### 2.2.6 Economics and business of OSS

One statement from Lerner and Tirole (2002) effectively sums up the current view on the economics of OSS: “Yet to an economist, the behavior of individual programmers and commercial companies engaged in open source processes is startling” (Lerner and Tirole, 2002a p. 198).

#### 2.2.6.1 Strategies for OSS companies

In their article, Lerner and Tirole examine four OSS projects in order to understand key economic factors for OSS. In addition, even if their findings, as presented above, include the motivation of the developers as an important factor, they also presented three OSS strategies for commercial companies. Even if their nomenclature differs from earlier works, the patterns recognized in earlier works are still present. The strategies they named are living symbiotically off an open source project, code release and intermediaries.

The first strategy is the “classical” strategy as presented by both Raymond (2001) and Stallman (2002), where a company offers complimentary products and services to existing OSS projects by subsidizing the project through e.g. the addition of paid programmers. As representatives for this strategy, Red Hat and VA Linux are named. The second strategy, code release, regards companies that release proprietary software to OSS in order to promote sales.
in related areas, with Hewlett-Packard and Netscape as representative examples. The third and final strategy, *intermediaries*, is used to describe companies like Collab.net which act as agents for OSS projects and programmers, and gain their revenues for services such as “headhunting” programmers or projects for a company, setting up contracts between parties, etc. (Lerner and Tirole, 2002a).

One of the first companies that decided to embrace OSS (and which was partly responsible for the creation of the term Open Source Software) was Netscape (Raymond, 2001). Hecker (1999) was one of the systems engineers that released the former closed code to OSS. His essay discusses the reasons for a company to “go open source” and provides hands-on tips for companies interested in releasing their source code. He also lists a number of “business models” that are reminiscent of Raymond (2001), but he also introduces what he calls “hybrid business models”. The models he presents are:

- **Support Sellers**: revenue gained from media distribution, branding, training, consulting, custom development and support.
- **Loss Leader**: an OSS product is used to promote a proprietary product.
- **Widget Frosting**: hardware companies provide OSS such as drivers and interfaces.
- **Accessorizing**: revenue comes from selling accessories like manuals, books, t-shirts and other paraphernalia.
- **Service Enabler**: OSS products are created for access to online services.
- **Brand Licensing**: charging other companies for the right to use their brand name in derivative products.
- **Sell It, Free It**: a proprietary software that is continually converted to OSS when it is appropriate
- **Software Franchising**: a company authorizes use of its brand in specific areas of interest (geographical or vertical markets) with support, training and consulting. This is a combination of several other models above.
- **Hybrid 1**: proprietary and OSS “side-by-side” [termed as dual licensing below] differentiating between users and use of software.
- **Hybrid 2**: allowing free use, but restricting or charging for modification of the code and/or distribution of modified software.

Aside from the models, Hecker also provides insight on the process for taking proprietary code to OSS together with implications for the continued product development (Hecker, 1999). Despite the “age” of his essay, many of his tips are still viable, and some of his models are used by companies today.

Behlendorf (1999) is on the same track as Hecker. He argues for why software companies should utilize OSS as part of their business strategies. He also presents a number of examples of how a company can implement OSS
strategies, and even provides hands-on tips for what roles the members of an OSS team would have in the company and their projected workloads, and what OSS licenses to use for different purposes.

Even if Behlendorf’s approach is more on the organizational side of how to set up shop for OSS instead of how the company would actually do business, his own personal experiences make his words something most people should listen to and consider.

Krishnamurthy (2005) presents three basic business models for OSS, the distributor, the software producer and the third-party service provider. However, Krishnamurthy also presents additional “flavors” of the basic models in order to present different ways to combine products and services to gain revenue streams. For the distributor model, he envisions three variants: the retail service (box with CD and manual), enterprise support services and upgrade services.

For the software producer, Krishnamurthy pictures two different approaches: the non-GPL model, where the OSS is embedded in proprietary product; and the GPL model, where the company still uses proprietary licensing, but also provides the source code for their users. The main difference, Krishnamurthy states, “is in terms of what the seller expects from the user”. In the first case, the non-GPL model is a user who is just interested in using the software and nothing else, but in the latter case, the user is empowered and interested to engage in a two-way communication (Krishnamurthy, 2005).

For the final basic model, the third-party service provider, there are two different strategies available: selling software combined with services or only offering services. In the first case, the company can combine software products, some OSS and some proprietary, into a well-integrated software suite and top it off with services. Services is a straight-forward approach to any kind of software, since most proprietary software producers already have services as a secondary revenue stream (Krishnamurthy, 2005).

In addition, looking at Krishnamurthy’s distributor models as seen below, Red Hat dropped the retail business and combined the other two strategies with their new enterprise offering. Their new strategy has turned out to be very successful, and it is being copied by other OSS companies.

Goldman and Gabriel (2005) have tried to take a holistic view on how a company could approach OSS. Their book is written with a perspective from Sun Microsystems, so the advice they give is targeted toward a larger corporation, but may be used by smaller firms. They present seven “classic open source business models”:

- Bundling OSS with other software and support, charging for the bundle.
- Add value with additional modules and sell the modules bundled with OSS.
• Subscription services with tested and assured code.
• Ancillary items such as books, T-shirts, mugs etc.
• Consulting services that leverage OSS.
• Sell hardware that runs OSS well.
• Sell software that uses OSS as a platform.

In addition to these, Goldman and Gabriel present two models for companies that retain the copyright for the source code and expect minimal contributions:

• Release the software as OSS, but offer proprietary licenses for companies that wish to use it in their proprietary product.
• Sell the newest version; release the previous version as OSS.

Aside from the revenue models, Goldman and Gabriel also present reasons for companies to consider OSS as a viable option for their software development. They offer guidelines for companies that want to make use of OSS in order to promote their proprietary software product. The main reasons are improved communication with customers (users), marketing, external help with support and even development and improved innovation and quality. They also offer some insights on why companies should use OSS in their ordinary business (2005).

The actual usage of OSS, aside from development, is of course of interest, since the users are potential paying customers. It could give a company who is interested in commercializing OSS some insights into market penetration or interest, and one part of the large European FLOSS research looked into just that issue.

2.2.6.2 FLOSS study – Use of OSS

The first report of the FLOSS study studied the use of OSS in “establishments”, a summary term used for both companies and public institutions. The survey, which was conducted by phone, queried 1452 establishments in Germany, Sweden and the United Kingdom. Out of these, 395 reported they were using OSS, or were planning to do so, within a year (Wichmann, 2002a).

The survey focused on two major objectives: firstly, to provide information on the actual usage of OSS, and secondly, to gauge the motivations for using OSS. What the report found was that the usage of OSS was greatest in Germany (and least in Sweden), and that most used OSS on their servers. Those who used OSS did it for a number of reasons, such as higher stability and independence from commercial licensing (Wichmann, 2002a).

In general, the first FLOSS report provided this research with hard data on usage and some insights regarding why companies and public institutions chose to use OSS products. The report also has implications for some of the research’s case companies, since at least one of them has their primary market in Swedish public institutions.
2.2.6.3 FLOSS study – Firms OSS activities

The second report uses a different method of research in comparison to the first. This report studied the (at the time) 25 largest software companies and their involvement in OSS projects through web sites and search engines. The researchers motivated this method with the claim that OSS development is a public endeavor, and that there are positive connotations that make it likely for the companies’ marketing departments to exploit their OSS involvement (Wichmann, 2002b).

Most of the companies did not show any visible OSS activity, but the report still indicated that most of them still used OSS, but were not actively contributing. In addition, for those who did contribute, the reasons for doing so ranged from standardization issues to low-cost component in product bundles to non-essential software such as OSS to compatibility issues. However, the study made some recommendations that could be called controversial. One was that the government should support OSS projects, since the providing companies’ efforts were considered sub-optimal. However, if such projects were encouraged, the GPL and similar licenses were not considered optimal, since they diminished further exploitation (Wichmann, 2002b).

Interestingly enough, the latter statement would be in direct violation of one of the most important factors for public institutions to use OSS, namely the freedom from commercial licenses; this is something from their previous report. In addition, this research also indicates that GPL is the preferred type of license of both companies and developers, since it prevents unauthorized “theft”.

2.2.6.4 FLOSS study – OSS Markets and Business Models

From the point of view of this research, the most interesting report regarding this work would be the third report. The report is a theoretical background for the whole FLOSS project and aims to analyze the phenomenon, the market and business models of OSS in addition to “best practices”. The report also defines how the software market operates in general and for OSS in particular (Wichmann and Stiller, 2002).

In the final part, Wichmann and Stiller (2002) describe OSS business models based on offerings and the revenue models. They then divide the companies into two groups, distributors and retailers and OSS-related services. Each of the business models is researched by asking three questions: what is the company’s main product or service offering, who are their customers and competitors and do they earn money or not? The models are thereafter described in some detail, together with a discussion regarding the market and the advantages and disadvantages of each business model.

However, the report is clearly indebted to Raymond (2001), both for how the business models are described, but also in that they are all based on what Raymond calls “non-direct sales values”. Moreover, the lack of empiri-
cal data is another flaw of the report. For instance, even if the dual licensing offering of MySQL is mentioned, it is not considered their main source of income, which it was at the time, and still is.

2.2.6.5 Gaming theory and other methods

A number of articles have presented attempts to utilize various formulas, such as gaming theory, to determine the feasibility for firms to use or adapt OSS in different scenarios such as software development (Johnson, 2001; Harison and Cowan, 2004), comparisons between OSS and proprietary software (Dalle and Jullien, 2001; Mustonen, 2003), the effects of OSS for a competitive firm (Hawkins, 2004), OSS adaption (Chi, 2007), and dual licensing (Comino and Manenti (2007). These articles usually show, and with differing degrees of clarity, that OSS is very well feasible in these settings. However, the scenarios are of course simplified, which hampered their usability for this thesis. Nevertheless, these articles span a theoretical bridge that enables an understanding of the underlying theorems that makes OSS “tick”.

2.2.6.6 Balancing closed and open source

One important part of business strategies for OSS regards how to balance what code is open and what is not. The protection of Intellectual Property Rights (IPR) has been one of the toughest nuts to crack regarding commercial interests and OSS. West (2003) investigated three multinational corporations and their strategies regarding some of their software. Apple, Sun Microsystems and IBM adopted different strategies for their proprietary and OSS platforms. Where IBM adopted Linux through their entire hardware platforms in their shift to become more services oriented, where software commoditization would not hurt their revenue streams, both Apple and Sun had to resort to different strategies. Interestingly enough, they adopted two different hybrid strategies, as West refers to them.

Apple decided to base their new operating system (Mac OS X) on the OSS operating system Free BSD, and Apple decided to grant all rights for a subset of their new system. West calls this strategy “opening parts”. Sun chose another direction; they released their entire source code for Java (a programming platform) and Solaris (operating system), but using rather restrictive licensing terms. This strategy West refers to as “partly open”. Sun also “submitted” to the Linux hype and started to offer Linux on their hardware, in the strategy called “if you can’t beat them, join them” (West, 2003).

West also proposes a “three stage evolution” for proprietary vendors to go to OSS. The time a company can sustain proprietary software is limited and only viable for a few market leaders. A starting point is to modify the software to incorporate open standards. The next step is to adopt one of the hybrid standards before taking the full step out (West, 2003).

Even if West’s article is some years old, the strategies for the chosen com-
panies have become even more polarized. IBM still considers OSS as a key strategy, opening new technologies and platforms up (IBM, 2008). Sun has turned all the way around and is opening up all their technologies to OSS, even adapting GPL as their license of choice, and purchased MySQL (La-Monica, 2006a; MySQL, 2008; O’Brien, 2008). Apple, on the other hand, seems to have turned back towards a more proprietary strategy again, fueled by their success from iPod and iTunes (Burrows, 2006).

Goldman and Gabriel (2005) argue that companies that own the copyright of the source code and use dual licensing or other hybrid licensing model should expect less community contributions. This may very well be the case, but in some circumstances, it may be hard to gauge. There are companies that have commercial models and yet retain a large community of developers; perhaps the developers have begun to accept the fact that OSS is becoming more commercial? There have been changes in the community, which the following articles clearly show.

2.2.6.7 OSS 2.0?

One of the pivotal figures of OSS is Bruce Perens, who together with Eric Raymond founded the Open Source Initiative, and whose policy document for the Debian Linux project became the Open Source Definition, which defines what software licenses are acceptable for OSS. Perens investigates the role of software in business and the paradigms for software development and their effectiveness. He demonstrates that the retail paradigm, the one connected to Microsoft, is really a small part of the total production of software. He also describes the ineffectiveness of that paradigm, and argues in which cases the OSS paradigm is more effective. His view is that the software that does not differentiate a business is more cost-effective than all other paradigms. Moreover, since most software is of this category, most companies would benefit by using and contributing to OSS (Perens, 2005).

Perens then moves on towards who contributes to OSS and how commercially effective the different contributors are, and identifies a number of different business strategies:

- **Proprietary Open Source** – the company is a software integrator or distributor that packages their services in a similar way to proprietary software vendors.
- **Mixed Open Source and proprietary software** – the company offers the same software under different license models [termed dual licensing below].
- **Open Source with proprietary accessories** – the company adds proprietary components to an OSS.
- **Open Source plus Services** – a company offers services for a particular OSS.
- **Hardware vendors**: a company that manufactures hardware subsidizes OSS development.
• **End-User Business plus contractors** – a company that utilizes OSS in order to provide content online.

• **Service Business** – a company that customizes OSS packages for a particular customers or service for a selected collection of OSS.

Even if Perens’ main discussion is about how different entities contribute to OSS development, his descriptions of the commercial interests in OSS provide us with insights in these matters. One of the examples he uses, Red Hat, has since the article was written proved that their business model is financially viable, something that Perens was not sure of (Perens, 2005).

Dahlander (2005) compares the commercialization of OSS with the commercialization of academia and the commons. Even if the focus of the article is focused on the comparison between academia and OSS, while studying new entrants within in the OSS market he found three strategies for how the companies commercialize OSS:

• **Founding projects** – the company releases source code under an OSS license in order to gain resources for developing and testing.

• **Sponsoring projects** – the company sponsors one or more OSS projects that are relevant for their business. The sponsorship comes in different forms, ranging from computer equipment to paid employees.

• **Using projects** – the company makes use of existing code in their own products or offerings.

Since Dahlander’s focus has been on the tension between commercialization and the commons, he has not discussed the strategies in depth, but the scope of his investigation makes it interesting to see how new entrants relate to OSS.

There is no doubt that the landscape of OSS has changed since commercial interests staked claims in the community. Fitzgerald (2006) has dubbed the term OSS 2.0 in order to describe the changes. He presents the evolution throughout the whole concept of OSS, from development to business strategies and licensing.

From the point-of-view of this research, Fitzgerald’s analysis is spot on when he describes the changes in not only the way the software is produced, but also the major changes regarding the actual users of OSS Customers are now willing to pay for professional support. In addition, he presents the business strategies that he feels have evolved from earlier strategies. The strategies are grouped into four categories: **value-added service enabling**, **market creating**, **leveraging community development** and **leveraging the open source brand**.

Value-added service enabling treats OSS as a platform and enables the company to bootstrap\(^\text{10}\) their special services or competence on top of the

\(^{10}\) Bootstrap connotes in this case the possibility to add abilities a firm normally would not be able to without OSS.
OSS.

There are four different market-creating strategies. The first is *Loss-leader*, where a company subsidizes OSS in order to create a market. The second, *dual licensing*, where the company offers two different licensing options for the same software. The third is *cost reduction*, where OSS provides low cost, reliability and interoperability between platforms, thus reducing costs for a customer. The fourth is *accessorizing*, where firms offer products other than software, for example books (Fitzgerald, 2006).

A company can leverage community software development and thus get free programming, which increases the functionality of the product. Fitzgerald also calls this phenomenon circular, since the more functionality the software gets, the more interest it attracts, thus completing the circle.

The final strategy, leveraging the open source brand, stems from the fact that government institution increasingly prioritizes OSS in their solutions. It is therefore important for a company that focuses on governmental customers to adhere their brand to OSS (Ibid.).

Another important economic factor for a commercial OSS company is how to handle their intellectual property (IPR). In addition, the revenues are directly dependent on which OSS license the firm uses, since the license may narrow or widen up the possibilities for different models to gain revenue. Therefore, OSS licensing plays an important part of the firm’s commercial success.

### 2.2.6.8 OSS licensing

When Richard Stallman created the GPL, he made use of the Copyright Laws of the United States. His agenda was to ensure that the source code would be accessible and that nobody would be able to hoard free software in order to make them proprietary and make a profit without benefiting the original authors (Stallman, 2001).

His creative use of the law and the characteristics of the license have caused different reactions from various agents. Microsoft condemned it and Linux as un-American, communistic or a cancer (Greene, 2001; Lettice, 2002). Others, with a more positive attitude, may even consider it the only OSS license from which you can create a business.\(^\text{11}\)

The “infectious” part of GPL has been discussed in and outside of the OSS community. Vetter (2004) shows that the “hard” terms may hurt the OSS movement, since the balance between OSS and proprietary software is impaired, pointing for example to Linus Torvalds modification of GPL 2.0. He considers the newer OSS licenses as better balanced for this matter.\(^\text{12}\)

\(^{11}\) Interview, Michael Tiemann, founder of Cygnus Solutions, currently Vice President Open Source Affairs, Red Hat Inc., October 25, 2005.

\(^{12}\) Even if it may appear so, Vetter does not miss Stallman’s points with GPL, but he argues for the benefit of the OSS initiative rather than FS, so his article could be seen as a part of the discussion between OSS and FS.
Lerner and Tirole (2005) studied the scope of OSS licensing and found that, in opposition to proprietary software, the decision on which license to use was not just based on the preferences of the licensor itself, but had to consider the community of users as well. “For instance, a commercial company releasing software to the open source community may choose a more restrictive license because of suspicion about its ultimate intentions” (Lerner and Tirole, 2005, p. 53).

Goldman and Gabriel (2005) take a wider stance on licensing and discuss not only licenses for OSS, but semi-open licenses, called gated communities, such as Microsoft Shared Source Initiative and the Sun Community Source License (SCSL).

Fitzgerald (2006) points out a shift in OSS licensing in his OSS 2.0. He predicts that corporate type licenses, for example Mozilla Public License (MPL) and IBM Public License (IPL), are becoming more important. He also foresees that border licenses, such as the Microsoft Shared Source Initiative, will be more common. However, the development of GPL Version 3 and the recent initiative from Sun Microsystems indicates that even large companies are more likely to accept the reciprocal attributes GPL offers and the signals it sends to the OSS community.

Olson (2006) discusses the dual licensing strategy as being one of three for commercializing OSS. The two other are service business, where the company offers service to OSS and hybrid business, where the company uses the OSS as a platform and adds proprietary software from which the revenue is collected.

Olson considers dual licensing as an innovative way to combine the benefits of both OSS and proprietary software, using the availability of the OSS to create a large and inexpensive distribution network and still gaining revenues for the software, as well as being able to bundle third-party software to the product. He points out that the company using dual licensing has to own the copyrights to their own products, which makes it harder for them to make use of community contributions. He states that “dual-licensing business relies on open source as a distribution strategy, not as a production strategy” (2006 p. 79).

The last statement may be somewhat moderated by individual cases, since the firm can make use of other strategies, such as licenses, which enables them to make use of the community contributions.

2.2.6.9 Licensing schemes

While Raymond and Stallman argue that the “Direct-Sale Value Model” is not applicable for FS or OSS, the fact remains that the copyright owner of a

---

13 The SCSL was originally used for technologies such as Java, but it was released under a new, more open license in 2006 (LaMonica, 2006).
software may retain the right to use a proprietary license alongside GPL (Stallman, 2001; Raymond, 2002; Bonaccorsi and Rossi, 2003; Välimäki, 2003).

Välimäki describes three cases – SleepyCat Software Inc., MySQL AB and Trolltech AS. They mainly offer the proprietary license to other companies that embed the OSS inside their proprietary software. By granting these firms a proprietary license, the viral effect of GPL is nullified. The use of GPL does have a strong network effect for the companies involved, which includes marketing, political and some developmental benefits. However, the legal requirement for dual licensing is that the “company has undisputed rights to the software product it wishes to dual license” (2003 p 71).

All of the companies involved have done most of the development of the core product inside their companies, while the community mostly develops auxiliary software to the core products.14

Bonaccorsi et al. studied the Italian software industry concerning OSS, in particular how firms entered the industry and the licensing schemes they used. They noticed that firms actively participating and working with OSS invested more in OSS but also more often used copyleft licenses, but very few firms had adapted a pure copyleft licensing revenue model. Most, about 60% combined the copyleft license with other licenses, including proprietary (Bonaccorsi and Rossi, 2003; Bonaccorsi et al., 2003).15

The reasons for the use of copyleft licenses were attributed to the use of Linux-based products, strength of social motivations and various involvements in the OSS community. Most of the firm had active relationships with the community. While the firms did not contribute much to the production process of OSS, they were much more involved in creating a larger user community, thus adding to the distribution process of OSS. The authors also noticed that since the firms used mixed licensing models, the implied worry about the “viral” property of the copyleft licenses was incorrect. The studied firms were instead well aware of the legality of different licensing schemes (Bonaccorsi and Rossi, 2003).

Välimäki and Oksanen studied the three main operating systems for desktops: Microsoft Windows, Apple OS X and GNU/Linux, and the differences in their business models. They concluded that the market is still dominated by Microsoft, which profits from its proprietary license and development model. However, Apple OS X, using a mixed license (based on FreeBSD) and development model and GNU/Linux have affected the market, creating open standards and stability for the operating systems. Even if the open source development model is socially more preferred, the proprietary model is still generating more revenues (2005).

14 Interview, Mårten Mickos, CEO, MySQL AB October 28, 2005.
15 Examples of copyleft licenses are General Public License (GPL), Library (or Lesser) General Public License (LGPL) and Mozilla Public License (MPL).
2.2.6.10 Enforcing the copyleft

What happens if someone breaks the license of OSS? Does the copyright owner have any protection? Proprietary software developers usually use legal means to sue the perpetrator, but the practice differs for OSS. Their approach is softer as their aim is to make the offending company to release the source code for the benefit for the whole community (Välimäki and Oksanen, 2006). The stricter enforcements of copyrights in the developing countries (who show very high rates of pirated copies of proprietary software) may bolster the development of OSS in these areas. There are already examples of developing countries actively promoting OSS in order to boost the software development in their countries (Ibid).

In effect, the copyleft licenses could be considered to promote competition. Välimäki studied how copyleft works in the light of EC Competition Law, and found it was closer to the goal of the laws than the current proprietary licenses (as exemplified by Microsoft licenses) (Välimäki, 2006).
3 Strategy, management and business models

In order to obtain an overview of the field of managerial research concerning strategies, processes or management, a starting point is usually essential. For example, a starting point helps one to categorize and catalogue the myriad of books and articles that have been produced during the last five decades on the subject.

In this research, Strategic Safari (Mintzberg et al., 1998) was used as such a starting point. It categorizes the different research streams into ten different schools of strategic and management theories. The other works presented herein have been chosen since they either directly or indirectly influenced the formation of the business model framework. Some of the research streams are old – but still refreshingly and even surprisingly contemporary in modern times.

This research's presentation is thematic, but also arrayed semichronologically, under four different sections: long-term strategy, short-term strategies, vision and culture and the business model. An attempt has been made to explain how each term was used in order to categorize the different schools of thought, beginning with strategy.

3.1 Long-term strategy

Strategy is an old word with a wide array of different connotations, but it usually implies planning and long-time formation of goals and even doctrines (Merriam-Webster, 2007). One way to see strategy is to say that a strategy aligns the organization towards its goal. This means that the strategy sets the goal, calculates and allocates the resources needed and formulates a general plan on how to achieve the set goal. In order to make the calculations and the formation right, a strategist needs information on opposition, technology and topology. Therefore, the use of the word strategy in the managerial literature is not very far-fetched; in fact, von Clausewitz compares business (or commerce) with warfare (Clausewitz, 1989).

Even if Mintzberg adheres the military maxims to the school of thought he calls the positioning school (1998), there are indications that earlier work, for example by Chandler (1962) and Ansoff (1965), were inspired by military sources. Chandler defined strategy as “the determination of the basic long-term goals [...] and the adoption of courses of action and the allocation of resources necessary for carrying out these goals” (1962, p.13).

3.1.1 Structure and Strategy

Chandler introduced the hypothesis that structure should follow strategy (1962). The structure was supposed to support the strategy because when a company grew larger, the pressures on the executives would be too great to handle. Therefore, he introduced four basic steps that a firm would encounter during its lifetime. The first was the “expansion of volume”; the second,
the “geographic dispersion”; the third, “vertical integration”; and finally, “product differentiation”. All these steps needed new structure to accommodate the changes in strategy (ibid.)

His general thesis was not concerned about the structure, but that the growth of companies came from an “awareness of the opportunities and needs [...] to employ existing or expanding resources more profitably” (Chandler, 1962, p. 15). Moreover, new strategies demanded changes in the structure to be able to be proficient. He stated that if a firm failed to alter its structure, it was because the top executives were either overworked or inadequately experienced or trained, a very normative view indeed. He also concluded that “growth without structural adjustment can lead only to economical inefficiency” (ibid., p. 16).

3.1.2 Planning

The school of thought which started with Ansoff (1965), took the strategic thinking even further, by introducing formal plans, audits and well-defined steps ready-made for implementation in the organizations. The concept of planning naturally also included forecasting. Ansoff himself defined the ideal length of planned time for the firm as the “period for which the firm is able to construct a forecast with an accuracy of, say, plus or minus 20 percent” (1965, p. 44).

The natural question immediately springs to mind: How long in advance is such a forecast possible? Perhaps not that long; as Field Marshal Moltke stated, “No battle plan survives contact with the enemy” (Wikipedia, 2007b).

However, to some extent, presumably in certain mature markets or industries, strategic planning might work – for instance in the extraction of oil or in the mining industry. In these industries, the planners have adequate information about conditions that would affect production – like projections on the amount of raw material that is left to profitably exploit – for a long time ahead. In addition, while the demand for oil (or gold) is high, there are small changes that would impinge on the plan. The only thing that fluctuates is the price on the product, and how these changes affect the extracting firm should be calculated, or articulated, in the planning, and well in advance.

However, the use of this theory had some serious setbacks, which diminished its use (Mintzberg, 1998). Nevertheless, these setbacks also opened up the research field and led to new managerial theories that attracted both researchers and students.

3.1.3 Positioning

One of these theories was spearheaded by Michael E. Porter (1980, 1985). As the theories above, the new theory offered a rather normative set of rules on how to make a firm a competitive one. Porter looked to the battlefield of business itself. The strategic position is the paramount recipe for success. This notion is of course not a new one; how to deploy your troops in the ter-
rain is an important part of any military strategy.

Porter constructed three “generic” strategies: overall cost leadership, differentiation and focus. He also constructed an analytical tool, based on the five forces which affected the firm’s position on the market: the threat of entry, the intensity of rivalry among existing competitors, the pressure from substitute products, the bargaining powers of buyers and suppliers. What he meant was that a firm must select only one of the strategies and stick with it, otherwise the risk was to be “stuck in the middle”, which would ultimately lead to the demise of that firm (Porter, 1980).

Porter later added the value chain concept as a complement to the five forces model to describe a basic concept on how a firm could create a low-cost or differentiation strategy (1985). This was an attempt to break into “the black box of the firm” and set the boundaries of the company. By configuring the different parts of the value chain these parts will add value and thus create competitive advantage for the individual firm (Hedman and Kalling, 2002).

The resource-based-view (RBV) (see below) was created as an answer to the market positioning school, addressing issues that the authors felt were badly covered in Porter for example. Porter answered by adding some of their thoughts when he addressed the need to find a more dynamic theory of strategy (1991). The article included the original thought regarding the market position, but also the value chain and the diamond of national competitive advantages added together with his view of the rivaling thoughts of RBV and game theory (ibid.).

Interestingly, a short time after one of the more known followers of RBV, Katherine Eisenhardt, published an article in Harvard Business Review regarding the “new economy” and RBV (Eisenhardt and Sull, 2001), Porter made a stern answer, and not only falsified the “new economy” concept, but also moved away from RBV (Porter, 2001).

3.1.4 Resource-based view (RBV)

The final school of thought of the long-term strategic part is the resource-based view (shortened RBV). The hypothesis concentrates on getting inside the “black box” of the firm (which Porter somewhat remedied in 1985). One of the early researchers of this strategic school was Birger Wernerfelt (Wernerfelt 1984, in Mintzberg et al. 1998).

The main feature of the theory is the interest of the resources and capabilities that reside inside a firm. Wernerfelt argued that new insights, especially regarding diversified firms, could be found, but also that new kinds of entry barriers, which he called resource position barriers (ibid.), affected the firm’s ability to obtain competitive advantages.

The connection to the positioning school or to the value chain is quite clear, as Barney says that the RBV “simply pushes this value chain logic further...” (1991, p. 105). There is also a connection to strategy. One of the rec-
ommendations given by RBV is to “hoard” resources and develop capabilities in order to exploit them; to obtain a sustainable competitive advantage, you then have to protect them.

However, not just any resource was valid; it had to be “right”. Barney labeled the correct resources as valuable, rare, imperfectly imitable and non-substitutable (VRIN) (1991). It is easy to understand that these kinds of resources really are rare and hard to obtain.

However, what happens with the resources during time? One part of the RBV has been interested in a “holistic approach”, where the resource life cycle and resource management affects the organization, activities, and market position of the resource. For instance, resource management might consist of five phases: identification, development, protection, internal distribution and usage (Hedman and Kalling, 2002).

One of the problems connected with RBV was the assumed inability to handle change. Hence, the aspect “dynamic capabilities” was added to the concept (Teece et al., 1997).

3.1.5 Dynamic capabilities
The dynamic capabilities were added to the RBV by Teece and his colleagues in 1997 as a way to describe how some capabilities could be shared across firms also could be essential in helping to attain competitive advantage, especially when it came to demanding environments (Teece et al., 1997). The term “dynamic capabilities” was created to describe the way the firm adapted to the environment by using managerial skills to adjust the firm’s resources and skills accordingly (ibid.).

So not only did the resources and capabilities matter, the way these were deployed and renewed in accordance to the change of the environment was the key element of this theory. In addition, the dynamic capabilities come in terms of processes, positions and paths, where the first points to internal workings of the firm, the second to external matters and the last to strategic alternatives (Teece et al., 1997).

This firm’s history and capabilities also makes it path dependent; there are just so many alternatives that a firm can make use of, and in the end this means that the finer workings of the dynamic capabilities are more or less unique for each firm (ibid.)

Eisenhardt and Martin clarified the term further as meaning that the dynamic capabilities might be shared across a variety of firms, perhaps as “best practice” (2000). In addition, even if these processes or routines are shared across an industry, they may they differ between industries or even companies. In a stable market, the dynamic capabilities are considered more as elaborate routines, but in more dynamic markets the capabilities are

---

16 Later Barney and Hesterly (1996) introduced the term “VRIO” where the “O” stands for Organized.
rather simple and experimental processes (ibid.)

Nevertheless, because of this, even if the details of the shared processes may be unique, there are certain commonalities between firms. This is because the options are limited when it comes to addressing certain problems associated with a specific capability, hence the “best practice” (Eisenhardt and Martin, 2000).

Nevertheless, what the dynamic capabilities really do is to connect the stable, strategic line of thinking of the RBV to the dynamic, tactical necessities for survival.

3.2 Short-term strategies and typologies

Short-term strategy hypotheses are more concerned with current, and perhaps even smaller, more isolated events, and affect a much shorter period than long-term strategies. In a firm the difference can be seen like this: while the board of directors formulates and evaluates strategy, the managers are more concerned with operational matters.

A good example of the difference between them:

But as we know, there are wars in which the enemy is difficult to engage, battle dynamics fluctuate, and the terrain is treacherous and unknown. Here, the strategy of choice is guerrilla warfare – moving quickly, taking advantage of opportunity and rapidly cutting losses (Eisenhardt, 2002).

Here the strategy is to use “guerrilla warfare”, which means the troops use other ways to achieve the goals. Moreover, although the former managerial theories worked with strategic planning, other models are more concerned on how the firm interacts with environment. Some of the research streams are focused on processes, while others look more to the environment. However, most of the short-term strategy models are primarily concerned with how change is handled within firms, markets and environment. Notice that most of the theories still use strategies, which gives the firm an alignment or a “vector” in which to move within the environment.

3.2.1 Domains and uncertainty

The relationship between the firm and its environment is of course nothing new. When Thompson wrote Organizations in action (1967) he mentioned two types of strategic literature, the closed-system (or rational) and open-system strategy (called natural). The former he connected to literature where certainty was given, and the prime concern was planning and control in order to achieve improved efficiency or performance. The term “open systems” deals with uncertainty, but also firms’ interactions with their environment (ibid.).

Thompson made this definition: “[...]we will conceive of complex organizations as open systems, hence indeterminate and faced with uncertainty, but at the same time subject to criteria of rationality and hence needing determinateness and certainty” (Thompson, 1967 p. 10). Moreover, he concludes
that a complex firm not only has to deal with a general uncertainty, but also contingency – where outcomes are determined by actions by the organization and of the environment – but also interdependence of components inside the organization itself (ibid.).

Thompson also said, “There is no one best way […] no single set of activities that constitute administration. Appropriateness of design, structure, and assessments can be judged only in the light of conditions, variables and uncertainties present for the organization” (1967, p. 162).

He introduced the concept of the domains, the environment where the company will work, offer products or serve particular customers. The domain together with the chosen technology determines how the firm will be situated when it comes to obtaining sustainable resources or other forms of support to ensure its survival. The claim of the domain is also determined upon approval from the actors (for example suppliers and customers) that are embedded in the particular environment, and this approval, or consensus, must be maintained during time and through changes, for example in technology, population and competition (Thompson, 1967).

Defending a domain can be costly, and Thompson proposes some suggestions on how firms can defend their domains and reduce the uncertainty through different kinds of cooperative and interdependent modes. Examples include: pooled, where two organizations share the same pool of resources; sequential, where for example one firm might fabricate, while the other assemble the final product; and finally reciprocal, where the organizational units are tightly dependent on each other.

### 3.2.2 Configurations

When Miller and Friesen (1984) looked to the firm, they linked their work with physics, stating that complex organizations are united in certain “quantum states and changes” or configurations. They felt that the contingency view was too simplified, and therefore took on the task of conducting a large study of firms to study not only the configurations, but also the transition between configurations.

Even if all organizations are unique, a small number of configurations encompass a large number of firms, which enables some kind of predictability regarding their future. A part of the work was to connect to the typology of configurations, based on theory, and while the major part created taxonomies based on empirical studies.

They presented five basic typological archetypes, which illustrates how the structure of different firms would look, depending on how the work is coordinated, and how the different elements of the organization will operate interdependently with the coordination effort (Miller and Friesen, 1984).

However, the extensive empirical work invokes the configurations or archetypes of real organizations. Their research identified nine different stable configurations; five of them were more or less successful, while the last four
were unsuccessful (Miller and Friesen, 1984).

When studying the firms that were in transition between the stable configurations they also found nine different transitional states. Finally, what they found to be the fact was that the transition is often revolutionary, rather than incremental (ibid.).

Miller added four imperatives collected from other managerial and organizational research environment, organization structure, leadership and strategy. These four imperatives are drivers, and aim to explain why a certain firm has chosen or ended up in a certain configuration (Miller, 1987). Later articles delve deeper into the reasons behind the configurations, for example the problem with external versus internal fit (Miller, 1992), simplicity (Miller, 1993) and competitive advantage (Miller and Whitney, 1999).

3.2.3 The Adaptive Cycle

As predecessors to configuration, Miles and Snow have a similar concept (1978). As they press the point on how managerial style determines the strategy and structure that a firm adapts to their environment, they are moving close to the focus of the next chapter (see below).

They call the model of the processes of organizational adaptation, which is actually a series of managerial decisions, The Adaptive Cycle. The cycle consists of three problems, which the top management has to balance and properly relate to each other at all times. First, is the entrepreneurial problem, which consists of selecting a market and objectives for it; second, the engineering problem, to create technological processes to service the selected market; and finally, the administrative problem, to create a structure and processes for the organization to control and coordinate the firm (Miles and Snow, 1978).

They have also identified four different strategic types, each with a unique pattern of adaptation that most forms of organization can use: Defender, Prospector, Analyzer and Reactor. The first three are considered stable, with sets of processes to adapt to changes in the environment, and as the Reactor lacks these processes or mechanisms, it is considered unstable and in a transitional state.

Miles and Snow meant that the adaptive cycles and the four strategic typologies above could be used in managerial work. Their conclusion is that when it comes to human resources, the manager influences the diagnosis and change in organizations. They also introduced a new organizational form, the market-matrix, which would be flexible, efficient and able to use a wide array of human capabilities (1978).

Finally, the issue of learning, described as a limited and expanded search, explains some of an organization’s behavior, good and bad. Some limit their search for new approaches and develop “distinctive competences” which help them compete, but at the same time their ability to address fundamentally different problems decreases, as well as their organizational learning.
The organizations using extended search would expose themselves to risks, but on the other hand, be better equipped to handle unexpected situations. However, since most firms must be focused to compete, a good way to get valuable information is to get it from outside the organization, e.g. through consultants, external board members or venture capital committees. Even if these sources are used, management must be able to assimilate it in order to learn (Miles and Snow, 1978). Only then can the firm be fully successful for the long term.

3.3 Vision and culture

While the vision and culture may represent something in common for both strategy and tactics, the aim for this research is to understand how the cultural processes within an organization really work. While the vision clearly relates to strategy as it aims to align the organization towards its goal, it is not a strategy formulation per se, and often lacks the components describing how it should be done.

In addition, while the organizational culture may be seen as something benevolent, it may be conceived as something that holds back needed change within organizations. Nevertheless, culture is something every organization develops, whether they want it or not.

3.3.1 Creative management

Firmly based on empirical material, mostly Swedish, Normann (1975) introduced the concept of the aligned organization, which is a cultural and social unit where each individual fully understands and embraces the firm’s business idea, growth idea or vision. The business idea is a system for dominance. In order to become fully functional, there must be a congruence of the actual market segment, the product or service offered and the internal structure and control systems of the firm. The business idea is also a manifestation of how the firm is working presently (Normann, 1975).

Since the business idea is concerned with the present, the growth idea or vision is about the future. The vision is never made permanent; it is supposed to change over time. It is not to be mistaken as a set of goals, but it has three clear uses for the firm: it controls the learning by pointing out the next step of the learning process; it controls the changes of the vision as a measure of the learning; and finally it makes it easier to create the driving forces in terms of ambition and engagement within the firm (ibid.)

It is important to handle tensions within the company, those that are connected to the business or growth idea. Normann points out that there are four major elements in the growth idea: structure, visionary and down-to-

\[ \text{structure, visionary and down-to-} \]

\[ \text{structure, visionary and down-to-} \]

17 The title of the English translation of Normann’s book is Management for Growth, but since the Swedish version is used here, the title is directly translated.

18 Business idea is a term directly translated from Swedish. Another word with a similar meaning is mission.
earth planning, management of tensions and the development of knowledge (1975).

He also presents the significant actors, who are the key elements concerning the dominant ideas and the power system, and their roles in different growth scenarios within a firm. He discusses what he calls “the ecology of dominant ideas” or growth culture, in which the organization gets involved with the ideas and how to manage tensions in the organization. Finally, the concept of the “leader as statesman” is introduced to illuminate on how leadership affects the growth culture (ibid.).

In addition to this, a whole array of concepts was addressed, such as organizational structures, fit, values, learning and images, among others. Mintzberg et al., adds that the work encompasses a lot of concepts from other schools, but the main concept of collective context or “reframing” qualified it to this typology (1998).

3.3.2 Vertical Market Systems

Normann inspired many researchers, especially in Sweden, after his first book. One of them was professor Staffan Brege. In his dissertation (1979), he both agrees and disagrees with Normann regarding the business idea. He studied change in supply and demand in Vertical Market Systems (VMS), and his findings made him add a dimension to the business idea concept by noticing a substantial difference between production-oriented firms and market-oriented firms.

The production-oriented business ideas are based on the fact that customers’ requirements are standardized, and that the firm’s ability to have efficient manufacturing, procurement and distribution processes makes it unique. Their aim is to produce products of competitive quality and pricing. In addition, for companies working with large-scale production, the latter aspect is often the most important. Production-oriented business ideas lead to vertical integration based on rationalities aimed at stabilizing the large-scale production.

The business ideas of market-oriented companies, as opposed to the production-oriented firms, are based on the ability to identify and satisfy individual customer’s needs. This ability is what makes the firm unique. The production is customized, and often sold as whole systems. Market-oriented business ideas lead to vertical integration mostly to withdraw from production-oriented competition and to increase the firm’s control of the VMS.

In addition to this, Brege also adds a dimension to the three components that Normann (1975) argues must fit: market segment, products or services, and internal structure, which is namely the firm’s raw material markets. Changes in these may affect the company to the same extent as changes in the market (1979).
Reframing business

In his new book from 2001, Normann discuss the new demands on successful companies, how the industrial paradigm has moved from production through market orientation and customer base management to the reconfiguration of value creative systems, where the customers are seen as a part of the process of creating new values. The change in the economy he explains with the principle of density, where the technology has limited the limitations and enabled a dematerialization of information and knowledge leading to the unbundling and (re)bundling of products and services, which creates density. This leads to profound changes in value creation and opens up new areas for exploitation. Moreover, through the principle of vacuum, in which nature always strives to fill empty niches, he envisions a new breed of firms that will utilize these areas (2001).

Some organizations he calls Prime Movers, which are good at innovate new ways of creating value. These firms use the vacuums created in new kinds of market, and are keen to reconfigure themselves. To become a Prime Mover demands a defining role of each individual mentality, or even as a state of being. For example, a Prime Mover might think of themselves as a part of their customers’ internal activities. They have also developed a certain capability, which enables them to mobilize and manage external actors outside their own company. This capability is both a network vision and a capability to set up actors with the right mix of assets and capabilities to introduce a new value creating system (Normann, 2001).

Normann uses the term proactive ecogenesis to describe how the Prime Movers shape or manipulate their environment in order to create infrastructures that influence the co-operative networks that form the base of their business. Part of the concept is to lay the foundation for co-production, which is the true source of the success. Moreover, the results of the cooperation are called offerings, which substitute the view of the transaction economy. An offering consists of configurations of value-creating processes, rather than single products or services (2001).

As a conceptual tool to achieve, proactive ecogenesis, Normann introduces the “crane”. In order to function properly, a crane needs a solid foundation, and its function is to reconfigure the organization by adding overview and access to new areas. This is a mental process, and Normann shows how the firm can map its past, present and future and provides an example on how he had used the crane as a tool (ibid.)

Finally, he connects to his earlier work (1975) in describing how the management should act in order to move fantastic ideas into fantastic firms, especially when it comes to handling tensions and changes in and outside the organization. As an interesting final point, Normann mentions how the need for reconfiguration or reframing will change the business models for the companies’ involved (ibid.). This is a good way to enter the final destination,
the emergence of the business model concept.

3.4 The business model

The concept of the business model is often connected with Information Technology (IT) and the dissatisfaction with existing strategy theories which cannot, one by one, explain how value is created when the business is exercised on the Internet (Amit and Zott, 2001). The business model concept could be seen as an answer to the fragmentation of the strategy theory, and some empirical implications indicate that instead of creating strategic plans, modern firms formulate business models (Hedman and Kalling, 2002). Even if organizations do not formulate their own business models, they can be used by external parties to understand how the organization operates.

3.4.1 Definitions

What is a business model? Michael Rappa starts his online lecture with the words: “Business Models are perhaps the most discussed and least understood aspect of the web” (Rappa, 2000). In addition, this sentence exemplifies part of the problem. Business models were strongly associated with the Internet Boom, and after the “bubble bust”, the concept quickly fell out of fashion (Magritta, 2002).

Nevertheless, the concept has other definitions. Amit and Zott (2001, p. 494 f.) define it as “the business model depicts the design of transaction content, structure, and governance as to create value through the exploitation of business opportunities”. Afuah defines it as a “framework for making money” (2003, p. 2), while Magritta (2002, p. 87) views it as “stories that explain how enterprises work”. Chesbrough and Rosenbloom (2002, p. 532) say the concept “provides a coherent framework that takes technological characteristics and potentials as inputs, and converts them through customers and markets into economic outputs”.

It is plain to see that the definitions vary, but there are also differences regarding their uses. While some see business models as descriptions of the way firms conduct their business (Magritta, 2002) or simple formulas of business concepts (Kraemer, et al., 2000; Rappa, 2000) others use the concept as an analytical tool to investigate organizations (Amit and Zott, 2001; Chesbrough and Rosenbloom, 2002; Hedman and Kalling, 2002; Afuah, 2003).

The storyline or the simple formula concept will not be further discussed here, but these aspects of the business model are still viable as dimensions of business models. Before revealing our use of the business model, it makes good sense to show how other authors have used the business model as an analytical tool.

3.4.2 Value Creation

Amit and Zott (2001) found through their study of a number of American
and European e-business that not any of the existing strategic or entrepreneur management theories single-handedly could fully explain how value was created in e-business.\textsuperscript{19} The solution was to construct a business model that used several of the theories in order to explore e-business. Their model implemented value chain analysis, Schumpeterian innovation theory, RBV, strategic network theory and transaction cost economy. This led to their definition of the business model as follows:

Definition: A business model depicts the content, structure, and governance of transactions designed so as to create values through the exploitation of business opportunities (Amit and Zott, 2001, p. 511)

By using this definition, they covered the goods or information exchanged, the parties involved in that exchange, and how the flow between the parties is controlled. Moreover, even if the business model is connected to an individual firm, the model in itself could encompass multiple firms in different industries (Amit and Zott, 2001).

However, Amit and Zott clearly differentiated the business model from plainly generating revenues. They setup a separate definition for this, called revenue model:

Definition: A revenue model refers to the specific modes in which a business model enables revenue generation (2001, p. 515).

The main difference between the two is that the business model regards value creation and the revenue model is concerned with value appropriation. When it comes to the factors that determine the business model, Amit and Zott concluded that there are four factors to reckon with, efficiency, complementarities, lock-in and novelty (2001).

Efficiency stands for transaction efficiency and shows how effective an e-business is to deliver the information or goods to the customers. The term complementarities stands for bundling of products or services that are together worth more than separate. Lock-in was coined to describe how customers or strategic partners are motivated to repeat transactions or prolong and/or improve the association. Finally, novelty stands for the innovations created by the firm concerning products, services, distribution or marketing (ibid.)

The business model connected to the value creation and the implications of the use of several management theories was a great contribution to the research field, but at the same time, other researchers formulated other ways to utilize the business model as an analytical tool.

3.4.3 Business model as mediator between technology and business

Chesbrough and Rosenbloom (2002) use the business model in a different

\textsuperscript{19} E-business is defined as "business conducted over the Internet" (Amit and Zott, 2001, p. 493)
way. As stated above, they see the business model as a focusing device for managers to merge innovations with customers and markets in order to make money. They also integrate several management theories, from the “design school”, “planning school”, “contingency theory” and RBV, but also cognitive theories on managerial decisions. They offer a new and, as they say, “detailed and operational definition” to the business model:

The functions of a business model are to:

- articulate the value proposition, i.e. the value created for users by the offering based on the technology;
- identify a market segment, i.e. the users to whom the technology is useful and for what purpose, and specify the revenue mechanism(s) for the firm;
- define the structure of the value chain within the firm required to create and distribute the offering, and determine the complementary assets needed to support the firm’s position in this chain;
- estimate the cost structure and profit potential of producing the offering, given the value proposition and value chain structure chosen;
- describe the position of the firm within the value network linking suppliers and customers, including identification of potential complementors and competitors;
- Formulate the competitive strategy by which the innovation firm will gain and hold advantage over rivals (Chesbrough and Rosenbloom, 2002, p. 533).

Using these functions, a firm can use six steps in order to create a business model. Chesbrough and Rosenbloom then use this framework to analyze the business models of Xerox and six spin-off firms from Xerox in order to find valid models. What they learned was that if a firm reshapes an initial business model, it “creates opportunities to discover new mappings between technical potential and economic values” (2002, p. 551).

However, this does not mean immediate success, since the uncertainty of the market may render some business models worthless. Nevertheless, this is hard to foresee without testing, and since spin-offs are likely to construct new business models, this could well be a vital way for the mother company to learn.

3.4.4 Business models for IT

Hedman and Kalling (2002) have also, as Amit and Zott (2001) above, concluded that older strategy theories do not cope with IT. Their aim is to integrate relevant components into one model. This model will have a wider scope than strategic theories, as it will encompass “resource bases, competencies, activities, organizational structure, culture and politics, products, markets, competitors and environmental factors” (ibid. p. 104).

They base their model from on 1) an article from Michel Porter (1991); 2) the work of Richard Normann (1975; 2001); and 3) entrepreneurial theories based on Shumpeter (1934; 1950), Penrose (1959), McClelland (1961, in Hedman and Kalling, 2002) and McGrath and Macmillan (2000, in Hedman and Kalling, 2002).

---

Their definition of the business model contains seven causally linked components: customers, competitors, offering, activities and organization, resources, factor and production input suppliers, in addition to managerial and organizational, longitudinal process. A business model is due to change or evolve, and single firms may have different business models, but if the differences between them are great, the chances are that the different businesses are organized more or less independently. In addition, to be considered is that a business model could have external stakeholders, thus becoming part of a network (Hedman and Kalling, 2002).

In order to illustrate the different levels of the business model, the components have been broken down into three levels: at the top, the offering; in the middle, activities and organization; and at the bottom, resources. The business model is then surrounded by market systems (ibid.).

Hedman and Kalling discussed how different IT systems may affect different parts of the business model, such as CRM (customer relationship management), SCM (supply chain management), CAD (computer-aided design), embedded systems, survey systems, VCO (value chain and organizational), ERP (enterprise resource planning) and finally resource and knowledge management systems. They also discuss e-business models and how changing technologies (like IT) affect business models (2002).

The differences between Hedman and Kalling’s models and those previously presented may look small on the surface, but it will be clearer after the presentation of Allan Afuah’s framework. The American models seem to have a certain amount of economical practicalities, like an obsession with costs that is not present in Scandinavian models (see more below). This divergence might be dependent on other theoretical sources, cultural variations or other factors, some of which will be discussed in detail.

3.4.5 Strategic management approach to business models

Allan Afuah (2003) has a slightly different approach to the business model, and in his book, he describes his model in detail. Even his primary definition is different from the previous:

A business model is the set of which activities a firm performs, how it performs them, and when it performs them as it uses its resources to perform activities, given its industry, to create superior customer value (low-cost or differentiated products) and put itself in a position to appropriate the value (Afuah, 2003, p. 9).

As Amit and Zott (2001), Afuah make the distinction between a business model and a revenue model. The business model must incorporate components to both generate money and describe the cost, while the revenue model is solely about the first component. This means that two companies can have similar revenue models but different business models (2003).

Afuah’s business model has five parts: Industry Factors, Resources, Activities, Positions, and Cost. The analytical model contains seven components which need to be analyzed: competitive position, connected activities, com-
petitive and macro-environmental forces, critical industry value drivers, capabilities and resources, change and sustainability, and cost of performing activities (2003).

Mentioned before was that American researchers are very concerned by costs connected with business models; Afuah is one of these, spending several pages on methods to calculate costs. Nevertheless, this is also because he has reduced the market strategies for the company into two: low-cost or differentiates. In addition, since the firm should aim to maximize its return, the calculation of costs for the product and services is vital.

Afuah has derived his business model from a great variety of sources. The book is made as a handbook for use in higher education, and is therefore designed for this purpose. For each part describing the business model a great number of authors are presented. He has been an active researcher for some time, and his experiences and ideas from this career are of course prominent. He has also worked with Michael Porter, and this cooperation has made a clear impression on parts of the business model.

In short, Afuah’s business model is a very extensive framework for analyzing a wide variety of organizations. The book draws from a vast number of sources, all of which have been tested empirically. The second part of the book is dedicated to ten cases in different industries where the business model has been tested.

3.4.6 Analytical tool

Aside from Hedman and Kalling (2002), research regarding business models has been conducted in other universities in Sweden. Kindström’s (2003) licentiate thesis used a refined business model as an analytical tool to describe and test how a service company went towards market orientation using IT tools.

His business model draws its roots from the concepts of Normann (1975; 2001), Porter (1980; 1985), RBV (Barney, 1991; Eisenhardt and Martin, 2000, Prahalad and Hamel, 1990 etc.) among others. Nevertheless, the basic concept is that the business is reduced to three components: strategic positions, operative platform and offering (Kindström 2003).

In short, the offering consists of the products and/or services the firm presents to the market.

The operative platforms are the firm’s internal resources, processes, structures and technologies that it makes use of to successfully operate on the market. These must be devised in such a way that the firm can achieve competitive advantage and optimal performance with regard to its competition.

The strategic positions are how the firm relates to its external environment concerning its offering. Moreover, the external environment is the market in general, more specifically the industry with its customers, suppliers, resellers and competitors. It also reflects the other activities the firm makes on or
towards that market (like positioning itself) (Kindström 2003).

However, in order to be successful, the firm must also reach dynamism and fit between the elements. Kindström described this as a dynamic interchange between the strategic positions and the operational positions (2003).

3.4.7 Refined model and tool

In his dissertation, (2005) Kindström has refined his model and tool even further. One of the changes is that by viewing the business model framework as a process framework, the researcher is able to add time as an aspect. This makes it possible to trace changes or even forecast the steps of an upcoming (or desired) change.

Some minor changes are that the name of the strategic positions (Kindström, 2003) has been altered to market positions and the offering has a more prominent role, as the interaction between the other two elements.

Kindström used his new and refined tool to illustrate how three mature companies’ integrated e-business into their operations, and how this affected their business model (2003). The companies were in different industries, which illustrates, despite the use of e-business as their common factor, that the business model could be used in a general way.

The business model will be discussed even further when presenting the proposed Business Model Framework in the next section.

3.5 The extended Business Model Framework

The Business Model Framework presented here is a revised model created and used by Kindström (2003; 2005). The most important factor, which warranted the extension, was the proposed importance of the offering for OSS companies. Since OSS companies supposedly would not be able to use direct-sale value models such as traditional and proprietary software companies, the need for an extended model for the offering, one containing both revenue models and distribution models, was conceived as essential.

Another purpose for this framework was to create a generic model that could be utilized in different industries and for other purposes. Each element has been refined to be easy-to-use and understand, and not cluttered with functions and concepts. The different elements are also meant to be “filled with content” by the individual researchers, where the key factors for the industry they study could be attached into the framework. In this research, and attempt has been made to single out the key factors that seemed to be the most revealing and interesting.

However, notice that the business model framework is aimed towards a single business unit or firm. A conglomerate or multidivisional corporation may have different business models for each department, and these may or may not be managed through a corporate strategy. In some cases, even very large corporations may operate using a primary business model. In addition, a firm or a business unit may very well have more than one business model,
but there is usually one primary model that may be coupled with other supporting business models.

The base model that was the starting point for the extended framework consisted of strategic positions and operative platforms, with offerings as an interface between the strategic positions and the market.

![Diagram of Business Model](kindstrom_2003)

The first decision made was to rename the strategic positions as “market positions” in order to explain the relationship these strategic positions have to the market, brand, positioning, etc. The result was that the model still has the strategic elements embedded, but the thought was that important factors such as customers should be a part of the position. Another important issue was to remove the connotation that strategy is disconnected from the operational platform, which indeed it is not.

### 3.5.1 Market positions

The market positions are abstract values of the company on how the firm and environment interact with each other. The market positions should never be stable, even if they might look like it if viewed from the outside. The reason for this is that they contain two different aspects, Intended Strategy and Current Positions, which corresponds to the concepts of intended and realized strategy (Mintzberg and Waters, 1985). Figure 3 illustrates some examples of what market positions could contain:
3.5.1.1 Intended Strategy

The intended strategy encompasses business ideas, visions, and goals and aspired positioning. The latter is how the firm positions itself when addressing its competition by, for example, pricing. Following Porter (1980), this strategy can be overall cost leader, differentiation or focus.

The business idea concept is adapted from Normann (1975); the business idea is a system for dominance. In order to become fully functional, there must be a congruence of the actual market segment, the product or service offered and the internal structure and control systems of the firm. The business idea is also a manifestation of how the firm is working presently (Normann, 1975). Here, this is used to show the firm’s view of its business – the reason for it to exist. Written plans are here considered as a part of firms’ resources, and are therefore placed among the operational platforms. A strategic plan or business plan is often an investment in time and resources for the company. Even if the plan is part of how the company perceives itself on the market, the plan is often the foundation for operations, and thus a part of the operational platform. The written plan itself is the receipt of that investment, and is something that could be shown to investors or owners to verify that the planning has been done - and is therefore a resource in itself.

In addition, the visions are adapted from Normann (1975). The vision is never made permanent; it is supposed to change over time. It is not to be mistaken as a set of goals. It has three clear uses for the firm: it controls the learning by pointing out the next step of the learning process; its changes are a measure of the learning; and finally it makes it easier to create the driving forces in terms of ambition and engagement within the firm (Normann, 1975).

Goals and aspired positioning are things that are used in most organiza-
tions, which mean that their use differs between different firms. Their use in this context is rather the goal and aspiration itself, rather than the plans. The plans are, as stated above, part of the operational platforms.

The pricing strategy is based on Porter’s generic strategies (1980). Overall, cost leadership concerns a low-price strategy and economy-of-scale. Differentiation is achieved by a firm when it is perceived as different or unique from the rest of the industry, which allows for higher prices. Finally, the term focus means a firm is targeting a certain market segment to serve it as well as possible. Focus uses one of the other strategies in that particular market. Moreover, if the firm does not follow one strategy, there is an increased risk that it will be “stuck in the middle” (Porter, 1980).

3.5.1.2 Current Positions

Current positions are, as the title suggests, the actual perceived market positions of the firm at a given time. This perception may be dual, since it concerns both an outsider perception as well as inside of the firm. These perceptions might not always agree with each other, which might result in faulty decisions, etc.

The current position is not always easy to measure for the company or external parties. Usually terms like “market penetration” or “market leadership” are used to describe a current position. The position also illustrates the relationship of the firm and its environment; Figure 4 shows one way to depict these relationships.

![Figure 4: The firm and its environment.](image)

Different theories could also be used to describe the current position for a firm, for example the *value chain* (Porter, 1985). Another matter is the “culture”, which reflects the firm’s own view of itself and the environment (Normann, 1975). If the business idea or mission is the formalized view (or in-
tended), the culture reflects the actual view.

No matter what theory is used, it is still the perception of the position that matters in the market position. All other issues, like contracts or agreements, are a part of the resources of the firm, and thus are in the operational platforms element.

The current position also encompasses the views of the firm from the customers, competition and partners, or in other words, how a firm is perceived by the actors in the environment. This would also include government, media and other participants in the environment. Why is this important? Well, the view from outsiders might affect how the firm looks at itself; depending on the current position or standing in the environment, for example, a firm might want to adapt (to conform to the external view) or reform the external perception.

3.5.1.3 Dynamism between intended and current market positions

A vision should never be fully realized, and if a goal is achieved, there should be new ones on the horizon. There should always be a tension between the two aspects for a firm to function properly. A firm may become a Stagnant Bureaucracy if changes in the environment are not noticed and reacted to (Miller and Friesen, 1978).

For the individual researcher, there is a dilemma on whether the information about the intention and the current view is a true view or a matter of history revision. This research has found that the stories that conceive the culture of a firm may often be altered in order establish the intended strategy and positions.

The market positions impact on the operational platforms, and the current position and intended strategies will affect how the firm uses its resources.

3.5.2 Operational platform

The operational platform are the resources, or assets, which the firm uses to interact with its environment. The resources could be either tangible, such as raw material, equipment, inventories or such, or intangible, such as experience, knowledge and history. Even if some of the latter also can be shown in the market positions, a firm’s history may also affect how it utilizes its tangible assets. Figure 5 illustrates the operational platform:
3.5.2.1 Tangible resources

The operational platforms uses a terminology borrowed from the resource-based view (RBV) which originated with Wernerfelt (1984). The main feature of the theory is the interest of the resources and capabilities that reside inside a firm.

To have access to raw material, production equipment and a stock of inventories are considered tangible resources. In addition, signed customer or reseller agreements are also considered as tangible resources, since these agreements contain resources in the form of services offered by a partner, or resources in the form of revenues from the customers in exchange for services or products.

Intellectual property rights such as patents and copyrighted material are of course tangible resources. Nevertheless, even equipment that is common, like computers and software, are of course assets, but their true value lies in the way they are utilized. This knowledge is however something different, namely intangible resources.

3.5.2.2 Intangible resources

When Teece et al. (1997) introduced dynamic capabilities; they added the tacit dimension to the resources. The term “dynamic capabilities” was created to describe the way the firm adapted to the environment by using managerial skills to adjust the firm’s resources and skills accordingly. The dynamic capabilities are presented in terms of processes, positions and paths, for internal workings of the firm, external matters (which were moved for the extended business model to the market positions above) and strategic alternatives. As for the strategic alternatives, the capabilities and experiences of a firm make it path dependent, which means that the intangible resources
are unique (Teece et al., 1997).

Eisenhardt and Martin (2000) did however claim that some capabilities could, or should, be shared across firms in an industry, identified as “best practice”. These shared dynamic capabilities are of course not unique, but for some processes, there exist a limited number of workable options, and if a firm did not use “best practices”, it could suffer. In addition, some processes, for instance financial statements, are often regulated, which limit the number of ways they can be done. These processes are still dynamic capabilities, but cannot be seen as unique and not a way for a firm to differentiate itself.

In modern service-based industry, the competence of the staff is often a very important differentiating factor. The tacit knowledge, i.e. the knowledge that comes from individual experiences and is hard to codify, is something that has become more and more important. If a company is able to make use of tacit knowledge and translate it to explicit knowledge, it will have an edge on its competitors (Nonaka and Takeuchi, 1995).

3.5.2.3 A matter of gaining, maintaining and using resources

Through RBV, researchers have gained some insights into the firm, and with these insights, they also gained new knowledge of the inner workings of a company. A firm needs to gain and maintain resources and then utilize them in an effective way to create a sustainable competitive advantage.

The operational platforms also influence the market positions, but not only through “position” as Teece (1997) puts it. The strategy may be based on unique resources, but the resources may also limit the number of potential customers. There is dynamism and a pressure to create a good “fit” between the market positions and the operational platforms. If there is a successful fit, it is often shown in the offering.

3.5.3 The offering

The final of the three elements is the offering, and as an element, cannot exist without the other two elements, namely the marketing positions and operational platforms. If the market positions are the interaction, and the operational platforms the tools, the offering is the exchange. The offering in this framework is more of a description of a firm’s business offer, since services and products are produced by the operating platform, and the marketing, sales and customers are handled by the marketing positions. Figure 6 illustrates the offering:
Normann argues for the need of a fit between the market segment, the product and the internal organization (1975). Kindström sees the offering as the interaction between the other two elements (2005). Normann (2001) discusses a complete offering, whereas the value proposition is gaining importance. Nevertheless, the complete offering may be followed, or supported, by lesser offerings aimed at different customers, or so-called teasers.

The offering is seen as an artifact, not a process like Normann (2004). The reason for making the offering an artifact is due to the fact that processes are a vital part of the capabilities of the firm and expressed in the operational platform, and the relationships with customers and partners are handled in the market positions.

In the sales process, or when working closely together with a customer in joint project, the details are still handled in the other two elements. In addition, there has to be an offering sometime in the process, i.e. one which stipulates the services, products and how much they cost.

This means that a business model, in theory, may consist of a limitless number of offerings, each with certain properties. However, having too many outstanding offerings at the same time could put a strain on the organization, and is therefore not practical.

The offering looks into several things, such as what the firm is offering, how the offer is made to the customers, how the product is shipped and what the firm wants in exchange for its products and/or services. Therefore, the extended offering consists of four items, the products and/or services offered, the revenue model, the way the product/services are distributed (distribution model) and how the offer is made (price model). A business model may have several different offers within itself, since the offers could differ between segments, or differences in the number of products or services of-
fered. However, an offering is usually valid for a certain time. After that time, the offering is discarded.

3.5.3.1 Products and services

The concept of the products and services should need no extensive explanation, since it is more or less, what the firm offers its customers. Notice that the amount of products and services offered may differ between different customers or segments. The products and services are connected to the operational platform.

The relationship between service and product business models for the software industry has been described as problematic due to the differences between the different positions. Nambisan (2001) identifies five key issues: intellectual property rights, product complementarity, returns from scale, abstracting knowledge and integrating technology and the connections with users.

Therefore, ideally the products/services offered should mirror the actual resources and capabilities of the firm, or at least a degree of it. This research has found that in technologically “heavy” firms, i.e. those with a firm engineering background, the offerings fit the resources and capabilities well. Nevertheless, the offerings also present a way for a firm to “test” the market with a new concept or product, one not yet fully developed or finished, to gauge the interest in the environment.

3.5.3.2 The revenue model

The revenue model is the way the firm is charging for the products and services offered. The revenue model is connected to the market positions. This means that the revenue model may also differ between firms, and in some cases, it might be a primary competitive scheme. This occurs often to services in connection with the Internet.

The revenue model may also differ between different offerings, even if the same product or services are offered. For example, Microsoft offers products with different revenue models to different segments of customers.

3.5.3.3 Distribution model

Traditionally, a product was usually delivered to the customers in some way that involved the act of physically moving the product from the site where it was produced to the place where it was sold. Commercial software is often distributed in a paper box, which contains the software product on some kind of media (today mostly DVD-ROM or CD-ROM) together with a manual or installation instructions. There are also products available for direct download from the Internet.

A service is usually produced at the same spot it is manufactured and does not need to be distributed. Of course, the consultant has to be “distributed” to the site where he “manufactures” the service, but in reality, the physical
location matters very little. Thus, distribution is more important for products than for services.

3.5.3.4 Offer model

The way the offering is being presented to the customers may differ between firms and even between different offerings of a single firm. Some rely on catalogs (IKEA for example), while others use a network of retailers or resellers, price lists (on paper or on the Internet), or various other ways.

The price model may be important when studying a company, since it can provide insights into how the firm addresses its customers. It may also be an indicator for how the company conceives its environment.

It becomes even more interesting if a firm has several offerings for different customer segments, and the way the offering reaches the customers. Studying the difference may give insight into how the firm perceives the different segments, but also how its different customers are used to having offerings presented to them.

3.5.3.5 Impacts on the other elements

An offering may affect both the market positions and the operational platform. In the latter case, it happens every time a customer accepts an offering. The actual offering disappears, but the agreement itself becomes a part of the resources in the firm. An offering may affect the market positions if it changes the way the environment views the firm. For example, a firm makes an offering into a new market, thus getting reaction from customers, competition and even suppliers and other potential partners. In fact, every new customer that accepts an offering is added to the market positions.

3.5.4 The new extended base model

To summarize, the new, extended business model consists of three (3) elements: market positions (M), operational platforms (P) and the offering (O). The market positions and the operational platforms, presented above, are interconnected to each other. The offering is more a reflection of both, as shown in Figure 7 below.
As is shown in Figure 7, there is an interaction between the market positions and the operational platforms. This interaction consists not only of needed adjustments to changes within and outside the organization, but also shows the actual exchange of concepts and actual physical objects. In this view, the content of the market positions are abstract rather than assets in the operational platforms.

3.5.5 A matter of control

One aspect of the business model that has not been extensively studied is the matter of control. Where can the firm exercise more control? Some may argue for the market positions, since part of the element is formed by strategy and vision. Nevertheless, at the same time the market positions consist of the current position of the firm in the environment, which means that the means of control that the firm may exert is limited.

Others may argue for the operational platform — that it still concerns the internal matters of a firm. On the other hand, resources may fade away, important staff quit, or worse. The control that the firm can wield in the operational platform is stronger than the market positions, but it is not perfect.

One important example: in most cases, perfect control, or at least close to perfect, can be applied to the offering. The offering, as an artifact, can be made without regard to anything (in theory); even if reality might dampen the number of ways, an offering can be made. Nevertheless, the second the offer is accepted, it becomes an agreement, one with certain capabilities that lessen the control of the firm (products and services offered at a certain price, delivery dates, etc.). This is illustrated in Figure 8 below.
3.5.6 Usage

The business model is primarily an analytical tool, which enables a researcher or a consultant to get a more holistic view of how a firm operates. The framework presented here is a generic framework, which points the researcher in the direction of the theories that inspired it, but the framework should be adapted to each individual researcher in order to be most effective.

By using the business model framework, the researcher may also find certain configurations or archetypes that are successful (or not) in a certain industry. By applying that knowledge, other firms may improve their business models in order to become more effective. A firm may also find deficiencies in the fit between the different elements that need to be addressed in order to make better use of resources or positions.
4 Methodology

This chapter is written by necessity in a more familiar voice. Since I am a former student of Humanities, the research methodology in itself has been somewhat of an interest for me, mostly because I like to ponder philosophy. This chapter is dedicated to my view of my conception of the world in general, but also the methodology used in this thesis.

4.1 Philosophical musings

In order to generate understanding in a third party about a person’s choice of research methodology I feel there is a need for some kind of philosophical discussion, which starts with that person’s own standpoint. However, unfortunately, from my point of view, there seems to been a need for researchers of philosophy of science to be orthodox, choosing one out of (preferably) two different positions. As an example, a researcher is said to be either positivist or hermeneutic (Andersson, 1979).

My point of view is not black or white, but consists of colors and nuances. I personally have no problems combining some positivistic thinking using some hermeneutic models – or the other way around. I am not alone in this, as for example Gummesson (1991) argues that a researcher may very well adopt different research philosophies (or paradigms). You could say I have seen models and methods in a pragmatic, and not philosophical, way. This is probably a product of my upbringing and experiences (see below). The closest theory I have found describing my conception of the world is the systems theory, with a little touch of an analytical perspective (Arbnor and Bjerke, 1994). This is because these two views in some ways are related, or even partly cover each other.

To explain why I see the world like this an analogy is helpful: I see the human race as part pack animal, part social being. Our bodies (and thereby our brains) are fashioned in a way that enables them to construct patterns in the environment in order to help us understand them.

The present patterns we see are partly a legacy from our biological past (as mammals), partly through our culture and language. From a positivistic point of view, everything is measurable; this is something I agree with. However, in order to understand what we have measured, we need to label it, which can only be done through culture and language – to which each individual has a special relationship.

Let us make a little exercise for the mind. Imagine we are transported into an alternative dimension in which everything is as we are used to, but some words have been exchanged. For example, the fruit we call orange is in this dimension called apple. This may sound trivial, but how does this change of labels affect the color orange? The exercise is a simple one, but the point is that our language actually colors our way to perceive the reality.
My view of reality creates a number of conditions:

1. Everything can be measured\(^{21}\)
2. Everything that is measured is labeled by language (and culture)
3. Humans look for patterns to understand their environment

In order to translate this into my research, you can say I am seeking patterns in the environment, but I am aware my interpretation is dependent on my grip of the language. The fact I am writing this thesis in English, and not in Swedish, which is my native language, may have some implications for the results.

However, one of the results of my point-of-view is that no researcher can enter into a research context empty (tabula rasa). The researcher is carrying all his or her history with experiences, knowledge and opinions. Into this mix, you also have to add the opinions of the research institution, supervisors and, as mentioned above, the use of language.

The research situation is even more complex, since the “personal backpack” with which the researcher brings to the research context also has to deal with the “personal backpacks” of all the other people involved in the situation. As if this was not enough, in the bottom of each “backpack” the whole culture (in my case the Western) is added together with some characteristics based from the legacy of human origin. All these things affect the research from day one.

To sum up my “position” in the positivistic and hermeneutic “axle”, I think I am somewhere towards the positivistic end, as illustrated in figure 9 below:

![Figure 9: My 'position' on the positivistic and hermeneutic axle.](image)

Regarding my conception of reality and research, I am confident I am not going to end up at any of the extremes.

### 4.2 Considerations for the research

My research in this thesis has thus far been centered on business models for companies involved in OSS. The fact that firms are trying to make a living from something most people consider free or gratis is of course one reason for the research, but the focus has been to find a framework in which these firms can be analyzed. Interesting enough, there has also been a movement within OSS where more companies are involved in the development of software, thus changing the process and methods previously used. This development has been dubbed OSS 2.0 (Fitzgerald, 2006).

My interest of this line of research is based on my former job as a manager

---

\(^{21}\) I really mean all. In the future we may very well be able to record and measure thoughts and feelings. I am not saying it is good or bad, but I really think it will happen.
of two firms using and developing OSS and the interest from my supervisors, Staffan Brege and Anna Öhrwall-Rönnbäck. There is a significant pragmatic aspect of the work, since the results are aimed towards contributing to a framework that could be used to analyze firms, primarily OSS-companies, but there is also a generic framework in mind. Since the research has been financed by Lundbergsstiftelsen, the pragmatic aspect has been encouraged.

During the work, my interest for the business model in general has increased. Together with other researchers in the institution, I have refined older models of the business model framework already in place. My next line of work, not to be presented here though, is to widen the model to other industries, for instance large heavy industries.

4.2.1 Educational background

My educational background from humanities should perhaps be indication that I would prefer a more qualitative approach since this is the method of choice for those disciplines, but that is far from the truth. Aside from working with hermeneutical interpretation of texts, comparison (also based of interpretation), I have also conducted pure applications of models. It has been a colorful journey so far.

Based on Alvesson and Sköldberg’s (1994) strategies of truth, I have tried to illustrate how the research journey of mine has progressed during my time as a student and researcher. As they describe it, the correspondent truth points towards external events, there are always external reasons for things to happen. If truth has meaning, it aims to find a deeper understanding for an event. The third truth strategy concerns application, and aspires to utilize the found truths. Figure 10 below illustrates that journey:

![Figure 10: The research journey.](image)

4.2.2 Pre-comprehension

One part of the positivistic view is the clean sheet researcher. In my case,
this is not the truth. So in this matter I agree with some of the statements of Glaser and Strauss (1967) regarding the grounded theory.\textsuperscript{22} I do believe that pre-comprehension may be a trigger to want to become a researcher, as in my case since I quit a well-paid job as a manager to become a PhD candidate. One of the most important reasons was that I shared the vision regarding science with one of my supervisors.

There is however, a risk of having experiences before entering a research situation, since the former knowledge may actually hinder you to see certain aspects of a context. I will discuss how I tried to negotiate this in the validation subchapter below. However, having some knowledge of an industry, you can take some shortcuts as a researcher, and with this knowledge actually propose interesting research questions (Eisenhardt, 1989).

### 4.2.3 Experiences

After my studies, I spent eight years working as a manager in small firms developing software, mostly based on OSS. The first company, Gladius e-commerce, developed software for e-business on secure systems from Digital Equipment, but as the work progressed, Linux and other OSS products came in focus. The company always had a service profile, so the transition from proprietary software to OSS was not a difficult decision. At that time, 1997, e-business was still in its early phases, so competent software for transactions on the Internet were still lacking.

My work was primarily as a regional manager, but quickly it turned out that all programmers became part of my responsibility. While still working with VMS as a primary operating system, the firm quickly embraced Linux and the free tools and databases connected to it.\textsuperscript{23}

Nevertheless, this was not my first contact with Linux. As a student, I was a member of the computer societies Lysator and Control-C at Linköping University. Lysator got Linux early, and in 1993, I was present when Linus Torvalds was the keynote speaker at the 20th anniversary of the society. In fact, the terminals that ran Linux often were the fastest terminals available, even if some of them were prone to crash quite often.

My next job, after Gladius, was at Cendio Systems, formerly Signum Support. Cendio was arguably the first company in Sweden that was created solely to support free software.\textsuperscript{24} My first job was to manage a new department for Linux systems. The income for the department was from a wide array of services, from remote management to educational courses. My job was not only to lead the department, but also to formalize new services in which the company could charge its customers.

\textsuperscript{22} I do not agree on the actual methodology of their theory, since in some applications it has a daunting tendency to produce trivial knowledge.

\textsuperscript{23} VMS is an operating system, which was developed by Digital Equipment Inc., but is owned and managed by Hewlett-Packard Inc. today.

\textsuperscript{24} At the time of the founding, 1992, the term Open-source Software has not been invented; all software that was non-commercial was called free software at that time.
Cendio is a special case because I worked in the company and was involved, implementing and influencing many of the changes which occurred during 1999 to 2005; as Cendio changed, so did my role. I became manager for Sales and Marketing, and then finally Chief Technology Officer. Cendio had then gone from a firm with about 50 employees, mostly consultants, to a software vendor with 10 employees. I was an active part of the business changes of the firm, and business models in general and revenue models in particular were important parts of my work.

My time at Cendio is the direct reason for me to conduct this research in order to get a better understanding regarding companies working with OSS. In short, I quit a job that was relatively well paid and interesting to become an overworked and underpaid PhD Candidate.

I consider most of those who worked at Cendio as my friends, some closer than others. When you work in a company that is forced to change its business in order to survive, you create a special bond with those involved. I still feel a lot for the company and the people working in it, and I guess that I always will, and one of the results I hope to contribute would be some advice for them based on this research.

4.2.4 Departmental research culture

With this in my “personal backpack”, I entered the world of science through a well-established department with a common set of research standards and methods. By using case studies and interviews and creating analytical models and tools to understand and validate the data, the “research paradigm” of the department is an abductive and iterative process, which fit my conception of the world (Alvesson and Sköldberg, 1994; Eisenhardt, 1989; Yin, 1994).

4.3 The study

The study was conducted in two steps, using multiple case studies in order to collect and correlate data (Eisenhardt, 1989; Yin, 1994). The study was originally planned to investigate American, Swedish and Chinese OSS companies, but due to time and resource constraints, the Chinese study was cancelled. For the Swedish study, a list from a web site with active companies working with Linux or OSS was used. Companies were selected that had more than five employees, and worked solely with products or services based on OSS. The plan was to find ten companies, but only five were contacted successfully. The American companies were chosen for their size and by contacts. Due to the limitations of time and money, interviews with five companies were planned, of which four were carried out.

25 We used the web site Linux.se in 2005 <http://www.linux.se/> but the old company list has been removed.
4.3.1 Stage one

The first round, conducted during the fall of 2005, consisted of nine companies with eleven interviews. The interviews were conducted in a semi-structured way using an interview guide; however, different versions were used for the initial Swedish and American studies (both exhibited as appendices). The reasons for the different guides were that it was thought that there would be less time with the large American corporations, and that some of the basic questions would be obtained through other channels. Nevertheless, the aim was also to get the respondents to talk freely and thus to avoid our interviewers missing any significant information. The guide evolved somewhat during this phase, but the main questions remained. All the interviews were conducted with at least two people, the author and either a colleague or an academic supervisor.

The interview guide was based on the business model made by Kindström (2003), and the first version was based on an earlier guide used by other researchers for the previous business model framework. The changes made were all due to the specific requirements for the OSS industry.

All the respondents were executives or company founders, and chosen by the author. In one case, Cendio, a former CEO of the company was interviewed in order to gain information of earlier events. Judging from the responses received, the chosen respondents in hindsight appear to be the right ones to answer our questions.

Aside from the simpler questions, one of the main aims of the interviews was to answer questions regarding a particular event. Normally we (the researchers) did not know about these events, but in one case (Cendio), the author of this thesis had full insight, since he was part of some of the most groundbreaking events. In the other cases, the respondents provided us with the information of events that may or may not have been known outside of the company. With the intention of gaining more information and insights, we used questions like “why” and “how” to make the respondent expand on the issues, all in accordance with the methods suggested by Yin (1994).

4.3.2 Stage two

The second stage began with a selection of companies from the first. Based on the result of the first series of interviews, we found that we had four companies that had similar types of offerings, development of a product based on OSS. The selection also incidentally consisted of four companies from the USA and Sweden, two large and two small; by coincidence, the study resulted into one large and one small company from each country. The companies, Codeweavers and Red Hat from the USA, and Cendio and MySQL from Sweden, were approached for a second round of semi-structured interviews in 2007.

The new interview guide was based directly on the new business model framework. By using this, we managed to create a structure that allowed for
63

a internal case analysis right from the start. The interviews in Sweden and the United Kingdom were conducted with the author’s supervisor, but the author alone conducted the American interviews. The interviewees were slightly different from the first round, with the exception of Red Hat. In the case of Cendio, there was continuous contact through meetings, telephone and e-mail throughout the process.

4.3.3 Validation

All the interviews were recorded using digital recorders. Most of that material was transcribed in order to make it easier to access and use. All of the recorded material was archived for later use. In addition to the interviews, access was granted internal documents as well as official material from the Internet and other sources. All data has been stored in its raw form together with the refined material. According to Yin (1994), the successful administration of all the data collected is one of the main advantages for using case studies.

The main reason for using case studies was because the research was in its earlier stages, which is why there was a need to explore the industry, using the business model framework, in order to gain insights into the workings of the firms within it. These insights then enabled us to further explore and interpret some of the results gained (Merriam, 1994), as described further in the analysis and conclusion chapters.

Another important matter is the number of possible companies to study. The vast majority of the Swedish companies consisted of one person, usually working as a consultant and not solely working with OSS. Since one of the concepts was to study differences (if there were any) between Swedish and American companies, one of the major ones was the size of the companies and their potential customers. As the FLOSS study clearly showed, the usage of OSS in Sweden, especially governmental institutions, was (in 2002) one of the lowest in Europe (Wichmann, 2002a). Moreover, since the main buyers of OSS products and services were governmental institutions, the lack of customers in Sweden may be the reason for the low number of active firms in Sweden.

Because of this, this work may be seen as relatively low in empirical material, but since one of the purposes was to create a theoretical framework to study these kinds of firms, the case materials may very well be enough.

In the view of the author and others (Pettigrew, 1997), most events and research are a process. Therefore, all the interviews and the results have been fed back to the companies. In some cases, this has generated internal discussions regarding customers or pricing issues. Moreover, to insure the quality of the material, project members, academic supervisors and colleagues have been part of the process of reading the case material and of creating the framework model and interview guides.

All the case companies presented in this case have had their case descrip-
tions sent to them for review. The material used for the analysis is the mate-
rial that has been approved by the companies.
5 Internal case analysis

All of the case companies in this thesis were developers of software products as their main source of income. In order to use the new extended business model framework, the key factors were grouped into the three elements. These factors were discerned partly from the theoretical framework, and partly from the empirical material. Since the framework was modified, the key factors could not be found directly in the material; however, it was collected through iterations of the material. Also, some subsequent interviews do ask questions directly based on the framework.

Table 3 will show the key factors in each of the three elements (for OSS firms):

Table 3: Key factors for business model analysis.

<table>
<thead>
<tr>
<th>Market positions</th>
<th>Operating platform</th>
<th>Offering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business idea, vision and goals</td>
<td>History and paths</td>
<td>Offer model</td>
</tr>
<tr>
<td>Competitors</td>
<td>Processes</td>
<td>Product/service mix</td>
</tr>
<tr>
<td>Competitive strategies</td>
<td>Experience and competencies</td>
<td>Distribution model</td>
</tr>
<tr>
<td>Partners</td>
<td>Intellectual property</td>
<td>Revenue model</td>
</tr>
<tr>
<td>Owners</td>
<td>Agreements</td>
<td></td>
</tr>
<tr>
<td>Community and customers</td>
<td>Economy</td>
<td></td>
</tr>
<tr>
<td>Brand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Promotion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Culture</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.1.1 Marketing positions factors

When studying the case companies, they were asked to describe several factors that influenced their market positions. These factors can be grouped into four different types.

The first type concerns the strategy of the company. Related to this are the business idea (also mission), vision and goals, which look at strategy building within the company.

The second type investigates the firm and its environment. The competitors identify the main competitors, and the competitive strategies show which way the company differentiates itself from its competitors. Of interest were also, who the partners and owners of the company were, and how the company interacted with its customers and the OSS community.

The third type concerns how the company markets itself. We look to the
The firm’s brands, both regarding customers, but also in the community. We also study how the company promotes itself and its products.

The final type regards the inner position of the company, or in other words its culture. This may be interesting if there is a gap between how the company considers itself in its environment and how it promotes itself.

### 5.1.2 Operating platform factors

The factors for the operating platform were grouped into three different types of factors. The first group of factors looks into how the firm operates based on its history and paths and some processes. These are usually dependent on each other.

The second group concerns the resources, such as intellectual property, competences and experiences and agreements.

The final and last item concerns the financial status of the company, its economy.

### 5.1.3 Offering factors

The final element of the business model consists of four factors: the offer model, which illustrates how the firm offers customers the products or services mix. There is also the distribution model, how the firm distributes products to the customers, and the revenue model, the way the company charges for its products and services.

### 5.1.4 Case selection

The presented cases were selected from a wider array of researched case firms based on three factors:

1. All the firms offer software products or packaged software to their customers
2. All the companies use proprietary or hybrid licensing schemes
3. All the firms are actively developing OSS – employees are being paid to develop OSS on the expense of the company

Important to note is that the factors above are not listed by the importance of the elements. In the analysis, some of the factors are bundled together, indicating that the findings or information have been scarce.

### 5.2 Cendio

Cendio was founded as Signum Support 1992 by alumni from Linköping University with the vision to provide support for OSS. In the beginning, the business was not based on free software (as it was called then), rather the low prices of the programming consultants. The

---

26 Unless stated otherwise, all the information comes from interviews with Per-Olof Moosberg, CEO, Cendio AB, March 10, 2006 and Inge Wallin, CTO and Co-founder, March 10, 2006. In addition, there was a continuous flow of information between the parties.

27 At the time (1992) the term Open Source Software did not exist, but for clarity I will use the modern term.
knowledge and interest of the local companies for OSS was low or nonexistent, which more or less forced Cendio to take whatever business that was available.

Later Cendio created a couple of products and the consultancy efforts were more targeted towards OSS, giving the firm a reputation as being an expert on OSS. During the late 1990s, Cendio received a capital injection and grew rapidly during the Internet boom years. During the same time, it expanded both its products and services.

After the Internet and telecom industry took a severe downturn in 2001, Cendio was forced to reduce its geographical presence and staff and reform its business, almost starting over. The new, slimmer organization then created a software product, which is proprietary, but the package the customers are using consists largely of OSS. Cendio sees OSS in itself as a vast resource that can be tapped in a wide variety of ways. In order to get useful products out of this vastness, a firm needs certain capabilities. These capabilities are usually in the form of specialized knowledge, in this case programming.

Seen from that perspective, one of the benefits of this is that the development department of Cendio does not need to be very large, which means less cost and better margins for its products. Nevertheless, selling a proprietary product made largely from OSS is not without controversy. Some of the open projects Cendio relies on may not go in the right direction, or have a very slow development. This is because development in OSS projects is based on a voluntary basis and the interest of the maintainer of the project.28

For Cendio, this subsequently means that it must put its own resources into the projects and contribute with skills, time and code. The pressure on the programmers to conform to deadlines and follow roadmaps to satisfy the customers (and the company) is in direct violation to the credo of OSS, which values code quality above deadlines. This puts a pressure of hiring more programmers to satisfy the needs, which increases the costs and lowers the margin.

5.2.1 Key marketing position factors

5.2.1.1 Business idea, vision and goals

Cendio develops software for server-based computing. The firm mixes OSS with its own proprietary code in order to create the best possible system for the customer.

Cendio’s vision is to become one of the leading software vendors for server-based computing software. In order to reach that goal, the company aims to work with resellers country by country. Cendio also works with a partnering

---

28 Maintainer is the term used for the copyright holder or project manager (usually promoted by the original copyright holder) of an OSS project. The maintainer is responsible for the development of the project and often controls the direction of the development.
strategy, trying to attract not only resellers, but also hardware and software vendors. The company has established itself as a good developer both in the OSS community and among proprietary software vendors.

Cendio is currently working to establish itself as a preferred software vendor for server-based computing for Swedish municipalities. In addition, Cendio is working towards achieving the same status at Swedish health services. The company also has resellers in Brazil and Malaysia, which promises great business opportunities.

5.2.1.2 Competitors

Cendio competes in a much-monopolized market. When Cendio started in 2002, there were two major vendors, Citrix and Tarantella, with Microsoft offering limited services. Today, Microsoft is the largest vendor, encompassing most of the market, while Citrix has been marginalized and pushed towards another end of the industry. Tarantella was purchased by Sun Microsystems, and its product is now part of the SunRay concept.³⁹

In addition to the main vendors, there exist hundreds of smaller, localized vendors offering different commercial solutions. On the Linux market, the Italian company NoMachine’s product NX is perhaps the most widely known.

There are also OSS competitors to Cendio. The most widely used is the Linux Terminal Server Project (LTSP), but there are several other projects devoted to server-based computing, and which base their projects on VNC or X.

A recent phenomenon is the web application business, also known as “Web 2.0”. This means that users get access to applications through the Internet using their web browsers. Products based on this technology are likely to be competitors in the near future. Large vendors in this industry are Citrix, Google, Yahoo, and Microsoft. This is the part of the IT industry that is widely believed to be the next “big thing”.

5.2.1.3 Competitive strategies

Cendio’s product, ThinLinc, is a mixed product from in-house, and along with OSS it enables Cendio to offer a wider functionality and adaptability than its competitors, seen from a development cost perspective. ThinLinc offers the customers the possibility to combine both UNIX and Windows applications on desktops.

The strategy is mainly a differentiation strategy, but with low-cost elements. The product is moderately priced compared to the main competitors with the same functionality, but for the customers the possibility to make use of cheaper hardware on the desktop and Linux applications may constitute a major cut in costs.

---

³⁹ SunRay consists of both hardware and software components.
5.2.1.4 Partners

Cendio uses a reseller strategy for its product, and its partners are among the largest in Sweden. This is partly due to the fact that the customers are too large for Cendio to handle when it comes to a full implementation phase, but also that the firm lacks the resources to fully support the customers during the lifecycle of the product. The resellers are usually systems integrators. The reselling partners abroad are usually smaller or in an early start up phase.

The company is a technology partner to the main Linux software vendors Red Hat and Novell, and a long-time partner to IBM and Fujitsu-Siemens Computers. The firm also works with vendors of thin clients such as Sun Microsystems, Neoware and IGEL. Aside from this, Cendio also enjoys a strategic relationship with Codeweavers (see more below).

Cendio is also gearing up to become an OEM partner to software vendors, where ThinLinc or ThinLinc components may become a part of the partners’ offering.

ThinLinc often gets very good reviews from testers and customers, and has attracted attention in other geographical markets. At this time, it has an established agent in Brazil and interested parties from other regions.

5.2.1.5 Owners

Cendio is privately owned; its employees and their families own some of the firm, but a private angel investor owns more than fifty percent. He is the chairperson of the board of directors, and firmly believes in the future of Cendio. He, along with the other owners, have reinvested in Cendio since 2002.

5.2.1.6 Community and customers

Since Cendio only offers a proprietary product, there is no OSS participation in it. However, Cendio is an active member of two OSS projects, rdesktop and TightVNC, where it is part of the main development team, and all development made by Cendio in these projects is fed back to the community.

The company’s main customers are Swedish municipalities. In order to create better communication, the company initiated the creation of a user group for ThinLinc. The Swedish market is dominated by Microsoft and Citrix, so OSS products have a limited impact; the firm therefore has hopes for markets in South America and Asia.

5.2.1.7 Brand

Cendio and ThinLinc are as of yet too small to get much attention in the OSS community. Cendio, and the former company name Signum Support, have made some impacts due to the age of the company and the fact that the company was active in the community at an early stage.

Cendio is also one of the main contributors to the rdesktop project, which
focuses on remote desktop protocol (RDP) used to present remote desktops (and applications) from Windows Servers to Linux machines. Cendio is one of two companies mentioned by the project web site, and they have three out of nine active developers.

Cendio has never spent much money on marketing; most of the promotion has been spread through “word-of-mouth”. During the dotcom era, Cendio managed to get some high impact contracts, like the Swedish-speaking clock system, that led to other telecom contracts.

The “new” Cendio, which is based around the product, has reached a position as a known vendor among Swedish municipalities.

5.2.1.8 Promotion

Cendio promotes its product mainly through direct and in-direct sales and through the company web site. The firm is also attending, and sometimes sponsoring, certain customer-related fairs and events, and hosts a user conference for its customers. Occasionally the company places ads in some specialized magazines in conjunction with a customer event or fair.

5.2.1.9 Culture

Cendio has had a turbulent past with expansions and layoffs. The present company is small and tightly knit, and all work together to achieve its goals. Openness and frankness is something that the CEO frequently communicates, even among customers and partners.

5.2.2 Key operating platform factors

5.2.2.1 History and paths

Cendio was founded as Signum Support in 1992. The first business concept was to provide support-free software, and was inspired by Cygnus Solutions (see more below under Red Hat). During the its early years, Cendio provided consultancy services, training and products based on OSS. The main customers for Cendio during this time were linked to the telecom industry. When the telecom industry recessed during 2001, this highly affected Cendio.

When creating a product or software package, Cendio has always used hybrid-licensing schemes. The scheme preferred for the early products was to bundle the software, which usually was a mix between OSS and in-house software, with hardware. When bundling an IT product with hardware, it is important to consider that the technology is rapidly changing, making the hardware obsolete in a very short time. This was often the case for Cendio when producing its firewall product; the hardware technology evolution forced Cendio to have very few machines in stock, which sometimes affected the delivery times negatively.

When Cendio turned software-only, it realized that the margin for pure software products was much higher than for the bundled versions, but at
the same time, that it took quite an effort to establish which systems the software could support.

5.2.2.2 Processes

In development, Cendio uses OSS tools on the desktop and server software such as CVS for version control and Bugzilla for bug or feature reporting. All development is done on Linux or UNIX systems.

In order to become a reselling partner to Cendio there is a certification process for the technician with education and tests developed and conducted by Cendio.

The reselling partners usually have a wide array of products and services to offer, which makes it hard for a small vendor like Cendio to get customers through them. Therefore, Cendio has developed a “test installation” process in which it offers the product and installation services directly to its end customers, as well as a test period. If the customer wants a full installation after the test period, they are turned over to one of the resellers.

5.2.2.3 Experience and competencies

Cendio’s staff has extensive experience with Linux and UNIX systems; they are therefore used to working within that infrastructure. ThinLinc has however forced Cendio to work with Microsoft and older Novell systems in order to integrate the end customers systems. These experiences, and the solutions they provided, are some of the reasons for the high regard Cendio has among its customers in Sweden. Some of the current developers have gained some positive reputation over the years, and some of the current developers are highly rated in their areas of expertise. For example, Cendio’s chief developer, Peter Åstrand, is part of the worldwide OSS project team for TightVNC.30

When the former CEO declared he wanted to step down Peo Moosberg was brought in. He came from the large systems integrator Alfaskop and brought decades of experience from the Swedish IT industry and under his leadership; the company was reformed into what it is today.

5.2.2.4 Intellectual property

ThinLinc is a software bundle consisting of both OSS and proprietary software, where Cendio retains the copyright of the proprietary parts. Cendio also owns the copyrights of its training material for the partner certification process.

Cendio has had some discussions regarding its licenses, since the pack-

30 TightVNC is one of the OSS projects that started from Virtual Network Computing (VNC) by AT&T and Olivetti. The original code was released as OSS in 2002. There are two main projects surrounding VNC. RealVNC is populated with some of the original team members of the VNC project that offer both an OSS and commercial product. RealVNC is more or less a continuation of the original project. TightVNC is focused to make VNC work on tighter bandwidths.
aged product consists of both open and closed software. However, the company maintains that there are no license issues, because of the nature of how the software applications communicate with each other.

5.2.2.5 Agreements

All paying customers have to agree to the ThinLinc End User License Agreement that states the rights of the customer and the relationship between the software in the bundle. Some customers have chosen to “lease” or “rent” the software, and they are required to sign a separate agreement in order to do so.

All resellers have to sign a partnership agreement and send and certify a required number of technicians in order to sell ThinLinc.

5.2.2.6 Economy

Cendio is not a profitable company today. However, its sales and income have been rising steadily since the launch of ThinLinc in 2002. Cendio works with the slimmest organization possible, making the need for new capital as low as possible. In addition, the owners have been very generous to reinvest in the company thus far.

5.2.3 Key offering factors

5.2.3.1 Offer model

The product is only offered through resellers, who distribute the offerings through price lists or written offers. One reseller offers the product through an online shop for downloading. Cendio also offers certification training directly on its website.

In Sweden, Cendio does offer test installations directly to end users, mostly in the form of governmental organizations like schools. If the customer likes the product, they are referred to one of Cendio’s resellers, who finish the affair.

5.2.3.2 Product/service mix

Cendio offers ThinLinc, with the addition of maintenance, support, certification and training connected with the product.

The product is usually delivered packaged with a maintenance agreement for one to three years. The support and training are offered separately.

5.2.3.3 Distribution model

The product is distributed through resellers, who either sell it through the Internet (downloads) or install it at the customers’ location.

5.2.3.4 Revenue model

Cendio offers ThinLinc as a proprietary product through resellers. There are different licensing forms, depending on the type of organization. The license is available per concurrent user, per seat (or computer) or per potential user.
The latter is mostly used in a School License Agreement. The license is offered with a maintenance agreement lasting from one to three years. In addition to the license, Cendio also offers support services for the products.

The price for the product is a mix between customer expectation, functionality and market pricing. The pricing of the competitors ranges from zero (LTSP) to very expensive (Citrix).

Cendio also offers certification training for resellers and customers; a certain number of certified technicians are required for resellers.

5.3 Codeweavers

Codeweavers came from a similar background as Cendio, even if Codeweavers was formed with a strategy to enhance existing OSS projects rather than create completely new products. In their case, the width of the market was not the problem, but rather the number of clients. Their position on the market, as experts of the so-called Wine project, which is a project that enables software built for Microsoft Windows to run under Linux, meant that the market was global, but the number of potential customers very few. The former customers had been large corporations like Corel, a software vendor with products like WordPerfect and Corel Draw, as well as Borland, a software vendor specializing in programming tools.

After the chaotic times following the Internet bust in 2001-2002, the Linux desktop market failed, which meant that the ISV service the company had hoped for failed to materialize.

Codeweavers had a global presence, but needed to expand its number of potential customers. The company’s new proprietary solution (made from OSS) called CrossOver opened up the market for both home users and companies. Codeweavers had full access to its main resource, the Wine project, through its maintainer/coordinator, who also became an employee.

The project has also been a resource when it comes to capabilities, since it has been used by Codeweavers to find and attract potential employees. Today, most of the active developers from the Wine project are employed by Codeweavers, and thus paid to develop both the OSS project and the proprietary software in connection to Wine. However, the proprietary parts, mostly installation and configuration utilities, are a very small part of the code base, approximately around 5% of the total code.

The downside of this is the fact that all programmers are spread across the globe, which means communication and managerial challenges. So far, there have been few problems, but the management is well aware that these problems may increase if the company grows.

There have been other attempts to create companies from the Wine project,

---

31 Unless stated otherwise, the information stems from interviews with Jeremy White, CEO and Founder, Codeweavers Inc., December 12, 2007 and Jon Parshall, COO Codeweavers, April 25, 2005 and December 12, 2007.
like the Cedega product from TransGaming Technologies (Cedega is a game platform for Linux), but since Codeweavers has been able to keep the main developers from the Wine projects hired and focused on other things, these projects have not affected the effectiveness of or the market for Codeweavers.

The biggest threat for Codeweavers now would be that Microsoft ported, i.e. made available, the Microsoft Office line products to Linux, since most of the business for Codeweavers are Microsoft Office Users on Linux. The recent agreement between Microsoft and Novell to cooperate regarding applications may very well be a severe threat to Codeweavers (Novell, 2005).

5.3.1 Key marketing position factors

5.3.1.1 Business idea, vision and goals

Codeweavers’ business idea is to develop software that makes it possible to use software that normally only runs Microsoft Windows on Linux and Mac OS X. The reason for this is to make migration between operating systems easier. The mission statement has changed from an initial focus on Linux to encompass Mac OS.

The company’s mission statement is, “Codeweavers’ mission is to transform Mac OS X and Linux into Windows-compatible operating systems” (Codeweavers, 2008).

Codeweavers is in the situation where its management team has to focus on day-to-day operations.

“We are very much focused on opportunities that will bring us revenue in the immediate sense. We are what we would call a “coin-operated” company and sometimes this hampers our ability to do long term strategic planning, because we are still not out of the mode of living from paycheck to paycheck.”

An important vision for the company is to be able to fulfill its original mission, which is to help ISVs to port their products from Windows to Linux and Mac OS. Interestingly enough, it is not the Linux market that has fathered these changes, but the Mac OS market.

Codeweavers has a planning window of about six months to a year. However, Codeweavers recently hired a Vice President of Sales. He is new to the company, but is on the verge of creating goals for the sales department, which is something new for the company.

A non-pronounced goal, but something that is in the minds of the executives, is to get a stable revenue stream that allows them to plan and widen their horizon from months to at least one year, or more.

5.3.1.2 Competitors

Aside from the Wine project itself, the competition for Codeweavers has risen since virtualization has become more popular in recent years. The products for virtualization offer most of the features present in CrossOver, but often to

---

a much higher price (and often lower performance). Some examples of competitors in this area are VMware and Parallels.

The most prominent commercial competitor that uses the Wine projects code base is TransGaming Inc., and their Cedega product. OSS community support for TransGaming is very low. After venture capitalists entered the firm, they decided to close the source. This decision made the community angry, and support questions coming from the company to the project web site went unanswered. Nevertheless, while Codeweavers focuses on office applications, TransGaming is focusing on taking PC games to the Macintosh and Linux operating systems.

Microsoft, the vendor producing the Windows operating system, which the Wine project is trying to emulate, is of course an important competitor. If Microsoft decides to fully take its applications to the Linux and Macintosh platforms, Codeweavers and its products would become obsolete. Even if this scenario has been considered more or less impossible for a number of years, Microsoft seems to have moved into this (feared) direction, if somewhat slowly. There is also a significant risk that Microsoft would try to prosecute Codeweavers for eventual IPR infringement. The agreement between Novell and Microsoft seems not to protect the Wine project (and thus Codeweavers) from that threat (Paul, 2007). However, Codeweavers owns legal research that has given it confidence that such litigation is very likely to be unsuccessful.

An interesting aspect of competitiveness is the aspect of the “free-riders”, mostly Linux users who predominantly download the free version, rather than using CrossOver. In a certain sense, they are considered the “worst” competitor for the company.

5.3.1.3 Competitive strategies

CrossOver is directly built on the OSS project Wine, but the commercial product also offers installation scripts and other integration features not present in the open project. The paying customer also gets access to professional support.

In comparison to other companies exploiting the Wine project in their products, such as for instance TransGaming, Codeweavers clearly has the community's support. However, the company (and companies like TransGaming) is also dependent on the maturity of the Wine project.
We are sort of caught in a sort of gray area at this point, Wine has got a lot better but a lot better means that it still only runs maybe 50% of all applications that are out there, and even those 50% will have a fair amount of warts on them. It is clear for us that if Wine were the sort of technology that would run 95% of all Windows-applications right out of the box that is a complete scene changer in terms of the technology landscape. [...] The question is how quickly we can get to that point. It is a very difficult technology to develop in; it requires a lot of time and effort. We just need more bodies, and that means more cash flow or a larger Open source community working on that technology.33

The recent success for Apple has resulted in an increase of Mac sales and a greater interest for companies to port their software from Windows to Mac OS. This has presented great opportunities for Codeweavers to retain its original plan with the company.

5.3.1.4 Partners

Codeweavers operate through resellers. Nevertheless, when it comes to identifying its most important partner, the company named the Wine community as the most important.

On the strategic side, the perceived threats from copyrights and patents make it hard for Codeweavers to obtain partnerships, in particular those with larger, more established firms such as IBM, Novell, HP, etc. Even if Codeweavers has shown there are no threats, through a fair amount of research regarding the legal threats, other organizations are still reluctant to work with the company.

In addition, even if the customers are welcome to visit the company’s international web shop, some corporate customers prefer to go through local or regional partners. The strategic partners are mostly connected through different OEM agreements.

5.3.1.5 Owners

The company is held privately, with Jeremy White holding the majority post, and the board of directors with a significant share as well. Some of the employees and families also own some shares.

Venture capitalists and other investors have approached them, but these contacts have never led to any deals. The managers clearly stated that they are not interested in spending the time and effort needed to create an agreement, and they feel that most investors are worried about patent or copyright issues in addition to the lack of their own intellectual property.

5.3.1.6 Community and customers

Even if the Wine project is free to download and use, the main customers for Codeweavers are normal users. The number of users buying the product was what saved the firm when it had to reform in 2002. However, on the Linux “side” of the community, the users are more likely to download the free version rather than pay for CrossOver, while Mac OS users seem to be more

likely to purchase the product, preferably in a box, than download the free version.

The community does contribute not only to the code, but also through bug reports and other kinds of feedback. In addition, Codeweavers employs the most skilled and dedicated developers directly from the Wine project. Since the company’s CTO is also the maintainer of the Wine-project, he personally reviews all code that comes to the project. Through this, he becomes aware of those coders who are doing good work, who are later approached by the company with an offer of employment.

However, the company feels that the Wine-community is not large enough, and this worries the management. If the company would get the chance to enlarge its business, the lack of developers would be a problem.

Codeweavers’ products and services are targeted towards customers that use Linux or Mac desktop systems, and who are interested in using Microsoft Windows applications on those platforms. The company offers its CrossOver product and services related to it.

There is also an opportunity for paying customers to suggest or direct Codeweavers towards the applications they want supported. Codeweavers’ original mission may return, when ISV’s want to port their applications from Windows to Mac OS. This market is on the rise.

Some of the better-known Codeweavers customers are Cisco, DreamWorks, Pixar, Disney and Google. Other customers include higher educational institutes, for example. In reality, Codeweavers’ customers are not concentrated in any specific industry; rather, they come from a diverse array of industries, markets and regions.

5.3.1.7 Brand

Codeweavers has a good reputation in the OSS community. The company also has employed most of its developers directly from the Wine project. The managers are true believers in OSS, and almost everything they produce they send to the Wine project.

Codeweavers in itself is a relatively weak brand name, but CrossOver is widely known. The firm has a partner strategy with resellers for the market outside of the United States, even if any customer is welcome to purchase the products directly from the company web site.

The new opportunity with Mac OS through a retail channel has increased the branding opportunities for the company through the boxes they are packaged in. In addition, Mac OS users seem to be prone to use the support services, thus adding more contacts with Codeweavers.

If the ISV porting market takes off, its visibility for Codeweavers will be considered low, since the fronting company rarely would like to “pollute” its own brand by adding others.
5.3.1.8 Promotion
Codeweavers promotes its products and services mainly through public and media relationships, in order to get as many articles, reviews and other media coverage. The company has recognized that in the Mac OS, customers are more likely to listen to “media movers”; therefore, Codeweavers has actively targeted influential reporters and bloggers and sent them free copies of the product for testing.

In addition, the company promotes the product through its web site, direct and in-direct sales. This year it hired a sales manager who is conducting direct sales, mainly through retail for the Mac OS product.

The company also enjoys promotion through the Wine project, for which the company regularly hosts conferences. Codeweavers has attended other fairs and events, but has lately withdrawn from the Linux World trade show in favor of the MacWorld show.

If the company becomes the primary application porter from Windows to Mac OS (and Linux), its branding opportunities will diminish; that, however, is not considered an issue for the company.

5.3.1.9 Culture
Most of Codeweavers developers work from their homes, making a “traditional” company culture difficult. The main form of communication is via chat, so the chat clients are always online and the company uses IP phones.\(^{34}\)

The different time zones are still an obstacle for this kind of communication and in certain cases, such as during product releases, the coordination within the company is hampered due to the disparity of locations.

However, the time zones may also have their benefits. At one time, for example, a severe bug was discovered one day before the release, but since the problem could be transferred throughout the different time zones a solution was found and implemented just hours before the official release.

The company’s employees are bonded together in their belief for OSS; this goes from all the way at the top with Jeremy White and Jon Parshall throughout the entire company. This and the sense of openness in the firm, including candid reporting on the financial situation, are the most important factors for why the developers have stayed loyal to the company, despite shortages of funds and even delayed salaries.

5.3.2 Key operating platform factors
5.3.2.1 History and paths
Codeweavers was founded in 1996, starting out as a programming consul-

\(^{34}\) An IP phone makes use of the Internet instead of normal telecommunication cables, and thus international calls costs the same as local calls (if there is any cost involved at all). One example of this technology is Skype.
tancy firm. The mission of the Company changed in 1998 to one of helping ISVs to port their software from Windows to Linux. The company received venture capital in 2000. The business model in 2000 was to make use of the expected growth of the Linux desktop market.

One of its biggest customers during the Wine project period was the Corel Corporation. Corel wanted to port its office products to Linux desktops; when the contract was withdrawn in 2001 (due to Corel’s sale), the firm decided to create a product out of the Wine project.

The company was virtually reborn in 2002 as a product company with the launch of the CrossOver Office product. From 2002 to 2007, the company struggled with the economy and flat or little growth in sales. In 2007, the Mac OS market opened when Apple switched to Intel processor architecture, creating great opportunities for the firm.

The founders and the management team are not fond of proprietary or hybrid revenue models. They think of their current revenue model as unavoidable.

However, the recent success for the product on the Mac OS market has forced the company to adopt an even more traditional consumer-oriented revenue model through retail boxes, a business that never took off for the Linux market.

Nevertheless, the most favored outcome for the firm would be to return to the original business mission, i.e. a service-oriented consultancy firm focusing on porting ISV software from Windows to Mac OS X and/or Linux.

5.3.2.2 Processes
Codeweavers and its products are closely tied to those of the Wine-project; Codeweavers has contributed about 40% of the total actual code of the Wine Project. The company has employed ten developers out of a community consisting of roughly 40 active developers.

Codeweavers uses OSS development tools and has a private code repository, using the version control system Git for its internal work. There are two separate repository “trees”, one for Free Wine and the other for CrossOver, but all code is submitted to the project before it is implemented into the products. After the code has been submitted, the software is tested before it is shipped.

5.3.2.3 Experience and competencies
Codeweavers’ development staff are hired based on their programming proficiency from the Wine project. When a new developer starts to make good contributions, the project maintainer, Alexandre Julliard, who is also the CTO of Codeweavers, notifies the management team. In addition, most of the developers are fresh from college or university education. However, the management team, consisting of Jeremy White, Jonathan Parshall and Alexandre Julliard, also are experienced in other fields.
5.3.2.4 Intellectual property

Even if Codeweavers does not retain the copyright on the Wine project, the company has significant influence since the maintainer of the project is part of the company management team. However, not every bit of the code produced by Codeweavers will be let into the OSS project.\(^3\)

In addition to the CrossOver products, Codeweavers has also developed a virtualization program called “Bottles”, which is essentially a GUI, and other small scripts or programs that are proprietary.

5.3.2.5 Agreements

The Wine project is licensed under LGPL (Lesser General Public License), which means that all customers are bound to those terms. Moreover, Codeweavers has an “End User License” for its customers.

In addition, the company has made reselling agreements which are somewhat standardized. With some OEM customers or partners, the company has signed non-disclosure agreements (NDA).

5.3.2.6 Economy

Codeweavers has historically not been a profitable company. Its goal for 2007 was to be able to pay all the salaries, which it did. From a positive angle, the company generates revenue from a broad range of sources, and thus is not dependent on a certain entity or market. The need is to gain more volume. The exchange rate the last year has also hurt Codeweavers, not only because of its sales abroad, but also due to salaries. One of the things the company would like to do, if the economy would allow it, is to hire more developers from the Wine project.

However, the new focus on the market for Apple computers seems already to provide them with a far more stable income and a profit. It seems like the Macintosh market may very well be the turning point the company has been waiting for.

The revenue for 2007 came from consumers (50%), corporate customers (about 20%), professional services (around 20%) and retail (10%); the latter being the most expanding part of the business.

5.3.3 Key offering factors

5.3.3.1 Offer model

All products and prices are displayed in US dollars and are purchasable through Codeweavers web site. Larger customers are offered a sales contract and adapted pricing. The resellers, in turn, may offer the products in different ways and or in local currency.

In the last year, Codeweavers has been adding a retail channel, mainly for

\(^3\) Alexandre Julliard has occasionally rejected code written by Codeweavers for Wine. This is often due to that some may be more or less dirty workarounds to enable certain software to function, but it is not something that would benefit the OSS project.
its Mac OS version. The product is packaged into boxes and shipped through distributors to shop in the United States. The company has plans to expand this offering to other regions as well.

5.3.3.2 Product/service mix

Codeweavers offers four different versions of the CrossOver product: CrossOver Mac, CrossOver Linux Standard, CrossOver Linux Pro and CrossOver Server. All products are derived from the Wine project, with additions, such as scripts and graphical user interfaces (GUI), to enhance the handling and installations, which are kept proprietary. As a part of the offering, Codeweavers guarantees that the supported applications listed will be functional, regardless of the status of Free Wine. To some degree, other support services are also included in the product pricing.

In addition to the product offering, Codeweavers also offers consultancy services for Wine, end user support for specific applications, porting services (making a Windows application to run on Wine) and software testing and evaluation. These services are offered through an hourly rate.

5.3.3.3 Revenue model

Codeweavers offers its products with a proprietary license, which in essence is a subscription. In addition, its services (aside from the support services included to the product) are charged at an hourly rate. The pricing of the product is set to be competitive compared to its competitors or substitutions.

The relationship between different markets has undergone significant changes; for example, the retail or direct customer sales is now 50/50 between Linux and Mac OS. However, the retail growth is almost entirely on the Mac OS side.

Regarding corporate customers, the relation is 60/40 in favor of Linux, while professional services are nearly 100% based on Linux.

5.3.3.4 Distribution model

5.3.3.4.1 Linux

Codeweavers Linux products are mainly distributed by downloads by the Internet. A media kit with a serial code can be made available for an additional cost. The company has a two-tier distribution model for Linux customers, working directly with resellers. However, in Germany and United Kingdom a three-tier model with distributors has been established.

5.3.3.4.2 Mac OS

The newer Mac OS version has a different mode of distribution. In addition to the normal downloadable version, the Mac OS product is also available as a boxed retail version, which is distributed through retail channels.

In Japan, the distributor NetJapan has an exclusive deal to distribute the product for the Japanese market.

In United States, the company goes through an aggregator called Red Frog
and the main distributor Navarre. Through Navarre, the product is shipped through retail chains like Best Buy, CompUSA and the Apple Stores. Moreover, the educational distributor Douglas Stewart has moved the product to campus bookstores.

One of the company's goals for 2008 is to find suitable distributors in Europe.

5.4 MySQL\textsuperscript{36}

The company MySQL AB is a Swedish firm founded in 1995. It is closely connected to the database product MySQL. The founders, David Axmark, Allan Larsson and Michael “Monty” Widenius, had earlier constellations when they worked as consultants, but after their success with the database, the firm was molded around the product. The name MySQL is derived from Monty's daughter My. The dolphin, which is part of the company logo, is named Sakila, which is a feminine SiSwati word. As of 2008, Sun Microsystems Inc. has acquired MySQL AB.

MySQL became popular in the late 1990s, becoming a part of the so-called “LAMP stack” (Linux, Apache, MySQL and PHP) which became a standard way to set up a Linux-based web server. The concept of the stack is still widely used, but nowadays PostgreSQL (making the “LAPP stack”) sometimes replaces the MySQL server.

Being popular is not always pleasant. MySQL has often been subjected to quite extensive criticism. Most issues regarding the functionality of the database, however, were remedied with the release of version five in 2005. MySQL AB became part of a business group after venture capital entered the firm. Today, MySQL is privately held with its headquarters located in Uppsala, Sweden, and subsidiaries in Finland, France, Germany, Ireland, Japan, Russia and the USA.

5.4.1 Key marketing position factors

5.4.1.1 Business idea, visions and goals

The main business idea for MySQL is to provide a database to those who build online services and applications.

The vision of the company is that the world is going online, and thereby there is a need for a new type of data management. Regarding the culture of the firm, MySQL made up a list of its values:

- The best and the most used database in the world
- Available and affordable for all
- Easy to use
- Continuously improved while remaining fast and safe

\textsuperscript{36} All information, unless stated otherwise, is from interviews with Mårten Mickos, CEO, MySQL AB, October 28, 2005, and David Axmark, Vice President and Founder, MySQL AB, February 28, 2006 and October 14, 2007.
MySQL AB and the people of MySQL AB:
- Subscribe to the Open Source philosophy
- Aim to be good citizens
- Prefer partners that share our values and mindset
- Answer email and give support
- Are a virtual company, networking with others (MySQL, 2007a)

5.4.1.2 Competitors

MySQL has three different kinds of competitors: the market leaders, firm-based OSS products and the OSS products themselves.

Among the market-leading databases, Oracle is the most vocal competitor. Oracle is the dominant world leader, and has in recent years worked closer with Linux, today even offering its own brand of Red Hat Enterprise Linux. In 2005, Oracle also bought the Finnish firm InnoDB whose product was distributed with MySQL (Oracle, 2005). InnoDB is still distributed with the enterprise product for MySQL, however.

MySQL has signed partnership agreements with the other market leaders, IBM and Microsoft.

There are other commercial OSS databases as well, and the most recent competitor who is also directly aimed towards Oracle and MySQL is EnterpriseDB, which is based on the PostgreSQL database (Danielsson, 2007). The company has aligned itself against Oracle, MySQL and PostgreSQL by offering specific solutions aimed towards those three (EnterpriseDB, 2007).

The main legal difference between the companies is as MySQL retains the original copyright for its product, EnterpriseDB does not. Nevertheless, since PostgreSQL is licensed under the BSD license, EnterpriseDB is free to make a proprietary license for its version.

Among competing OSS projects, PostgreSQL is a very popular and powerful OSS database, licensed under the BSD license. This license allows any user to make free use of the product, even repackage it and charge a proprietary license for it, making it a very wide spread database engine.

In the terms of actual users, PostgreSQL does not boast the same number of users as MySQL, but given the freedom of its license, it is very hard to track down its actual usage. In the European FLOSS report, the numbers were quite clear with 71% of the planned or installed base using MySQL, while PostgreSQL had 14% (Wichmann, 2002a).

Even if there is other OSS database engine software, PostgreSQL and MySQL are by far the most popular today.

5.4.1.3 Competitive strategies

The company's main competitive strategy is to have a product perfectly designed for the online world, sold under a disruptive business model. Moreover, its product is open source; they have a vast number of users and an active network of developers. This keeps the development cost low for the
company enabling a low-cost, market-of-scale strategy.

**5.4.1.4 Partners**

MySQL has a wide array of partners. Most of these are connected to the OEM business, but there are also third part vendors who add functionality to the software suite (MySQL, 2007d).

MySQL touts some of the world largest computer companies as its partners, such as Dell, HP and IBM, who not only integrate the software into their products, but also offer training and consulting for these products. However, the training and consulting they offer primarily regard their own products, and not MySQL alone, which can interfere with MySQL’s business.

**5.4.1.5 Owners**

Private investors, angel investors and venture capitalists hold MySQL. The company has had significant investments from highly acclaimed venture capitalists. There are also rumors about an initial public offering not too far in the future (Aslett, 2007).

However, during the work with this thesis, MySQL has been acquired by Sun Microsystems (MySQL, 2008; Sun, 2008)

**5.4.1.6 Community and customers**

MySQL is among the most popular OSS in the world. (MySQL, 2007b) The firm has registered more than 11 million product implementations (MySQL, 2007c). The number of downloads is believed to be ten times or more than the installations, so it is clear that the number of downloads does not have to signify an actual installation, but there is always a problem measuring the impact of OSS products.

The community does serve another purpose as well. Today most of MySQL’s developers are recruited directly from the community. In addition, in accordance with the OSS credo, MySQL uses the actual work of the applicants to review the developers.

Since the company was founded, the number of corporate customers has been rising. Still the number of paying customers is very low compared to the number of downloads. As far as the company can see, about one out of a thousand users or more are paying customers. Nevertheless, the widespread reputation in the OSS community also enabled MYSQL to attract internationally acclaimed investors to invest large sums in the company.

One of the problems, as seen from a marketing perspective, is that the main business for MySQL, its OEM customers who embed MySQL within their own products, are not always interested in advertising this fact

**5.4.1.7 Brand**

MySQL has a very serious commitment to the OSS community; for example, the company has a vice president, Kaj Arnö, assigned directly to handle such contacts. The co-founder and vice-president, David Axmark, is also highly
involved with the community.

What the company gains from being an OSS firm, from the developers, are all the third part tools, manuals and other miscellaneous products needed to create a fully-fledged offer.

Even so, MySQL has had its share of mishaps and misunderstandings in its relationship with the community. Some business decisions can create many angry voices, such as asking users to register (albeit on the commercial web site). Regarding the business agreement with SCO, in the latter case the outcry seems to have been blown up by the media (Jones, 2005). Another, more recent, disapproval had to do with the decision of removing the tar ball (a way to distribute binary code) for the community server and only make the enterprise version available for paying customers (Clarke, 2007).

Nevertheless, Mr. Mickos himself has no illusions about how to treat the community:

I see it like starting a nightclub or a discotheque. You have to give free tickets to the VIP's and let all the celebrities in. They attract the normal people and the normal people are willing to stand in a queue and pay in order to get access to the same things the other's don't have to queue or pay for. However, it works. In addition, we have the same situation; we have to let the open source community in for free and ahead of the queue, while our regular customers have to queue and then pay for entrance. Both groups are content.37

MySQL is a firm and is trying to be profitable, but at the same time, it realizes the value the community has for its business.

The firm branding for MySQL was initially kept low, mainly due to financial issues. Nevertheless, when Mr. Mickos entered as CEO and the firm got investors, the branding of the company was been stepped up, and an interesting side effect occurred. When the corporate branding increased through ads and media articles, downloads of the OSS program also increased. This indicates that there is a reciprocal effect for branding efforts, at least when one, such as MySQL, uses the same brand for the product, company and OSS project.

Since MySQL retains the copyright of the core product, it has and is able to offer proprietary licenses in addition to the OSS license it uses (currently GPL 2.0). Formerly, MySQL offered companies that wanted to include the product inside its products a proprietary license. This licensing scheme is referred as “dual licensing” throughout this thesis.

Today MySQL also offers a subscription package in addition to the proprietary license – which gives a company who uses the database professional support and services.

As one of the leading OSS product companies, the firm automatically attracts the interest of the media, which helps the process.

37 Interview, Mårten Mickos, CEO, MySQL AB, October 28, 2005. The quote has been translated from Swedish.
5.4.1.8 Promotion
MySQL is mostly promoted through the OSS project and the web site it maintains (which also promotes the product), but also through different marketing campaigns, including indirect sales and through fairs and events.

5.4.1.9 Culture
The company culture of MySQL is unique; since most of the development is made from home, there is of course a problem establishing some sort of culture for the developers. However, there seems to be no need to create a certain policy from the management to create a certain culture; since the company is an integral part of the OSS community, openness is always promoted.

5.4.2 Key operating platform factors
5.4.2.1 History and paths
MySQL was created from a consulting firm owned by David Axmark and Michael Widenius. It was in this firm that the database engine was initially developed.

The product has historically always had a proprietary license attached to it, especially the early Windows versions. MySQL was among the first to create a dual licensing scheme together with the GPL.

Even if the decision to make MySQL OSS was made relatively early, the firm has always been focused on becoming profitable. This is the main reason for experimenting with different licensing schemes for its product.

MySQL also realized that it was hard to recruit OSS developers to work on the database engine since it is very complex. It is therefore quite hard for a developer, working on a hobby basis, to absorb the knowledge he or she needs in order to be able to develop it. Very few have managed the effort, and they have most been hired to work for the company.

5.4.2.2 Processes
The code complexity of the database engine makes it hard for a new developer to start developing for it; hence, most of the development of the core product is done in-house. This way, the firm also has a definite grasp on the copyright of the product, which enables it to use a hybrid-licensing scheme. The development process is similar to that of OSS in general, using generalized tools accessible by the Internet, such as software depositories like CVS and a custom-made bug-reporting tool made on PHP.

Since most developers work from their homes, most of them have their own depositories (and other systems needed for development) at home; when turning in their code, they synchronize the versions with a central server.

Other prominent processes are connected to marketing and PR, which is conducted mostly through the web site, seminars, fairs and similar events. The subsidiaries of MySQL have not been founded based on marketing. In-
stead, they are created when there are a certain number of developers working from a country in order to ease salaries and taxation.

MySQL AB is not a public company, but it still faces rules and regulations regarding its financial status in Sweden. There were ambitions to make an IPO in the near future, but since then the company has been acquired by Sun Microsystems.

5.4.2.3 Experience and competencies
The founders realized the potential for the firm, so they decided to recruit an experienced managing director. When Mårten Mickos was asked, he initially turned the offer down, but since the founders persevered, he finally accepted. With his assistance, MySQL managed to attract large venture capital investors, who also provided help with the management. The management team and investors of MySQL are well experienced in the matters of the software and database industry and OSS.

The founders are regarded as competent programmers, and many of the developers are employed based on their achievement in the community. The company has been able to recruit many very competent software developers during the years.

5.4.2.4 Intellectual property
MySQL retains the copyright for the database engine and the trademark and logotype for MySQL. The copyright comes from the fact that the founders created the software, and that it conducts the development of the core software product in-house.

5.4.2.5 Agreements
MySQL offers partnership agreements for technological (third-party software products), consulting and training. In addition to this, it offers OEM agreements for commercial firms using the database engine embedded in its own products.

5.4.2.6 Economy
MySQL is not profitable yet, but it does have a positive cash flow. The company is also working to improve its presence and brand name in the market, and is investing on development for product improvement. In addition, the company is gearing up for an IPO, getting the organization and systems in place.

5.4.3 Key offering factors
5.4.3.1 Price model
Most of the prices for the products and services are presented through the company web site, but a number of customers, for example OEM customers, are handled through the sales department.
5.4.3.2 Product/service mix

MySQL offers its product with different licenses (see Revenue models below). The commercial offers consist of proprietary licenses (for OEM customers), software packages and services (installation services, software upgrades and support). In some countries, additional consulting services and training are available; these services are mostly offered by partners, however, and thus will not be further discussed here.

5.4.3.3 Revenue models

The MySQL dual licensing scheme consists of an OSS version (free-of-charge using OSS licensing), an OEM version with a proprietary license and an Enterprise version, which is delivered as a subscription. The subscription fee is annual and has four levels (Basic, Silver, Gold and Platinum). Both the OEM and Enterprise versions can be delivered with closed third-party software and services. This licensing scheme is considered dual because of the proprietary license offered to OEM customers.

5.4.3.4 Distribution model

All MySQL products are distributed through downloads from the Internet.

5.5 Red Hat

Red Hat was the first OSS firm to make an IPO (Initial Public Offering) in 1999, and it was the most successful introduction on the stock market at that time. The public interest was huge and the stock soared for quite some time.

Red Hat management realized that their original business concept, retail selling of products distributed via CD-ROM, was deteriorating fast as the Internet connections made large downloads easier for everyone. They decided that they needed to change their focus from retail customers to business customers. Red Hat had installations at large corporations, but not in significant, business-critical systems. One of the problems was the fast prototyping cycle, which made it impossible to certify its products, i.e. ensure quality, robustness and functioning of the system, as well as the distribution on large enterprise systems. Another aspect of certification is the proper interoperability with other systems; the lack of a defined hardware and software ecosystem made this transition very challenging.

When Red Hat decided to change the development cycle of its Linux distribution, it defined a release process that effectively shut out customers and community developers once Red Hat selected its baseline open source software packages. The new product was to be more closed in order to control its

---

38 When not stated otherwise, facts have been collected through interviews with Michael Tiemann, Vice President Open Source Affairs, Red Hat Inc., October 25, 2005 and December 10, 2007.
content. However, by doing this, they very effectively shut the important users and developers external to Red Hat out.

The company was on the verge of losing one of its most important assets, the open innovation (and bug testing) made by customers and users. In order to change this, it created the Fedora distribution – which basically is unsupported, but has faster cycles and makes it easier for developers to test new concepts. The enterprise edition is now significantly improved by the Fedora distribution.

One of the most important concepts of OSS is that the users are more important than merely unpaid developers: they are developers solving problems that matter to them. They are very hard to manage using traditional methods, but their contributions, when aggregated across hundreds of thousands of customers and millions of open source developers, provide an important competitive advantage over proprietary competitors.

5.5.1 Key marketing position factors

5.5.1.1 Business idea, vision and goals

Red Hat considers itself as the world-leading Open Source and Linux vendor. The company offers its enterprise technologies through annual or longer subscriptions. Its product offering consists of an all-inclusive package of software, problem resolution, continuous corrections and certified compatibility with certain hardware and software applications (Red Hat, 2007d).

In addition to this, Red Hat also supports a free Linux distribution, Fedora, which in some sense could be regarded as a development platform for the enterprise product. The company firmly believes that OSS provides the most innovative, competitive and sustainable basis for selling those subscriptions.

In terms of the mission statement, we do have a mission statement that something likes something: “To build some quality product at the lowest possible cost supported by an active community of users and serviced by brand”. It puts together the concept of high value and low cost, community and brand.39

Red Hat’s vision is “To be the defining technology company of the 21st century” (Red Hat, 2007a).

Goals are set primarily for the Sales department. “Sales is the lifeblood of our growth strategy”, says Michael Tiemann. In order to develop its market strategies, goals are set for the diverse types of sales, like inside direct sales to the third-party reseller channel. The goals may look internally competitive, for example horizontal versus vertical and geographical versus centralized, but with the diversity of the sales systems, it is to be expected.

From the marketing standpoint, the strategy has been to be disciplined when it comes to messaging, for example the participation in press releases. The main goal is to develop instead of “consume” the brand.

5.5.1.2 Competitors

The main competitor that Red Hat identifies at this time is Microsoft, which is considered a challenge since it prevented Red Hat with access to the desktop market, mainly through proprietary formats, for example in media.

What I see as the future competition is Red Hat and Google. Google is an OSS company that sells proprietary services. Red Hat is an OSS company that sells open services. [...] We are in a world now where Open source have actually become the endgame. Google have said, “The way we will play the endgame is to exploit Open source technologies and build proprietary services that run on our massive infrastructure. [...] Then there is the Red Hat play, which is "we are going to build a platform that the customer can maintain".40

Three years ago, Red Hat identified Sun Microsystems as a great threat. The Solaris UNIX operating system has the most common operating system for industries with high demands on computing. When Sun announced that it was going to make an OSS version of Solaris called OpenSolaris, this was seen as a significant threat to Red Hat (Shankland, 2005). However, this action is thought to have been made too late to hurt Red Hat.

Most of the UNIX operating systems should be considered as threats to Red Hat, but the main vendors like IBM and HP have started to downplay their own UNIX brands to the benefit of Red Hat, making Sun Microsystems as the only remaining real UNIX threat.

Among the Linux vendors, only Novell, which purchased SuSE in 2003, is the best-positioned competitor to Red Hat. SuSE Linux was the second-largest Linux distribution, with a strong grip on the European market, mainly Germany, before it was purchased. Novell offers similar products as Red Hat, but the product is complemented with proprietary products from Novell’s earlier product portfolio (Novell, 2007).

However, it is hard to discern the real revenue from Novell due to its current reporting practices, but it seems clear that Novell is not even close to Red Hat in sales when it comes to revenue based on OSS products. Novell’s sales may also have been inflated by the pre-paid coupons bought by Microsoft as part of their agreement (Microsoft, 2006; Evers and Shankland, 2006).

Another strong competitor is Microsoft. Microsoft is one the largest software vendors in the world with strong positions on the operating systems market.41 Microsoft has traditionally been very hostile towards Linux, but surprised many when they signed an agreement with Red Hat’s competitor Novell in 2006 (Novell, 2006).

In the application server or middleware market, Red Hat premier competi-

---

40 Interview, Michael Tiemann, Vice President Open Source Affairs, Red Hat Inc., December 10, 2007.

41 The actual market share of an operating system is hard to measure. OneStat offers one method, where the operating systems are measured through what the web browser reports at certain web sites (OneStat, 2007).
tors are some of their closest partners, such as IBM and Sun, but also the former partner Oracle. These actors have other offerings in several other markets, so while one division is a good partner, another can be a fierce competitor. However, the biggest specialized competitor is BEA Systems and their Weblogic solutions, which is the market leader for Java-based application servers. In addition, Microsoft is also a competitor with its .NET products. However, the Microsoft product can only be used on Microsoft operating systems, while most others can be used on several hardware and software platforms.

5.5.1.3 Competitive strategies

Red Hat is proud to present its OSS heritage and maintains that the performance and cost effectiveness of its enterprise solution based on OSS creates the most value for its customers (Red Hat, 2007d). While Red Hat’s enterprise Linux products are sold through subscriptions, the company still maintains a free beta testing Linux distribution called Fedora. Fedora is used as a prototyping and development platform, where Red Hat can test technology strategies for future generations of the enterprise product range (Fedora, 2007).

Even if Red Hat, compared to other Linux distributions, is considered highly priced, Red Hat does have “cost effectiveness” as one of its leading phrases. Comparing the Red Hat enterprise product with other comparable products, such as UNIX and Windows Server, Red Hat is indeed priced in the lower range. Looking at the total offering with the services offered, but without a software license on top, Red Hat does have a low cost alternative to that kind of competition.

In reality, what Red Hat aims for is a differentiation strategy. Close to identical Linux distributions, such as CentOS and Oracle Unbreakable Linux, does not offer the same services as Red Hat does. In the application server market, where Red Hat offers the JBoss solutions, it is however, priced lower compared to the big commercial competitors, such as IBM, Sun and BEA Systems.

5.5.1.4 Partners

Red Hat cooperates with some of the world’s major enterprise IT vendors like IBM, HP, Sun, Intel, AMD, Oracle, and Dell. The partners range from: hardware partners, which provide drivers and such for the products; OEM partners, which incorporate parts of Red Hat products into the partner product; training partners, which offer Red Hat courses and certifications; and resel-

---

42 Even if the actual product is sold through a license look-alike “subscription”, all the individual files remain open and downloadable from a vast array of servers worldwide. What the subscription covers, which is not included in the files, is the patch and upgrade system, as well as support and installation scripts, which makes the implementation of the product easier.
lers, who resell Red Hat products and services.

5.5.1.5 Owners

Red Hat is a publicly held company with several large investment firms which hold large parts of its shares (Yahoo, 2008).

The eventual influence these kinds of shareholders have on a company like Red Hat is of course considered inside information. Privately held companies, with a few active owners, are those that have a significantly larger impact on a firm’s strategy, and they obviously differ between publicly held companies where the owners may be more passive. In this particular case, where Red Hat is the only publicly held company, the difference is of course even more significant.

The ownership structure of Red Hat is therefore of little interest for this research, nor should it influence the outcome of this thesis.

5.5.1.6 Community and customers

Since the introduction of Red Hat’s enterprise product in 2003, all community effort has been directed to the Fedora project. Even if the community reaction was hostile from the beginning, a large number of the enterprise product has purchased the retail version of the Red Hat Linux distribution, and the protests have abated. Today, Fedora is considered a successful project, which may have enabled Novell and Sun to start similar projects.

On the customer side, Red Hat has always had a strong presence in the American financial sector and governmental institutions, a presence that was reinforced after the introduction of the enterprise products.

However, with its present platform of products and services, the company is addressing a very horizontal and wide array of customers. Red Hat is at this point accumulating ten to twenty thousand customers per quarter, which further confirms this horizontal position.

5.5.1.7 Brand

Red Hat is one of the most popular Linux distributions, and has been so for a number of years. Even if the company has geared itself towards larger enterprise customers, the brand recognition within the OSS community remains very high.

The company does not have a separate branding strategy between the community and its enterprise customers. Red Hat branding takes more of a unified and holistic standpoint, where some parts appeal more to the community. Michael Tiemann states:
We see ourselves as innovators, we see ourselves as Open source true believers. We see ourselves as committed to the community in terms of participation. We are not just cheerleaders; we have people in the field actually taking the hard tackles.\textsuperscript{43}

The participation is not only through paid development in OSS projects, like the Fedora Linux distribution, but also by actively working in other organizations like Open Source Initiative (OSI). In the management team, Michael Tiemann serves as the liaison with the community. By giving Fedora its own brand, Red Hat wanted to communicate that it is an independent distribution, but supported by the company.

The great reputation within the OSS community may have been one of the factors for the successful public offering in 1999, which made large investors interested in the stock. After the successful public offering (IPO) in 1999 and the general hype of Linux, Red Hat quickly became a widely recognized brand. This also made Red Hat Linux the market leader among Linux distributions. Other commercial distributions, like Caldera and TurboLinux also made public offerings, but never gained the interest that Red Hat did. Red Hat’s OSS project was also the largest Linux distribution with a market share close to 50 percent (Shankland, 2000).

The concept of creating an enterprise version of Linux, which Red Hat was the first to do, shows the innovative commitment of Red Hat and its targeting of key customers. In addition, the products have been ranked “top enterprise software” for several years, further establishing the corporate brand.

Red Hat also acquired JBoss, one of the most well known application servers (LaMonica, 2006b). JBoss was a close partner to Red Hat and had similar offerings, with subscriptions such as the preferred revenue model. With the acquisition, Red Hat did a forward vertical integration with the purpose of strengthening the OSS platform it offered.

One important part of the corporate branding has been partnerships with multinational companies in the IT industry such as IBM, HP and Dell.

\textbf{5.5.1.8 Promotion}

Red Hat as a brand and its products are promoted through direct sales, indirect sales, channel sales, and partnerships (such as pre-loaded computers), but mostly through the company web site. It has also been a distinct policy for Red Hat not to do virtually any advertising.

In its promotions, Red Hat drives a theme of not focusing on products but on what customers \textit{can do} with the current level of technology. The web site and online Red Hat Magazine do not promote a product in particular, but is aligned towards customer productivity and presents success stories to enhance its claims.

The company also attends and sponsors selected fairs and events. However, Red Hat recently decided to drop the LinuxWorld conferences and other

\textsuperscript{43} Interview, Michael Tiemann, Vice President Open Source Affairs, Red Hat Inc., December 10, 2007
industry events, in order to focus more on its own Red Hat Summits events.

5.5.1.9 Culture

Red Hat strongly emphasizes that the company stands for OSS ideals. Even if the company now spans five continents and has more than 2000 employees, it still thinks it can retain its small-company spirit and remain flexible enough to adapt with new challenges (Red Hat 2007b).

The marketing department is in charge of maintaining the culture. The company tries to reinforce a culture of openness despite of its relatively large size, and the threats from competitors and demands from investors. This attitude may very well be a main reason for attracting the best developers from the OSS community.

Nevertheless, the challenge lies in the transcendence of OSS that goes beyond corporate boundaries. In a large corporation, different aspects of the inside and outside worlds collide. In some aspects, the company may influence the culture, while in others, as with OSS culture, it may not.

5.5.2 Key operating platform factors

5.5.2.1 History and paths

Red Hat was founded in 1995 and was one of the first larger Linux vendors, offering its “flavor” of Linux in a retail version via CD-ROM and manuals. In addition, to this Red Hat also offered consultancy and support services, but mostly on the American market.

However, the profits were low and the demand for Red Hat to make profit was sharpened after the IPO in 1999 and the so-called “dotcom-bust” in 2001. In response, Red Hat introduced its Enterprise line of products in 2003, which targeted the enterprise market, rather than the retail market. This change has been successful and Red Hat has proved profitable for several years.

In 2006, Red Hat widened its offerings by purchasing one of the leading OSS middleware companies, JBoss. JBoss develops a middleware software platform which handles web-enabled applications (which the customers program). JBoss is for applications using the Service-Oriented Architecture (SOA) (Red Hat, 2007c).

Red Hat has been the commercial market leader for several years, and when the company made the enterprise fork, the relationship between the company and the outside world suffered greatly. This included the relationship with its paying customers.
After we released the enterprise Linux, version three we started to get some very consistent feedback from our biggest customers saying, “You guys are more difficult to work with, in terms of new features, than even Microsoft. How the hell did this happen?”

After this, Red Hat fully understood the importance of maintaining a balanced relationship with the ongoing community. The company has worked hard to rebuild the trust between the company, the community and the customers.

In essence, Red Hat is also the company that introduced the subscription model for OSS. Some, mostly regarding the content of the product, have criticized this revenue model, while others are more positive (cf. Shankland, 2004; Asay, 2006 and 2007).

Red Hat was among the pioneers when it came to commercializing Linux distributions, but was also an early adapter for the new revenue models. In the beginning, a great part of the revenue came from selling accessories, like T-shirts and mugs. Red Hat has offered services for years, and not just consultancy services, but also training and certification of customers. The Red Hat certification for Linux administrators are the oldest, still active, certification program for Linux.

The step from the retail business to the enterprise market should be considered a quantum leap for the industry. The question remains if any other Linux company could have made that step and still survive. The successful IPO and the company’s market lead were probably the reasons it managed to endure after all.

Red Hat is among the few profitable companies that still adheres to the OSS principle. The company has been an innovator, not only considering its products, but also in how it makes revenue.

5.5.2.2 Experience and competencies

Red Hat recently hired a new CEO in order to handle the growing corporation. As a public company, the management team and board of directors have to be experienced and competent. Since the company is also able to acquire inventive firms, many of the most competent developers and managers are still within the company. One example is Michael Tiemann, co-founder of the first all-OSS company Cygnus Solutions and very active in the OSS community.

Red Hat has also been a market leader in the Linux community, and as such, it has been able to recruit a great number of top developers over the years. Red Hat is still a popular company to work for, and is able to attract some of the best programmers in the world.

44 Interview, Michael Tiemann, Vice President Open Source Affairs, Red Hat Inc., December 10, 2007
5.5.2.3 Processes

Red Hat manages its enterprise product in-house using version control systems and other managerial software tools. Fedora is completely managed outside of the company, on separate servers, Internet connections and software tools.

Requests or new features are first introduced upstream, secondly into Fedora, and finally, after being tested and certified, into the enterprise product.\(^{45}\)

Nevertheless, in the end, when it comes to the actual content of the products, for example the enterprise Linux product, approximately 85% of the content stems from outside of Red Hat.

5.5.2.4 Intellectual property

Red Hat does not have copyrights or other patents regarding the Linux core, but it does have a portfolio of other copyrights and patents. Red Hat is one of the co-founders of the Open Invention Network (OIN), which is a mutual patent pool for companies to share their patents in a way that promotes innovation.

Red Hat retains patents and copyrights for defensive purposes, but in addition, the company has published a “patent promise” which protects those who uses them in reciprocal licenses (like GPL). The company does not extend the promise to “light” licenses like BSD, because the code and patents may be picked up by hostile entities, which then could use the patent against Red Hat.

5.5.2.5 Agreements

Red Hat has a wide array of agreements, some aimed towards customers like Service Level Agreements (SLA); others are used when dealing with partners, like partnership agreements or reseller agreements. Most of the partnership agreements are created case-by-case. The use of agreements is regulated by the laws of United States, and that legal context may be a hindrance in other legal contexts.

5.5.2.6 Economy

Red Hat is a publically held company and has been profitable since 2004. It is traded on the New York Stock Exchange (NYSE), and all financial reports are public.

5.5.3 Key offering factors

5.5.3.1 Price model

Red Hat sells its products both directly and indirectly. The prices and differ-

\(^{45}\) Upstream means that if a feature or a patch is considered interesting enough to be included in the main development branch, the patch or feature is first presented *upstream* to the core developers (asking them to review and enhance it).
ent offerings are clearly shown. The prices are also shown in the different currents on the web site dedicated to a certain country.

The reseller channel is mostly promoted through discounts, but some partners, depending on their business, may have their own price lists, which are regulated through their agreements with Red Hat.

5.5.3.2 Product/service mix

Red Hat offers products and service solutions directly from its web site. The products range from Red Hat Enterprise Linux, which is for servers; Red Hat Enterprise Linux Desktop, which is for desktop computers; JBoss Enterprise Application Platform, which is the application middleware; Red Hat Command Center, which is a monitoring system; and Red Hat E-training courses. In addition, Red Hat also offers solutions based on its products.

5.5.3.3 Revenue models

Red Hat offers its products and solutions through a licensing subscription. The subscription contains the software, upgrade of the software (through the Internet) and support connected to the software. The subscriptions that are offered through the web site are for one year, but it is possible to sign longer agreements, but only through direct contact with a reseller or the sales department.

In addition to this, Red Hat offers additional services such as Service Level Agreements, training and certification and of course the essential accessories, all through the company web site. However, there have been some discussions regarding the Red Hat subscription model, claiming it limits freedoms normally associated with OSS. For example, a customer to Red Hat is not able to have any other Linux distribution connected to the subscription (Murdock, 2004). Regardless of if the subscription is proprietary or to what degree, in this thesis it is considered a hybrid license.\(^\text{46}\)

An interesting feature, which Red Hat and some of its partners offer, is the Red Hat Exchange. The Red Hat Exchange offers a unified Service Level Agreement, which enables a customer to pick and chose services by any constellation of thirteen OSS software products. One of those partners is MySQL.

5.5.3.4 Distribution model

All Red Hat software is made available electronically through download; the preferred channel of distribution. Other partners may offer pre-loaded software as a bundled offering. Red Hat also offers media kits for those who prefer that kind of distribution.

\(^\text{46}\) Whether the subscription matches the open-sourced code, or limits the freedom of the user is not really a matter of importance unless a third party feels wronged and sues Red Hat. This issue may very well be discussed back and forth for years to come, as some are for and others against it (cf. Asay, 2006; Asay, 2007; Murdock, 2006).
6 Cross-case analysis

This chapter looks to all the cases, and compares them with each other.

6.1 Features

There were three principal factors found common to the cases: 1) they offer a product-like software package to their customers; 2) they are all deeply involved in the OSS community; and 3) their revenue model has a hybrid-licensing scheme. Most of them (3 out of 4) have also changed their business strategies from service-oriented to product-oriented ones, and the other two have refined their initial offerings in order to generate more revenue oriented and the other two has refined their initial offerings in order to generate more revenues.

6.1.1 Summary of the case companies and their offerings

Red Hat offered its brand of Linux in 1995 called Red Hat Linux, with a retail oriented revenue model. The company packaged software onto a media (usually floppy discs or CD-ROMs), added a manual and sold the package inside a box together with some basic installation support services. This offer was centered towards desktop computers and sold through the Internet and large retail vendors. In 2002 this approach was abandoned and a more business-to-business oriented product was introduced, Red Hat Advanced Server, later Red Hat Enterprise Linux. This product is licensed through a subscription revenue model, with annual fees for upgrade and support services. At the same time as the enterprise product was introduced, Red Hat abandoned its former ties to the community. This created uproar among both customers and the community, so in 2003 Red Hat created the Fedora project, which would be the community project from which the enterprise version is based on. Red Hat is the top corporate contributor to the Linux Kernel, but is also active in other OSS projects (Kroah-Hartman et al, 2008).

MySQL was originally a consultant firm, and whose internally developed database engine was used as part of the services offered. After releasing the database as a product for Windows, a dual licensing scheme was used, where Windows users had to pay to get access to the database, while Linux and UNIX users received it free. MySQL still offers a dual license, as part of an Enterprise offering, consisting of the software and support services. The revenue model is a subscription model with annual fees. MySQL was released as OSS quite early and has a well-grown community, where development is focused on third-party add-ons to the product. The database engine software is open, but most of the development is made in-house.

Codeweavers was founded as a consulting firm. Codeweavers worked as consultants in the Wine OSS project, which enables software from Windows to run on Linux. When its main customer, Corel, pulled out from the contract in 2002, Codeweavers decided to continue as a product company em-
ploying among others the maintainer Alexandre Juillard. The CrossOver product consists of software from Wine and additional closed components and is offered as a fully proprietary licensed product. In addition, Codeweavers also offers services based on its product and the Wine project. The Wine project is a relatively small OSS project, and most of the active developers are employed by Codeweavers.

Cendio, founded as Signum Support AB in 1992, had as its first business mission to provide support for OSS. In time, the company developed products based on OSS but packaged with hardware, like the ReadyNet server and Fuego Firewall. A main part of the revenue still came from consulting services, mostly in the form of programming. In 1999, after the change of name, Cendio had three different divisions: consultancy, products and systems. All business was tightly connected to the telecom industry, and in 2002 most of the firm’s customers had withdrawn their contracts. The owners and management decided to recreate the company as a software product company. In 2002, Cendio introduced the ThinLinc product, which was offered as a proprietary product. In addition to the licensing revenue model, Cendio also offered certification training for resellers of the product as a part of its offer. ThinLinc is not an OSS and does not have a project, but Cendio is working and contributing in other OSS projects, such as TightVNC and rdesktop.

Table 4 shows the cases, their product and the OSS product from which it is based:

<table>
<thead>
<tr>
<th></th>
<th>Product</th>
<th>OSS project(s)</th>
<th>Revenue model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cendio AB</td>
<td>ThinLinc</td>
<td>TightVNC, rdesktop etc.</td>
<td>Proprietary</td>
</tr>
<tr>
<td>Codeweavers Inc.</td>
<td>CrossOver</td>
<td>Wine</td>
<td>Proprietary</td>
</tr>
<tr>
<td>MySQL AB</td>
<td>MySQL Enterprise</td>
<td>MySQL</td>
<td>Subscription/dual license</td>
</tr>
<tr>
<td>Red Hat Inc.</td>
<td>Red Hat Enterprise Linux</td>
<td>Fedora, Linux Kernel, RPM etc.</td>
<td>Subscription</td>
</tr>
</tbody>
</table>

6.2 Business model analysis

By using the business model framework as an analysis tool, it is possible to compare the companies regarding key factors. The first part of the analysis, therefore, will be conducted through the “strainer” which the framework provides.

6.2.1 Market positions

6.2.1.1 Business ideas, visions and goals

All four companies have similar business ideas (or missions): to bring high-
quality software based on OSS to paying customers for a profit. However, as business ideas go, most seem to be diluted and general, rather than sharp and to-the-point. Aside from this, the companies’ size and aims differ widely, but the common vision is to expand their market shares and generate more profit based on their current business strategies. Both the similarities and variations in the business missions are well aligned to Normann (1975) regarding a system for dominance and made in accordance to their market.

There is however one exception, namely Codeweavers. Its ultimate goal is to become a consultancy firm once again, offering solutions for companies that are interested in porting their Windows software to Linux or Mac OS. Its current mission statement is considered temporary, and by offering a product, the improvement the software needs in order to serve the ultimate mission is financed.

Looking at the short-term goals, most are similar in all of the companies. Even if Cendio, Codeweavers and MySQL’s goals are aimed towards profitability, Red Hat’s goals are to enhance its current market position and offer better products to its customers. However, since Red Hat is a public company, the “usual” goals preferred by analysts – such as market growth and higher profitability – are a part of the picture.

6.2.1.2 Competitors

The case companies are all active on separate sub-industries; this means that they have a different set of commercial competitors and different ways to handle that. However, Codeweavers has one product that is somewhat competitive to Cendio’s, but the companies are still partners.

Two of the firms are essentially competing with themselves, since Codeweavers and MySQL have OSS projects with virtually the same product as their own offerings. This applies to Red Hat to some degree, even if the Fedora project does not share all the features of the enterprise product. Cendio does not have this issue, since its main software component is proprietary.

All the companies studied for this research have OSS competitors or substitutes, aside from their own OSS projects, which emulate some or all of the features in their products. This is not considered an issue, but rather a sign of healthy competition.

6.2.1.3 Competitive strategies

The dominant strategy for the case companies is the overall cost leadership strategy (Porter, 1990) (also known as the “low-cost” position), with the possible exception of Red Hat. Depending on which market you look at, Red Hat’s offerings are either low-cost (such as its application server product) or high-cost (comparing Red Hat’s Enterprise Linux with other Linux distributions), and thereby work through differentiation or even focus strategies (Porter, 1990).
Nevertheless, this pattern correlates to all the cases. Cendio’s product is a low-cost alternative to the market leader Citrix, but high-cost compared to OSS products or even Microsoft. Codeweavers’ product is a low-cost competitor on the Mac OS market, but is highly priced compared to Wine itself. Moreover, MySQL has the same situation as Codeweavers, since it competes with its own OSS product, while having a low-cost strategy compared to the market leaders such as Oracle, IBM and Microsoft. In these cases, the offerings from our case companies try to differentiate themselves from their competitors. For example, Cendio’s product offers more functionality than the cheaper products, and outshines the market leader in some ways, but not in others.

It is clear, however, that all the companies benefit from the user-driven development from OSS. Since most of the case companies, with the exception of Cendio, have one project they can draw OSS contributions from, they are able to develop their products at a lower cost than proprietary competitors, thus enabling them to charge less for the software. Even Cendio benefits from OSS, but not as directly as the other case companies, since a lot of the functionality of Cendio’s product stems from OSS components; it also have a clear competitive advantage compared to some of its competitors.

6.2.1.4 Partners

Partners are important for all the case companies, and the relationships are mostly sequential (Thompson, 1967). However, the importance that the partners understand the philosophy of OSS is very clear. Even if no solid evidence was found in this research, there are indications of cultural issues that may render a partnership successful or a failure. As Normann (2001) notes, in a partnership that creates a new value system, the partners need to be at least somewhat aligned. Since the openness and flexibility are key cultural elements for these companies, the assumption made here is that the partner company must share some parts of these elements in order to be viable.

Some of the case companies are partners. Cendio and Codeweavers’ products complement one another, so Cendio acts like Codeweavers’ reseller in Sweden. MySQL and Red Hat’s products also complement one another, and the two have collaborated for a couple of years.

6.2.1.5 Owners

The issue of corporate ownership and strategic guidance differs extensively between the companies studied. In fact, the differences are so great that is difficult to find any common ground or indications of best practice.

Cendio would not have survived without its financial backers; today, it is buying time to gain market share and volume for its product. From another perspective, the financial backers do not have the means to boost Cendio, but rather just keeping its head above the water. As for the guidance, the
owners seem not aware of the true benefits of OSS, and are thus a hindrance for Cendio to become more open source.

Codeweavers does not have any substantial financial backer. The company is being approached by venture capitalists, but the lack of understanding of how OSS works and the perceived risk of being sued by Microsoft has thus far led to no financial injection. This has forced Codeweavers to become very lean, working more or less hand-to-mouth and not being able to plan. Moreover, since the CEO is also the biggest owner, the strategic guidance is applied on a daily basis.

MySQL, on the other hand, managed to secure not only substantial capital injections, but also benefited from strategic guidance made by board members appointed from the investors. These professional executives provided the company with valuable advice and support during their terms, and may very well be a significant part of its success. However, since MySQL has been acquired by Sun Microsystems, which is a public company traded on the NASDAQ Stock Exchange, the company will make major changes in the ownership situation.

Red Hat is a public company traded on the New York Stock Exchange, and as such its ownership, managerial team and board members are constantly scrutinized and evaluated. The ownership and the strategic guidance provided by them may therefore shift during time, making it hard to make any assumptions about quality or consistency.

6.2.1.6 Branding

Branding is important for most companies, but for OSS companies even more so. An OSS company may have to work with three or more different brands; there is one brand to represent the company, one brand for the product and one brand for the OSS project. In addition, the brands affect each other. A solid brand name in the OSS community may very well be positive for the company and product brand. An interesting analogy is the hypothesis presented by Thompson (1967) regarding domains and the open-system strategy. Since the company needs approval from the community in order to access the pooled resources it needs from the community, the similarity is fitting enough. In order to maintain the support from the community, the firm needs to balance the brand between for-profit and sharing demands.

Two of the companies studied had a significant marketing advantage on the other two. MySQL and Red Hat are well known, if not famous in the community and well-known brands. The status of the OSS brand has, during time, also leaked into the corporate brand, which trigger interest among both customers and investors that (often) reflects back to the OSS brand. However, even if both Cendio and Codeweavers are known, their brands are often more recognizable in a smaller, more specialized arena.

By looking at four different market positions, the brands of the case com-
panies are more or less recognized in different arenas. This research has looked into four different settings, specifically how known is the company brand in: a) the OSS community in general; b) the main OSS projects where the firm is active; c) the “market”, i.e. paying customers in general; and finally d) more localized and focused groups of customers. The legends are N for “not known”, LK for “less known”, K for “known”, VK for “very known” and finally F for “famous”. Table 5 below shows the market positions of the case companies.

Table 5: Market positions

<table>
<thead>
<tr>
<th></th>
<th>OSS Community</th>
<th>Primary firm OSS-project</th>
<th>Customers (general)</th>
<th>Customers (local or focused)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cendio</td>
<td>N</td>
<td>LK</td>
<td>N</td>
<td>K</td>
</tr>
<tr>
<td>Codeweavers</td>
<td>K</td>
<td>VK</td>
<td>N</td>
<td>LK</td>
</tr>
<tr>
<td>MySQL</td>
<td>F</td>
<td>K</td>
<td>K</td>
<td>F</td>
</tr>
<tr>
<td>Red Hat</td>
<td>F</td>
<td>F</td>
<td>VK</td>
<td>F</td>
</tr>
</tbody>
</table>

MySQL and Red Hat are brand names that are well known within the community, perhaps because they have been active for a long time and both have been part of what is known as the LAMP stack for web servers.

Cendio retains some fame for earlier products and services, both in its present name and the former brand Signum Support. However, regarding its new product, the information is not as widespread. An interesting notion is that there are a lot more hits for pages in the online manual of the product than for the product itself. This is because the manual covers issues of using proprietary systems, like Novell and Microsoft, to run alongside Linux.

Codeweavers is not very known even among its customers, since it has downplayed its company brand in favor of the OSS project (Wine) and their line of products (CrossOver). A customer may very well be a long-time customer of Codeweavers and still be unaware of the name of the company.

When a firm is known in the OSS community, it enjoys several competitive advantages, some through marketing, and others for the operational platform; the latter is discussed below.

6.2.1.7 Corporate branding benefitting from fame

Both Red Hat and MySQL are enjoying being well known all over the, not only in the OSS community but also in the market in general. This means that they have easier access to customers, partners and/or investors. Even if

---

47 The information about how well known the company is has been collected from secondary sources like Wikipedia and searches for corporate and product names on Google. Since this information is highly volatile, no efforts to save individual searches were made.
both are working to enhance their brands outside the OSS community, their marketing budgets are significantly lower than for a proprietary firm in the same position, freeing up marketing resources.

An interesting notion is that both companies have merged their brands. MySQL has the same name for all three parts, namely the company, the OSS project and the product. Red Hat chose to have a different name on its OSS project in order to communicate the independence of the Fedora project.

Merging the brand names creates interesting synergy effects and works reciprocally. If the firm strengthens its branding towards paying customers, the interest for the OSS projects also increases, which in turn increases the interest from the market. MySQL has used this cycle to attract venture capital, and was recently acquired by Sun Microsystems.

6.2.1.8 Positioning without fame

In comparison with MySQL and Red Hat, Cendio and Codeweavers have a different situation. In Cendio's case, neither the brand for the product nor the firm is well known. Rather, the two have been forced to devote extensive time in order to gain resellers and to be recognized within their target markets. In addition, they suffer from the fact that their reselling partners often have a large portfolio of products and services; in some cases, the resellers even offer competitive products or solutions.

Moreover, since Cendio as a very small, privately held company, and lacks the resources for marketing, the resellers have not focused on its product. Cendio has therefore been forced to employ sales staff who conduct direct sales to end customers, and who turn over projects to resellers at a certain stage. Aside from this, the executives for Cendio have had problems explaining the benefits of using OSS in the product for investors and customers; in the past few years, however, this has improved significantly.

Cendio did focus on the municipal and governmental market very early. This kind of customer is usually interested in economically beneficial solutions and is not likely to use unsupported software. By aggressively focusing its marketing measures on this market, Cendio has obtained a solid reputation among municipalities in Sweden. Cendio is now using this advantage to gain access into the healthcare market, which in Sweden is similar to the municipalities.

Nevertheless, one of the key issues for this kind of customer is the time between the sale and getting the order. Another big issue for Cendio is the lack of volume in its current markets. In order to be truly profitable, the company needs to gain magnitudes of customers, but if it succeeds, its development resources may be insufficient. Therefore, Cendio is looking to tap into more resources from the OSS community, which would be primarily for marketing, but secondary for gaining development resources. Because of this, Cendio is looking into turning its product open source.

In the case of Codeweavers, the situation is altogether different. The OSS
project (Wine) and the product brand (CrossOver) are more known than the company name. However, this situation was created on purpose, and it still creates problems when seeking investors and partners. Recruitment of developers has primarily been through the Wine project. Codeweavers is not interested in working its brand name as such; its desired position would be as a partner for ISV, which would like to run its Windows software on Linux or Mac OS. However, in the emerging Mac OS market, the lack of marketing and a known brand makes Coldweavers’ product less known than its competitors’ products, such as Parallels and WM-ware. Moreover, since this market is becoming more and more important for Codeweavers, this can be a problem.

Codeweavers’ second issue, which could be translated into an opportunity, is that on the Mac OS market, the Wine project brand is virtually unknown, since users of Linux and Mac OS up until now have lived in two separate worlds. The differences between the users are great: Linux users prefer to download the free Wine version, while Mac OS users prefer to purchase a box through a retailer. This also means that for the Linux market, Codeweavers can make use of the Wine project as a marketing channel, but this option is closed for Mac OS. In order to address this, Codeweavers has stopped appearing at Linux trade shows, focusing instead on the Mac OS market.

The two small companies have struggled to become profitable, but each is coming from opposite directions: Codeweavers seems to have found a promising market, and Cendio needs to enlarge its niche market. It will be interesting to see how both companies manage to obtain their goals in the next few years.

6.2.1.9 OSS effects for marketing positioning

The effect for OSS companies concerning market positioning is that if the product or company reaches fame, it gains a significant competitive advantage compared to proprietary competitors. However, the advantage may turn into a curse; the fame may very well be fickle and perhaps impossible to manage. For both MySQL and Red Hat fame, was built during a succession of years before it could be exploited; the interest could just as easily been directed to some of their competitors. It was luck, and of course, hard labor, which created a functioning product or service and enabled their competitive advantage.

The difference between the case companies becomes clear when looking at the smaller firms, Cendio and Codeweavers. Codeweavers is profiting somewhat from the Wine project, but for Cendio there is no project to profit from. Cendio really has had its work cut out for them; without big marketing funding, it had to work its market “by hand”. Now, several years after releasing its product, the company has gained some awareness in some markets.

This means that for smaller firms working without the fame of a well-
known OSS project, there is really no difference between being proprietary or OSS; in fact, it might even be a slight disadvantage, depending on the penetration of OSS in their target markets.

6.2.1.10 Promotion

It is clear that one of the most important competitive advantages for OSS products is in marketing (Goldman and Gabriel, 2005; Fitzgerald, 2006). None of the case companies spends large amounts on marketing; most of the marketing comes from the community, the users and via word-by-mouth. A large community of users spreads the word very effectively, which lessens the need for excessive marketing.

MySQL, however, has marketed its commercial product somewhat in order to place a focus on its professional side. However, the amount spent is significantly lower compared to most of its competitors.

The companies mainly promote their products through their web sites, via the sales department and by attending trade fairs. Nevertheless, most of the companies have also begun to focus their attention on special, rather than general, events. Cendio, for example, has aimed its marketing efforts towards specific events for municipalities, and most recently healthcare. Codeweavers has changed focus from Linux to Mac OS, leaving the Linux events in favor to Mac OS fairs. Finally, Red Hat has decided to cut down its presence at Linux events, focusing on other events more suitable for its needs, such as developers’ conferences, etc.

6.2.1.11 Cultural findings

Company culture is was found to be extremely important for all the case companies. Openness and a flexible organization are attributes that all four companies consider essential. For the smaller firms like Cendio and Codeweavers, this is of course their only option, but MySQL and Red Hat stress the fact that openness and the feeling of a small company with little hierarchy is one of the most important factors for their success. Culture is of course influenced by the OSS community, since it is part of the firms’ domain (Thompson, 1967). The company not only gets development, support and testing from the community, but also since the main part of the staff in all the cases are technicians or software engineers, many are hired from the OSS project.

Both Red Hat and MySQL are working hard to preserve openness, and Red Hat points out its wish to be perceived as a small company by its employees, rather than the multinational corporation it really is. MySQL is blessed, or hampered, depending on one’s point-of-view, by the fact that many of its developers work from home. On one side, the image of the small company is of course benefited by this fact. On the other side, the social aspect of being a company suffers. MySQL is therefore trying to convene developers’ conferences several times per year all over the world. This is of course costly, but
the company usually offers non-employed staff to join, which creates an opportunity for the community to meet.

As for the smaller companies, Cendio and Codeweavers benefit from the short decision paths. However, Codeweavers shares the same problem as MySQL, having most of its development staff outside the office. The solution to its problem is similar to MySQL’s solution, but the scale and number of events are smaller and fewer. Since Codeweavers also is a lean company, space is premium; when managers share offices in the same way as the staff, it reinforces the image of a small, open company. For Cendio, where all the staff is gathered in a single location, the openness and small-scale imagery is no problem.

The cultural findings are well in line with what Normann calls “aligning the company” (1975). Even if the business ideas, vision and goals differ, it is important for the companies to preserve the identity of being part of the OSS community. This is important for these companies in order to balance the community, customers and owners.

6.2.2 Operational platforms
The operational platform of OSS firms is where the competitive advantage becomes clearest compared to proprietary firms. Large OSS projects may have thousands of users and programmers who test and develop the software for free. These resources, albeit not fully controlled, come without cost for the company and are therefore a huge competitive advantage compared to proprietary software companies. However, even if this kind of resource does not reside within the firm, not every company is able to utilize the community to its fullest; it should therefore be considered a competitive advantage in the light of RBV (Wernerfelt, 1984). In addition, considering that the IT industry is considered a demanding industry and that the mood of the community can change rapidly, the capacity to handle the community should be considered a “dynamic capability” (Teece et al., 1997).

In theory, companies could exploit this, but in reality, the main code contributions come from a select few, many of whom are employed and paid by companies to develop OSS code (Fitzgerald, 2006).

From a company standpoint, this would look excellent, but using or working with OSS projects may not always be a “walk in the park” (Goldman and Gabriel, 2005). Many situations can easily become conflicts of interests. Perhaps the most important conflict regards deadlines. Since one of the main points when promoting OSS has been the quality of the software, the concept of deadlines has never been an important issue for the project (Raymond, 2001). A company has always had the freedom to work with the software until it is “as good as possible”; this is contrasted with the paradigm of
“good enough”, which has been used for proprietary products. Nevertheless, the companies working closely with an OSS project or the OSS community at large have to walk a narrow path and balance between the market and the community. This is when the pressure for deadlines that the company exerts on the community may affect the release of products. If the company pushes too hard, or tries to constrict the community with rules and legislation, the perceived legitimacy of the company can be hurt, and subsequently community support and contributions may grind down to a halt. MySQL and Red Hat have witnessed firsthand what happens if the community is taken for granted, or if attempts are made to impose rules or deadlines on them. Even Codeweavers watched when TransGaming, a competitor who used the Wine project without giving anything back, got virtually no support from the community at all. Keeping the community happy is something very important for the case companies.

The size of the project is also a good indication of its popularity. However, the popularity of the project may not reflect the numbers that actively contribute code to it. The code base may be so complex that it would take months or even years for a new programmer to become proficient enough to contribute. In addition, aside from that, the possibilities to exploit the code for commercial firms are dependent on individual copyright and licenses (cf Lerner and Tirole, 2005, Goldman and Gabriel, 2005; Fitzgerald, 2006). Table 6 shows some of these factors for the case companies:

**Table 6 : Popularity, IPR and complexity**

<table>
<thead>
<tr>
<th>Main OSS project(s)</th>
<th>Project popularity (usage)</th>
<th>Retain essential copyrights</th>
<th>Permissive OSS license</th>
<th>Complexity of the code base</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cendio</td>
<td>TightVNC, rdesktop</td>
<td>Differs</td>
<td>No</td>
<td>Differs</td>
</tr>
<tr>
<td>Codeweavers</td>
<td>Wine</td>
<td>Popular</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MySQL</td>
<td>MySQL</td>
<td>Popular</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Red Hat</td>
<td>Linux Kernel, RPM, Fedora</td>
<td>Popular</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

For companies that are oriented through a single project, such as Codeweavers and MySQL, it is of course easier to manage development. In both cases, the firms are in control of the development of core software, but less so when it concerns add-on or third party applications. Nevertheless, since a main part of their business exists because of third party applications, Codeweavers has to pay more attention to them.

48 The concept of “good enough” has been dismissed as a myth, but looking at the software business during the last three decades, the concept of releasing software early has been an important economical factor, promising new features, functionality, and locking the customer in with a need for continuous upgrades.
The situation for Cendio and Red Hat is different. These companies have interests not only in their own in-house software, but also have stakes in several other OSS projects. Since one of the principles of OSS development always have been “quality first”, deadlines has traditionally neither been respected nor needed. For a company like Red Hat, one with a larger number of employees and resources, the adaptation of an important project to make a deadline is less of a problem compared to Cendio, which works with limited resources. For a small company like Cendio, with little or no means of controlling the important OSS, it might be forced to adjust its roadmap in order to be able to include important updates from the OSS projects into its new releases.

6.2.2.1 History and paths

The cultural underlining all of the companies is reflected in their history. All of the companies were founded in the 1990s, and all wanted to work with OSS (or free software as it was known at the time). The oldest company, Cendio, was founded in 1992. Nevertheless, the oldest company working with free software was Cygnus Solutions, which was founded in 1989, acquired by Red Hat in 1999 and retains some of the staff.

This means that the company has had extensive experience working with OSS, and in most cases has tested different business models, with consulting services as a common ground; this path may very well have formed its later choices (Teece et al., 1997). Most of the companies turned, relatively early, towards other businesses, such as distribution (Red Hat), hardware and software packages (Cendio) and product development (MySQL and Codeweavers). Cendio is the company that has had the most changes in terms of different products and offerings. Cendio and MySQL also tested hybrid and proprietary revenue models earlier than the other companies did.

For the companies that decided to test a new way of making money from OSS, they gained an ability to charge. Nevertheless and regardless of the cause, all case companies now possess an ability to charge their paying customers beyond the scope of hourly rates. All the studied companies went began with consulting, which is mostly charged by the hour, but soon realized that the margins were too slim. It is easier to charge per hour since it can be based on actual cost, rather than from market pricing. However, there is a limit on how much revenue one can generate from consultants, and some of the solutions created for a client might fit others. Therefore, the step from going from a service-based consultancy firm to a software product company could seem small, but in actuality over 87 percent of these ventures fail, as it is an entirely new game with a new set of rules (Nambisan, 2001).

It may be so that OSS companies have an easier journey to become software vendors than other proprietary companies do. Out of the five issues Nambisan (2001) presents, namely intellectual property rights, product complementarity, returns from scale, abstracting knowledge and integrating tech-
nology and the connections with users, all but returns from scale are handled through the OSS project. However, in order to be able to charge, the firm also has to set the price correctly; this is usually the pitfall for many burgeoning software vendors.

This is where experiments on revenue models can help an OSS company to gain the ability to charge customers. Looking at the case companies in this research, all but Codeweavers had numerous experiences in revenue experiments, which would make the transition from a service-based to a software vendor company easier. Codeweavers still considers the proprietary revenue model as temporary until the time comes when it can become a company of consultants again.

It is safe to say that the recession in the IT-industry in 2001 and 2002 put pressure on all the case companies, and each of them responded by changing their business models. Cendio and Codeweavers turned consulting firms into a software development companies. MySQL and Red Hat aimed their products towards the enterprise market and changed their revenue models. Moreover, for the two small companies, Cendio and Codeweavers, the change was a necessity for survival.

MySQL and Red Hat have been pioneers when it comes to revenue models. MySQL was one of the first companies that used the dual licensing model, combining a proprietary license with an OSS license (Wichmann, 2002). MySQL’s take was also different, since it did not offer different software, only different licenses. Red Hat, on the other hand, was among the first to use subscription as a revenue model for the type of services it offered. The combination of software, updating services and support for enterprise customers was new to the OSS industry (Asay, 2006).

All these companies have been in the business for an extended time, and all share a common foundation in the form of consulting services, but all evolved through different paths to reach their present status. When looking at their history, experiences and paths, all the companies seem to endeavor towards the same future, where OSS is dominant. Where Codeweavers would like to leave product development and return to its original business mission, Cendio strives to become even more open source. Red Hat struggles with the balancing act between being a large corporation and retaining the roots of OSS, while MySQL has been acquired be one of the largest IT corporations on earth. The future looks bright, indeed.

6.2.2.2 Experience and competencies

All of the companies have very skilled programmers. Companies with an OSS project can also use the project as a recruitment tool, picking the most skilled developers based on their achievements in the project. It is an easy and powerful way to recruit top people. Cendio, lacking an OSS project, has managed to recruit very skilled developers anyway who have been accepted as core members in several projects. It is clear that if a company embraces
OSS, it gains a very powerful tool when it comes to recruiting competent programmers.

Another, somewhat surprising, matter concerns the management teams. For Red Hat, as a public company, the management team is scrutinized by the market which pressures the company to recruit top managers. However, in all of the case companies, the management team was very experienced and well suited for the task. Most of the CEO’s, with the exception of Codeweavers, had been recruited from outside of the firm and with good results; it would be interesting to see if this pattern is consistent within the whole OSS industry.

6.2.2.3 Processes

Of all the companies that work directly with the community, all but Cendio hosts both a public and a similar internal development platform. They all consider the differences between the internal and public versions important, but only aim to manage the internal development (somewhat in opposition to Fitzgerald, 2004).

However, all companies work through an internal roadmap, which is not affected (as much) by public development. In most cases, however, the company controls when new major versions are released.

Cendio is investigating the possibilities of open sourcing its product, and one of the most important issues is how the development will be handled. It will be most interesting to see what solution they decide on.

6.2.2.4 Intellectual property rights (IPR)

The intellectual property rights are one of the most essential key factors for a company that produces software, be it an OSS or proprietary company. The copyright of the source code is, by far, the most important element if the company would like to enjoy direct-sale value revenues. However, there is a risk that the community feels that a license is too restrictive and then will abandon a project. Some companies choose to use a more restrictive OSS license, such as GPL, in order to keep the community happy (Lerner and Tirole, 2005).

Cendio and MySQL retain the copyrights of the essential software in their offerings, which enables them to use proprietary or dual licensing in order to gain revenues. Without that copyright, this kind of licensing would not be possible (Välimäki, 2003).

Codeweavers does not own the copyright of the essential software, but the company employs the maintainer of the project. Since the OSS license allows it, the company is able to charge for the use of its software (cf. Wichmann and Stiller, 2002; Bonaccorsi and Rossi, 2003; Bonaccorsi et al., 2003). The lack of copyright and perceived legal implications of its software has an abhorrent effect on investors, who have hampered the development of the company somewhat.
Red Hat, on the other hand, owns some copyrights and patents, but not for the most essential software, the Linux Kernel. This means that Red Hat cannot charge downright proprietary fees, but rather has to create other means of gaining revenue. In addition, the copyrights and patents could be a problem with the relationship with the community, since many such as Stallman (2002) are opposed to software patents. However, the patents are not as essential for the company, which has enabled it to make a strong “patent” promise to calm the OSS community (Red Hat, 2008).

This promise, along with freeing the Fedora brand from Red Hat, are some of the measures that OSS companies have to do in order to be perceived legitimate by the community. If the community withdraws its support and contributions, most of the competitive advantages the company enjoys will be eliminated. The balance between the control of the IPR and the blessings of the community is something both Red Hat and MySQL have prioritized.

In the future, it will be interesting to follow the future of MySQL as a part of Sun Microsystems. Since Sun has moved towards a more open stance towards OSS, changing its licenses to GPL or similar, its take on copyrights, patents and dual licensing will be interesting. However, only a few months after the acquisition, signs of trouble seem to have appeared (Manchester, 2008).

### 6.2.2.5 Economy

The only company that is profitable is Red Hat. Its financial situation has been positive the last few years, and shows a healthy increase both in revenues and in returns. MySQL was on the verge of becoming profitable, showing positive cash flow the last two years, when they it was purchased by Sun.

Cendio is struggling with bad finances, but slowly gaining new customers and interest outside Sweden. However, the company does have a strong financial backer that has given it time to gain momentum. Finally, Codeweavers, which has been struggling for years lacking financial backers, seems to have found the market opportunity it needs to become profitable. It’s only obstacle now is a lack of funds, which makes marketing more difficult. It is looking at the future, however, in a positive way.

### 6.2.3 The offerings

The offering is another factor which the OSS firm can use to distinguish itself from its competitors. The offering is consistent with what Normann (2001) refers to as a bundling of services. Often, the offering consists of more than one product or service. For OSS companies, this is even more the case, since what the company offers may even be services based on software, which is produced by community contributions. Even if they, at a glance, might look alike, the number of services and revenue models offered might attract a larger number of customers.
6.2.3.1 Product and service mix

This research has identified four different services offered: packaging, development, consultancy, and training.

6.2.3.1.1 Packaging service

The term “distribution” is used when a software vendor packages a number of software applications into an installable operating system. Examples of this kind of vendor for Linux are Red Hat, Novell and Ubuntu. Even if they all perform Linux distributions, however they differ in various ways. As shown above, Red Hat has aimed its Enterprise Linux towards servers, while Ubuntu has been more oriented towards desktops.\footnote{There are other differences as well, for instance the software that handles the packaging. This is not relevant in this matter.} Novell also offers, in addition to its Linux distribution, several suites of proprietary software, which may be purchased together.

The packaging service companies are generally OSS companies because their operating systems are not linked through a hardware vendor. Another example of a distribution companies is BSDi, which offers a commercial version based on the BSD operating system. BSD stands for Berkeley Software Distribution, which is OSS, but with a very permissive license.\footnote{The BSD license permits anyone to create his or her own version sell it commercially. Therefore, most commercial UNIX distributions are based on BSD code, including Mac OS X from Apple.} On the other hand, Microsoft is the world’s largest software distribution company.

6.2.3.1.2 Development service

Development is just what the name says: a firm offers programming services to other companies. Here it is differentiated from consultancy (see service below) since it is more of creation than of service. Whereas consultancy may attain simpler programming, for example script programming, development is usually tiered towards larger systems and programs. However, service companies often offer development services as part of their consultancy portfolio.

6.2.3.1.3 Consultancy service

The most common offering for OSS companies is services. Services can differ from packaging (not distributions, but other software packages), development (see above), general computer consultancy, system maintenance, system monitoring, and support. These firms try to book their consultants for as high an hourly rate as possible.

An example of this is the vast numbers of large and small consulting firms, which offers all kinds of consultancy services for OSS.

6.2.3.1.4 Training

Training is about education and is aimed towards other companies. The courses may be certification courses to obtain certain diplomas or certifi-
iates for software. Training may also be a part of a consultancy firm, but it is more common that specialized companies offer training. In order to secure product knowledge among the resellers, software vendors often create mandatory courses with degrees or certifications; Red Hat, for example, has several certifications connected to its distribution.

6.2.3.2 Revenue models

The revenue model is one of the most important key factors for these firms, and is how the company appropriates value (Amit and Zott, 2001). Eight different revenue models have been found: subscription; fee (per opportunity, hourly rate and clip card (and similar)); proprietary license; value-added hardware; value-added proprietary software; and hybrid licensing. A firm may very well offer several or all of the services mentioned. It is likely that companies offer a combination of product and services and use different revenue models for each service or product. The term “hybrid licensing” is used when the revenue model is a combination of proprietary and OSS licenses.

6.2.3.2.1 Subscriptions

Subscriptions are as close as a firm that does not hold copyrights for an OSS can come to a proprietary license. A subscription is an annual fee rendered for a combination of services of distribution and consultancy services, such as upgrades and support services (cf. Goldman and Gabriel, 2005; Perens, 2005). The subscription may also contain proprietary parts such as installation scripts and other software, which can be added without violating the OSS license.

The distribution and the services are only accessible for paying customers, making it almost proprietary. A proprietary software license only gives a user the right to USE the software under certain conditions, while an OSS license gives away extensive rights to the users. By discriminating the access to the software or services through the subscription, it emulates a proprietary license.

MySQL and Red Hat use subscriptions as their primary revenue model for their software products.

6.2.3.2.2 Fees

The perhaps most traditional way to gather revenues for services is through fees. In this research study, three different ways for which services fees are collected were found.

The most common way for a consultancy firm to gather revenue is through an hourly rate. The price per hour differs widely depending on market, type of job, competence or other factors. The pricing may also change several times, often with short notice, also depending on the same factors. Some consultancy firms offer agreements, which settle prices depending on different factors, such as Service Level Agreements (SLA) or the equivalent.
When it comes to training, the revenue structure differs from other consultancy missions. Hourly rates do exist, but since training normally depends on pre-written material, location and other effects, it is more common to charge a fee per opportunity. This could be per person attending, or for the whole seminar or class. The pricing differs widely depending on who is conducting the course or seminar and attractiveness of the material, etc.

Another way for a firm who conducts training courses or seminars is to offer clip cards, which enable a single person or firm to attend several courses or seminars at a fixed price.

All of the studied companies offered some sort of consultancy and training services for customers or partners using these revenue models. However, none of the case companies considered consultancy or training services as their main revenue model, but they are all complements to their software products.

6.2.3.2.3 Proprietary licenses
A proprietary license is the most common revenue model for commercial software vendors (Hecker, 1999; Raymond, 2001). This type of license offers the end user certain, often very limited, rights to use the software under certain conditions. The licenses are mostly based on the copyright laws of the United States and are commonly called End User License Agreements.

As shown above, of note is that an OSS already has a license which gives away many of the rights which the proprietary license withholds. If an OSS firm uses both OSS licenses and proprietary licenses for its product, the scheme is referred as “dual licensing”. The only way for the firm to achieve this is either to own the copyrights for the software or have an OSS license that allows for dual licensing (not all do).

The proprietary license usually has other agreements added to it, such as support agreements or less commonly updating or upgrading agreements.

As for the pricing, the licensing may be charged per user, per seat (installed computer) or per concurrent user. The price variation is very wide depending on the type of application, target markets or customers etc.

Of the studied companies, Cendio and Codeweavers offer proprietary licenses as their main revenue model for their software, but MySQL also offers this kind of license for its OEM customers.

6.2.3.2.4 Value-added hardware
In some cases, a firm creates a product by adapting software (or vice-versa) to a specific hardware, and then selling the whole bundle for one price. This model has been dubbed “widget frosting” (Hecker, 1999; Raymond, 2001). In the bundle there might also be third party or other proprietary software for which the company charges. In addition to this, the firm may also add service agreements such as support and software upgrading. This revenue model was the most common in the early days of computing, and is still
used by companies such as IBM, HP and Dell.

None of the studied companies uses this method today, but some of them have used this model earlier, and it is possible that some of the companies will use it again.

6.2.3.2.5 Value-added software

Some OSS is not complete or does not have all the parts a customer requires. In those cases, if the OSS license allows it, a company may add its own, or third party, proprietary code to the OSS and then repackage it as a new product (Wichmann, 2002; Goldman and Gabriel, 2005; Krishnamurthy, 2005; Perens, 2005). This new product is then normally appended with a proprietary license.

The software may also have service agreements attached to it, such as installation help, software support and upgrading agreements. As in all the other revenue models, the pricing differs widely between companies depending on market, industry etc.

Cendio, Codeweavers and MySQL add proprietary software, mostly their own or from third parties, to the OSS. This way they create a bundled software product for which they can use a proprietary license.

6.2.3.2.6 Hybrid licensing

The term hybrid licensing connotes when a company combines OSS (which normally cannot be licensed for a fee) together with proprietary software or hardware (cf Hecker, 1999; Wichmann, 2002; Krishnamurthy, 2002; Perens, 2005; Fitzgerald, 2006). As presented above, it is quite common for hardware manufacturers to combine their hardware with OSS, which may be an operating system, drivers or utility software. Software vendors add third party or in-house proprietary additions to open software in order to create a product better suited for their customers.

Code Weavers base their product, CrossOver Office, on the Wine project. All development in Wine is fed back to the community, but not every change is accepted. In addition, the product is shipped with additional proprietary software, such as “wizards” for installation, and service, software upgrades, and support by mail or telephone.

MySQL offers an enterprise edition of its database product in the form a subscription. The product is based on the OSS project, but in addition, MySQL offers services and support for the installation and other software-related problems. A customer may also get other services, such as MySQL Enterprise Monitor, which is not available for non-paying users.

Interestingly enough, the OSS hybrid licensing scheme and the product business model incorporate services as part of the product offering. This is mostly due to the legal aspects; Red Hat for instance, does not own the copyright for the Linux operating system and most of the software incorporated in their offering, so they are more or less forced to produce a service-like
product offering.

OSS companies that adopt a product-oriented business strategy can all be associated with the returns from scale factor and the need for continuous revenue streams (cf. Nambisan, 2001). At the same time, the use of OSS in their products more or less negates the four other factors.

An OSS product company does not need to rely on intellectual property rights for its product; the copyright itself is considered good enough, but in one of the cases, it does not matter at all. In all the cases except for Cendio, all the development made by the company is made public through the OSS license, while Cendio retains one part of the product as proprietary.

By using OSS, the interconnectivity between products and product platforms are “built in” automatically making the product complimentary with other platforms and systems, since this development very well may be conducted by others outside of the company. All the products are functional on at least two different hardware platforms, and some even work on a wider array of platforms. The main part of the work of adapting the software for other platforms has been made outside the firms, with the exception of Cendio’s proprietary parts.

Integration and knowledge abstraction is a key feature of OSS. The basic software may very well be platform-independent and generalized, and a product firm may very well adapt the software towards a certain customer segment or vice-versa. All the case companies studied for this research tended to create generic products, even if the development outside the firm might adapt the software for different specializations.

Finally, perhaps the main benefit working with OSS is the connection with the users. The main difference between traditional proprietary software companies and OSS companies is that the end user may communicate to individual developers and even contribute directly to the source code. Even if the contact with the users is not always free of obstacles, all of the case companies stressed the importance of inputs from the community and customers. It is in fact a key issue for them that they have a functional connectivity to the, what used to be a demeaning term, “free rider”, since they may provide the firms with important information and bug reports.51

Table 7 below shows the revenue models found:

---

51 A free rider is a non-paying customer. In the traditional sense, they would use pirate licenses and have little contact with the vendor firm, while for OSS the non-paying customer may very well be a star developer in an OSS project. Since the term is no longer stigmatized, the users are encouraged to express their opinions about the software and report bugs if possible. However, there is no pressure on the end users to do anything aside from just using the software.
Table 7: Open source revenue models found.

<table>
<thead>
<tr>
<th></th>
<th>Distribution</th>
<th>Development</th>
<th>Consultancy services</th>
<th>Training</th>
<th>Hybrid licensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscription (yearly)</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Fee (per opportunity)</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Fee (hourly rate)</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Fee, (clip card etc.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proprietary license (per copy/seat/user)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Value added (hardware)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Value added (prop. software)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

6.2.3.3 Price model findings

The case companies present their pricing through the company web site, normally through an online shop. The company pricing is the official price towards the customers, but all the companies also applied adapted prices through partners or towards large or selected customers.

Among the cases, Cendio is the only company that did not offer prices or full downloads through its web site. The customers can download a limited test version, but in order to buy licenses or other services, the customers is referred to Cendio’s partners. This “clean” partner strategy is partly a result of the lack of resources for Cendio and partly a way to keep the partners happy. The latter may be essential for small players like Cendio, since they lack the resources to handle the large customers they have targeted.

In general, the pricing varies due to the size of the project concerning services needed. All four cases have their own sales force in addition to reselling partners who are entitled to tailor-make individual agreements.

The pricing models do not differ from proprietary software vendors, and is not a significant key factor for OSS companies.

6.2.3.4 Distribution models

For most of the cases, the distribution model differs very little, and is an important key factor. All of the companies studied prefer to distribute their software products through downloading from the Internet. The cost for distributing the software is therefore very low or even close to zero, raising the margin for the products even more. All of the studied companies work together with partners, and even if some of these partners offer the products on media as part of their value-added services, the majority still prefer to download the software themselves.

However, for Codeweavers, the differences between Linux and Mac OS us-
ers, where the latter prefers a more traditional distribution model through retail packages, the opportunity is too great to ignore. Even if the company still prefers to distribute its software through the Internet, it is willing to put in the extra time and effort needed in order to expand its market.
7 Conclusions

This chapter will reiterate the theoretical implications of the study, present the results of the analysis and make suggestions for further research.

7.1 Theoretical implications

In addition to the results of the cases, this thesis also aims to contribute to the research of OSS in two ways: first, by adapting the business model framework, and second, by using it to study the commercial applications of OSS.

7.1.1 The extended business model

The motivation for a new business model framework was to create an extended model for the analysis of OSS firms. In addition, the aim was to create an analytical framework that could be applied to other industries and purposes as well.

The new extended business model framework consists of three basic elements; marketing positions, operational platform and offerings. The latter element is the main contribution to the earlier model, but another important issue has been to refine the model, so it is easy to comprehend and to use. Figure 11 illustrates the new model.

![Figure 11: The extended business model.](image)

Using the new business model as a guide for the semi-structured interviews, a structural case analysis was created, where the questions and answers provided a “ready-made”, internal case analysis. These case analyses were then used for the cross-case analysis.

The new extended business model framework should prove a good tool for further analyses and even for cross-industry studies. However, it may be too
simple or crude for more elaborate case studies; even so, the potential for somewhat larger samples may prove its worth, and it has certainly been useful for this work.

7.1.2 “Business models” for OSS companies

One of the findings from the first stage of the study was that the companies used what Raymond (2001) called “direct sale-value models”, meaning charging the customers directly for the right to use software, in opposition to earlier research which favored the indirect sale-value, or service-oriented models. However, most of the other authors have mentioned the possibilities for this kind of revenue model, but not emphasized their use. The reasons for this could be what the legal and moral proposition hybrid licensing means for the community.

These commercial applications of OSS were frowned upon, not only by the community, but also by researchers, perhaps since they could signify a “retreat” from what OSS stands for in many cases, for instance regarding innovation, openness, and quality. However, as more companies have committed themselves to OSS, sponsoring projects with paid labor and resources, there has been a shift towards more acceptance for hybrid models.

The era when most of the programming was made by unpaid volunteers may be gone, since many of the larger projects, as well as some of the smaller ones, more often have a great number of paid developers.

The results also show that it is the failure of the purely service-based revenue models that have forced the case companies towards hybrid revenue models. All four have had some revelation or even threat to their existence that has pushed them towards new revenues. In addition, the more traditional OSS business models may have proved unsuccessful when it comes to scaling them to a larger company. The reason for this could be the recession for the IT industry in general after the millennium shift, but something triggered a change in the larger companies in regard of their business models. Red Hat, for example, moved away from the retail market and created the enterprise offering because of the low profitability of the former model.

In the present day, as long as the companies are honest about their involvement, and guaranteed free access to the source code for developers, the community seems to be more akin to accept commercial revenue models. Perhaps this is the real paradigm shift towards OSS 2.0?

One important issue that needs to be addressed is that how the term business model is used. All OSS research uses business model as a descriptive term concerning what firms offer and how they gain revenues. While this thesis has used the business model as an analytical tool, a true comparison between earlier descriptions and the results presented in here may look haphazard. Given this, Table 8 below may still shed some light.
<table>
<thead>
<tr>
<th>Table 8: OSS business models summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use-value funding models</strong></td>
</tr>
<tr>
<td>-----------------------------------</td>
</tr>
<tr>
<td><strong>Raymond</strong></td>
</tr>
<tr>
<td><strong>Krishnamurthy</strong></td>
</tr>
<tr>
<td><strong>Lerner and Tirole</strong></td>
</tr>
<tr>
<td><strong>West</strong></td>
</tr>
<tr>
<td><strong>Wichmann</strong></td>
</tr>
<tr>
<td><strong>Hecker</strong></td>
</tr>
<tr>
<td><strong>Perens</strong></td>
</tr>
<tr>
<td><strong>Dahlander</strong></td>
</tr>
<tr>
<td><strong>Goldman and Gabiel</strong></td>
</tr>
<tr>
<td><strong>Fitzgerald</strong></td>
</tr>
<tr>
<td><strong>Rosén</strong></td>
</tr>
</tbody>
</table>

- **Funding models**
  - Raymond: Cost Sharing, Risk-spreading, Loss-Leader
  - Krishnamurthy: Distributor (1)
  - Lerner and Tirole: Code release
  - West: Partly open
  - Wichmann: OSS development and Community Enablers
  - Hecker: Loss-Leader
  - Perens: End-User Business + Contractors + Service Businesses
  - Goldman and Gabiel: Bundle OSS with other software + Hardware
  - Fitzgerald: Contraction
  - Rosén: Contraction

- **Indirect sale-value models**
  - Cost-sharing: Loss-Leader
  - Risk-spreading: Give away the recipe...
  - Loss-Leader: Widget Flosting
  - Accessorizing: Free the future
  - Free the software, sell the brand: Thrid-party Service Provider (software and service)
  - Service Enabler: Dual licensing (Hybrid 1)
  - Proprietary: Open Source

- **Direct sale-value models**
  - Free the software, sell the content: Third-party Service Provider (software and service)
  - Service Enabler: Dual licensing (Hybrid 1)
  - Restricted modifications (Hybrid 2): Proprietary
  - Subscription services: Ownership of OSS
  - Using projects: Using projects
  - Using projects: Using projects
  - Using projects: Using projects
Red Hat has pioneered the OSS with a close-to-perfect service-based model, but technical, legal and commercial matters forces it to lock in some part of its offer in order to maintain profitability. I am sure some may argue that its revenue model, the enterprise subscription model, is not a hybrid, but fully legitimate. However, it impinges on the distribution rights of OSS, in that the SLA does not allow a customer to have other distributions covered by the offer. That is why I consider it a hybrid model.

Regarding Codeweavers, it would prefer to leave its hybrid model for a pure service-based model, helping Independent Software Vendors (ISV) to port their software from Windows to Linux or Mac OS X. Codeweavers’ current model is an economical necessity in order to survive.

Cendio, with its history of hybrid revenue models, is striving to become more open source. It is the only company in this study that does not benefit from an OSS project, and it had the least developing resources. For Cendio, it might be necessary to tap into the benefits an OSS project might give, be it marketing or development. This strategy change is a low-risk project, since its target customers, i.e. governmental, municipal and healthcare institutions, are not likely to become free riders. Instead, going open source may very well significantly increase the number of customers as well as other benefits, provided the software gets the attention it needs.

Finally, MySQL, which was recently acquired by Sun Microsystems, now faces a new phase. How will it cope as a part of a multinational corporation with a lot of other software? How will Sun position MySQL towards other OSS databases and partners such as Oracle? The people of MySQL have interesting times ahead.

7.2 Analytical results

It may very well be that hybrid licensing for OSS firms is a very viable revenue model, since the combination of a higher margin, in the form of less paid development, as well as the negation of traditional traps regarding service or product-oriented software business strategies, creates a wholesome business situation. The former service-oriented OSS business thus combines the benefits of high-end user contacts and user innovation together with a stable revenue stream.

There are many hurdles for a company to tap successfully into concerning both revenues and community participation. The first obstacle is legal. The firm has to retain the copyright for the source code in order to be able to use dual licensing or a proprietary license. The second obstacle is visibility. Three of the four case companies have an OSS community of developers and users that gives the brands visibility and resources to utilize.

The first obstacle means that the company has to be in control of how code contributions are handled, by either licensing, asking for copyright transfer or hiring the developers. Moreover, the second obstacle restrains the com-
pany, which is why the first option is not used by any of the case companies, since it may hamper the contributions; instead, different variations of the two last options have been utilized. For Cendio, which is the only company lacking a significant OSS project, the strategy for open sourcing its software has not yet been decided, but it is likely to be any or both of the latter two options.

Moreover, the studied companies show us that hybrid licensing and revenue models are in use, and that they are, in most cases, successful. Red Hat and MySQL have both proven the worth of their hybrid revenue models. However, their success also shows us that in this case, size really matters! The measure of the success seems to be dependent not only of the size of the company, but also the size of the community. The larger the community, the larger the company can become and thus attract more customers and by doing that, enlarging the community.

Balancing the OSS community is important from another standpoint, namely recruitment. The case companies, regardless of their size, have very competent programmers. In most cases, the developers are recruited directly from the projects, based on the quality of their contributions. It seems like being connected to OSS could be enough to attract programmers, however, since even Cendio has managed to recruit very skillful developers. This is of course a significant competitive advantage to most other software companies.

7.2.1 Key factors for OSS software product companies

Based on my findings, I have tried to create a model that illustrates the key elements and factors for OSS software product companies, as shown in Figure 12 below:

![Figure 12: Key factors for successful software business development.](image)

The three key elements for a successful OSS software development company
are the offered product and services, the OSS projects and the revenue model. All the elements are interconnected with each other, but are also affected by eight key factors, each of which influences the rate of success. They are brand for the product, i.e. the company and the OSS project; community, that is the sum of the non-paying users and developers connected to OSS projects; and resources, which refer to community-based resources such as development and testing. There are also legitimacy, the perceived legitimacy regarding licenses and the revenue models; control, the control the firm has of the software; and ability to charge, how the company can charge for its services. The last two factors are customers, the paying users, and the volume, the number of paying customers.

The OSS Project(s) denotes the community-driven OSS projects in which the company is active. Two of the companies, Codeweavers and MySQL, are primarily active in a single project, while Cendio and Red Hat are active in several each. The OSS Project(s) can provide resources, are affected by the brand of the company and its products, the community itself and legitimacy regarding licenses and control (IPR).

The offered Product/Services are affected by the brand, the community and customers that build up the volume of the sales. There is no connection between resources and the offering because primarily internal resources are used to make the product; the legal issues surrounding licenses and the content of the products plays a part in this.

The revenue model is the actual way the company charges for the offered Product/Services. It the element that is affected by most factors, such as customers, volume, ability to charge, legitimacy and control. All of these factors influence the revenue model and may limit the way the company is able to gain revenues.

Aside from connecting to the main elements, the factors are also interconnected to each other. The perhaps most important key factor is the brand. The brand can consist of more than one part, and a company may have one brand for the company, one for the product and one for the OSS project. However, the brand affects both the customers and the community reciprocally. A known brand grows the community and number of customers, and a large community and more customers grows the brand. This means the company is able to raise its own brand by marketing towards customers or by marketing towards the community or both. This is how MySQL and Red Hat operate. Another way to grow the brand is to connect the company to an OSS project, and when the community grows, get a free ride for its own brand. This is how Codeweavers operates, and how Cendio is trying to. However, if the project is too small, the brand awareness may not penetrate outside of the community, which is why Codeweavers has struggled to compete with some of its competitors.

Going on the lower side of the model, the customers affect not only the
brand, but also the volume of sales. The more paying customers a company has, the larger volume of sales it gets, in addition to higher brand awareness. The paying customers may very well be a part of the OSS project, and play a part of the community resources. Nevertheless, this is not generally the case.

The revenue model also affects to the volume, which is connected to the Product/Services. If the strategy for the company is low-cost, for example, this will raise the volume of sales, while a high-cost strategy usually diminishes it. The volume of sales may also affect the OSS project since a large volume will attract more attention to a project, despite the attention coming from paying customers. The effect may be downloads or browsing through online support documents, even if these services are provided by a vendor. The volume directly affects the firm’s ability to charge, insofar that a greater volume may increase the confidence of the company to charge its customers.

Turning back to the brand and going to the top side of the model, the size of the community directly affects the amount of resources the OSS project can utilize. Even if the community, in this form, is the non-paying users, the community does affect the products or services offered, mainly through communication; if the company listens to the community, they can convert some users into customers.

Looking at the resources, these mainly affects the OSS project, but if the community feels that the legitimacy of a company is “dodgy”, they may very well stop contributing as in the case of the Codeweavers’ competitor Cedega. The resources consist of not only developers, but the mainstay of active users that bug-test the software and make requests for improvements.

The legitimacy is strongly connected to the OSS projects, and concerns the perceived legal view of the software, mainly based on the community’s point of view. It is being affected by the revenue model and may be connected to control (IPR).

The key factor control (IPR) is special, since it may be connected to legitimacy; this only happens if the company has retained the copyright of the source code and this is accepted by the community, as in the cases of MySQL and Codeweavers. Is in these cases, the control gives the companies more ability to charge their customers. A company may very well own other forms of patents or copyrights not connected to the main OSS project, such as in the case of Red Hat, but OSS companies usually make a “patent promise” to calm the community down. The control is generally applied to the revenue model, but also the OSS project, insomuch regarding who owns the copyright of the primary OSS project.

The ability to charge key factor is, as stated above, connected to the confidence of the company regarding how to charge. The control factor may enable the firm more options on how to charge, such as dual licensing or proprietary licenses, but innovative companies such as Red Hat have managed
to create new hybrid revenue models based on their confidence and ability to charge.

7.2.1.1 The most important factors

Summarizing, the most important key factor is without doubt the *brand*. It will not only determine the number of customers, but also affect the size of the community. A firm interested in selling OSS-based software products should pay most attention to brand issues, since most of the other factors and elements are dependent on it.

The other important factor, one which can affect the company financially, is the matter of *legitimacy*. If the community feels betrayed or conned, the resources may very well dry up, which means that the competitive advantage the firm would have over its competitors would decline.

The third and final important factor is the *control*. If the company can retain the copyrights of the source code of its most important software, its ability to create revenue streams increases. The ability to offer service-based contracts in addition to proprietary licenses must be attractive to most companies; however, it must be balanced with the brand and legitimacy factors in order to be fully effective.

7.2.2 Closing discussion

Considering how the brand affects the success of an OSS company, it should be noted that the OSS projects from both MySQL and Red Hat have existed for some time. The rise of popularity has taken a long time, before the companies fully could benefit from it. In addition, how the various brands are managed could be a decisive factor. Both the larger companies, MySQL and Red Hat, have tried to merge the different brands for the OSS project, the product and the company name into one. For example, MySQL is the common brand for all three items, while Red Hat chose to rename its OSS project to Fedora to “liberate” it from the Red Hat brand in order to promote community sharing and contribution. This is something neither Cendio nor Codeweavers have done. However, this option may not be open for everyone; perhaps MySQL is unique in this matter. You could also argue that Codeweavers’ strategy with three different brands is consistent with its original business mission in addition to the community reaction if the product was branded Codeweavers’ Wine.

In addition, this also raises the question of whether it is possible to go “the other way” by using a strong brand name and resources and create a new community. Several items indicate that this could be possible. During the first part of this research, IBM and Sun Microsystems were investigated. These corporations, which operate worldwide and have strong brand names, decided to become active in the OSS community. Even if their attempts were met with skepticism at first, they have since been accepted. Earlier research also confirms that commercial corporations are able to gain support in the
community and benefit from it (cf. Hecker, 1999; Goldman and Gabriel, 2005).

It would be very interesting to study how companies active in the OSS community balance their brands, legitimacy and control in order to contribute and benefit from the community participation.

7.2.3 Future challenges
The large case companies now have different challenges to overcome. Red Hat needs to continue to be profitable, but at the same time not alienate the community. If it loses the support of the community, its current competitive advantage will fast become moot. MySQL needs to be able to work inside a much larger corporate body and at the same time keep its ties to the community. In this case, Sun seems to have been the optimal buyer, since the corporation has been working hard to gain the trust of the community. In all, the two large case companies seem to be on the right track and the future looks quite bright.

Our two smaller firms also have their work cut out for them. Codeweavers needs to be able to break into the burgeoning Mac OS market, where it will battle with larger competitors with deeper marketing wallets. Cendio, on the other hand, needs to grow its market share in order to become profitable, and at the same time find its way back into the community in order to tap into the marketing and development possibilities. Cendio and Codeweavers are in an awkward position and needs to find ways to gain momentum. If they can handle the challenges well enough, they may very well be in the position of Red Hat and MySQL in a couple of years.

7.3 Related research
With a study like this, other interesting possibilities sometimes present themselves. When we studied some of the case companies, we noticed patterns in how their business models changed. In order to understand how a firm works and evolves over time, path dependencies and change patterns may be key components to investigate.

Even with a brief investigation, we were able to establish two sets of patterns of change. The findings seem to indicate that there are a limited number of patterns that such changes have, and that some patterns that occur are dependent on internal or external trigger events. A focused study may shed additional light on these events.

7.4 Recommendations for further research
A natural way of testing the viability of the new business model framework would be an extended study, encompassing more companies and other industries. The first item of interest would be to make an industry-wide investigation regarding the key factors found in this thesis. A clear result from a
study like that could help entrepreneurs and investors to understand and find viable strategies for success in the OSS industry.

Another interesting option would be to test whether the framework stands up when it comes to cross-industry research. If the same framework could be used throughout several industries, perhaps key factors that differ or connects different industries could be identified, and used to further our knowledge about the business world as a whole.

One area that was somewhat lacking in this research regarded partnership. What are the key elements for a successful partnership? There were indications that at least some cultural aspects may be part of the mystery, but was not in the scope of this study. It would be interesting to see whether the demands of OSS companies differ from the “normal” IT-industry, however.
8 References

Literature


Henkel, J. (2003). Open Source Software from Commercial Firms – Tools, Comple-
159-178.


sand Oaks.

**Miscellaneous**


<http://computersweden.idg.se/2.2683/1.105492> (2007-11-14)


Fedora (2007). Fedora Overview. (Electronic) Available:


<http://www.theregister.co.uk/2001/06/02/ballmer_linux_is_a_cancer/> (2007-08-17)


Wikipedia (2007c). *Tactic (method).* (Electronic) Available:


**Interviews**

Axmark, David, Founder and Vice President, MySQL AB, February 28, 2006 and October 12, 2007.

Mickos, Mårten, CEO, MySQL AB, October 28, 2005

Moosberg, Peo, CEO, Cendio AB, March 10, 2006

Wallin, Inge, CTO, Cendio AB, March 10, 2006
Appendix 1 – Swedish interview guide (1)

Bakgrund
- Hur länge har du arbetat på Företaget?
- Hur länge har du arbetat i den rollen som du nu innehåller?
- Var arbetade du tidigare?
- Har ditt arbete förändrats under det senaste året?
- Hur ser du Företagets roll i förhållande till era kunder/andra bolag?
- Hur har Företaget ändrats genom åren?
  - Organisation
  - Kultur
  - Produkter
  - Erbjudanden
  - Strategi
  - Kunder

Framtid/Vision
- Vad ser du som Företaget affärsidé/modell?
- Vad ser du som Företaget kärnkompetens?
- Kan du berätta var du ser Företaget i framtiden (i förhållande till t ex kunder, dig själv etc)?
- Vilken vision har du i förhållande till detta?
- Har ni diskuterat framgångsfaktorer? Vilka?
- Hur ser du på öppen källkod och dess framtid?
- Identifierar du Företaget som ett öppen källkods-företag?
- Öppen källkod jämfört med proprietära program?

Organisation/Struktur
- Skulle du kunna beskriva organisationen utgående från din position?.
- Vem ansvarar för produktutveckling och identifiering?

Roll
- Är du klar med vilken roll och vilket ansvar du har?
- Tycker du att det är svårt att se var ditt ansvar slutar och andras börjar?
- Hur interagerar din roll/du/din uppgift med andra roller? Med andra delar av organisationen?

Teknologi
- Var det så att tekniken möjliggjorde sådant som ni funderat på eller drev tekniken nya lösningar?
- Vilka fördelar respektive nackdelar har öppen källkod?
- Vilka karaktäristika har öppen källkod som gör det lön-samt/intressant?

Erbjudanden
- Kan du beskriva ett erbjudande?
  - Vad är det?
Vad innehåller det (t ex service produkt)?
Vem skapar nya erbjudanden?
- Hur säljer ni/marknadsför ni era erbjudanden (eller produkter)?
- Hur resonerade ni runt era erbjudandena?
- Vilka är de element i era erbjudanden som du anser är mest intressanta, värdeskapande etc jämfört med "vanlig" programvara?
- Hur genomfördes erbjudandeidentifieringen?
- Hur hanterar ni licensieringsproblematiken?
- Hur resonerar ni kring era intäktsströmmar?

**Process**
- Har ni tydliga processer? (Vilka och vad?)
- Hur skulle du definiera en process?
- Hur identifierar ni era kännprocesser?
- Vilka processer finns du med i?
- Hur ser inlärningsprocessen ut?

**Kunder/Leverantörer**
- Hur identifierar ni era kunder?
- Känner du att ni är en kundfokuserad organisation?
- Vilka processer har ni med (i samarbete) kunder?
- Vilket förhållande har ni med era leverantörer?
- Hur och vilka hanterar (vårdar) era kund- respektive leverantörsrelationer?
- Beskriv hur kunden når er organisation?
- Hur fångar ni kundens behov?
- Särskiljer ni på stora respektive små kunder och i så fall hur?
  - Segmentering
  - Positionering
- Kan du identifiera var och hur ni skapar värde för era kunder?
- Vilka idéer har ni i framtiden för kunder respektive leverantörer?

**Anställda/resurser**
- Vilken "sorts" anställda har ni?
- Hur är mixen gammal/ung bland anställda?
- Hur fördelas resurser?
- Ser du några problem med resursfördelningen?
- Vilka resurser finns respektive finns ej inom företaget?
- Var ligger fokus: produkt eller försäljning?
- Hur säljer ni?

**Ledarskap/Kommunikation**
- Vilken roll anser du att ledningsgruppen har?
- Hur förankras beslut i organisationen?
- Hur kommuniceras direktiv uppför?
- Hur sker kommunikation horisontellt, mellan både individer samt funktioner?
Övrigt

- Support system
- Mätning:
  - Hur mäter ni kundnöjdhet?
  - Hur hanteras återkoppling till berörda?
- Nytt värdeskapande:
  - Hur diskuteras detta?
- Kultur
- Vilka konkurrenter har ni?
- Vad är era konkurrensfördelar?
  - Jämfört traditionella konkurrenter?
  - Jämfört OSS-konkurrenter?
- Hur positionerar ni er på marknaden?
Appendix 2 – English interview guide (2005)

Internal strategic issues
1. Please describe your business model regarding OSS.
2. How important are OSS for your company today? (In terms of revenues, marketing, research and development or other.)
3. What were the driving forces behind your move to/towards OSS? (Internal – external, primary, secondary, tertiary etc.)
4. How are you working with OSS strategically today? (Organization, management, development etc.)
5. Does your product development of OSS differ from your proprietary software?
6. How do you decide which project(s) to be OSS and which project(s) to be proprietary? (Like IP, patents, third parties IP and patents, etc.)
7. Is there any softwares (owned by your company) you think is unlikely to be revealed (if any)?

External strategic issues
1. What do you think is the greatest threat to OSS (if any)? (software patents, licensing, marketing campaigns.) Please answer both generally and specific for your company (if possible).
2. Some claim that OSS will be the norm of all software in the future? Is this reasonable, plausible or even desirable?
3. How do customers respond if you offer them solutions based on OSS?
4. What is your opinion of the competitive advantage of OSS (if any)? Please answer generally and specific regarding your company (if possible).
5. How do you perceive the service of OSS?
6. Is there any market that you see have greater opportunities for OSS than others? Please answer both generally and specific for your company (if possible).

Industry formation
1. Policies and standards are are starting to emerge regarding OSS. Does this affect your company in terms of strategic assessments?
2. The formation of open standards is one part of the policies. Can you describe how your company work with open standards (if possible)?
3. Some countries, like China and South Africa, implements policies to use OSS (mainly) in order to strenghten local software development. How do you think it might affect your company? Please answer both generally and specific for your company (if possible). (Short and long term)
Appendix 3 – Interview guide 2007

Marketing/sales
- What is your business idea/mission?
- What is your outstanding mission? (Overall, per department)
- What are your goals? (Short term + long terms)
- Do you have a branding strategy? If so, how is it formulated? (Towards paying customers and community)
- Who are your competitors? Biggest threat (of all) OSS competitors etc.
- Describe your corporate culture. (Does it differ depending on department?)
- Name some of your official customers. (The ones you CAN mention).
- How do you market/promote your products? (Commercial/non-commercial)
- Name your most important partners? (Mention in what way they are important if possible)

Operational/development
- How do you manage (do you?) your in house development? (Version handling, etc.)
- Does your management differ between commercial and non-commercial development?
- Do you have any significant developer not employed by you working with your projects?
- What are your experiences regarding revenue models and development management?
- Describe your corporate history. (Founding, business etc.)
- Do you own patents, copyrights and if so, how are they defended?
- What kind of agreements do you have? (EULA, partnership, certification etc.)
- How is the company organized? (Departments etc.)

Product/service offering
- How is a customer informed regarding price and offerings etc?
- Describe your products and services. (What do you offer your customers?)
- How do you charge the customer? (Revenue models, licensing, subscriptions etc.)
- How is the software distributed to the customers? (Download, media etc)