Open Source Strategy: A Change of Perception through the Lens of Innovation

The Case of Open Source Software (OSS) in Sweden

Master Thesis within Business Administration

Authors: Celal Can Bilen
Zahra Alavizadeh

Tutors: Anders Melander
Duncan Levinsohn

Date: March 2011
Förord

Därför att vi båda ursprungligen är ingenjörer inom inbyggda system, skulle vi vilja skriva vår uppsats om något som skulle kombinera vår tekniska bakgrund med våra nya kunskaper inom företagsekonomi som vi lärt känna under vår utbildning på Internationella Handelshögskolan (IHH) i Jönköping. Således kom vi i första hand med "öppen källkod strategi", och med tanke på att vi skriver uppsatsen i Sverige, beslutat vi att uppsatsen skulle undersöka öppen källkod strategi bland svenska företag som arbetar med öppen källkod utveckling.

Vi skulle vilja tacka våra handledare, Prof. Anders Melander och Duncan Levinsohn, från Internationella Handelshögskolan (IHH) i Jönköping, för att de styrde oss genom avhandlingen och utvecklingsprocessen, delade med sig viktiga och värdefulla förslag och idéer om hur avhandlingen kunde förbättras ytterligare.

Vi skulle också vilja tacka alla våra kontakter för att de vänligt accepterat vårt intervjuförslag och gett oss värdefull och intressant information som har hjälpt oss forma vårt examensarbete och för att de gav oss inblick i deras företag.

Celal Can Bilen & Zahra Alavizadeh

Förutom de som har nämnts, vill vi också tacka följande individuellt:

Två år har gått ganska fort, och jag har haft möjlighet, tack vare den här uppsatsen, att arbeta inom ett område som jag alltid har varit intresserad av. Jag känner mig tackbar för haft möjligheten att arbeta med ett så intressant och populärt ämne, genom att kombinera två områden, teknik och företagsekonomi. Jag skulle vilja tacka min familj för deras ständiga stöd ända sedan jag först hamnade i Sverige, med alla upp och nedgångar genom åren.

Celal Can Bilen

Jag vill tacka min familj för deras oskiljaktiga stöd, uppmuntran, tyst tålamod och orubblig kärlek som de har gett mig genom åren och att de har låtit mig vara så ambitiös som jag ville. Utan deras hjälp skulle den här uppsatsen inte ha varit möjlig.

Zahra Alavizadeh
Preface

Because we are both originally embedded systems engineers, we’d wanted to write this master thesis about a topic that would combine our engineering background with our new background of business administration, which we’d had the opportunity to come to know during our education at Jönköping International Business School (JIBS). Hence, we came up in first place with "open source strategy", and based on the fact that we are located in Sweden, we decided that the thesis would be an investigation of open source strategy among Swedish companies working with open source development.

We would like to thank our supervisors, Prof. Anders Melander and Duncan Levinsohn, from Jönköping International Business School (JIBS), for guiding us throughout the thesis development process, sharing with us important and valuable suggestions and their ideas about how the thesis could be improved further.

We would also like to thank all our contacts for warmly accepting our interview proposals and giving us valuable and interesting information which helped us shape our master thesis and gave us the insight to their companies.

Celal Can Bilen & Zahra Alavizadeh

In addition to those who have been mentioned, we would also like to thank the following individually:

I would like to thank to my family for their inseparable support, encouragement, quiet patience, unwavering love that they have provided me over the years and allowing me to be as ambitious as I wanted. Without their help this thesis work would not have been possible.

Zahra Alavizadeh

Celal Can Bilen
# Table of Contents

**Abstract**................................................................................................................ vii

**Sammanfattning** ..................................................................................................... vi

**Preface** ................................................................................................................... ii

**Förord** ..................................................................................................................... i

1 Introduction ............................................................................................................. 1

1.1 Definitions .......................................................................................................... 1
  1.1.1 Open Source (OS)/Open Source Software (OSS) .............................................. 1
  1.1.2 Free Software .................................................................................................. 1
  1.1.3 Free and Open Source Software (FOSS) ............................................................... 1
  1.1.4 Proprietary Software/Closed Source ................................................................. 1
  1.1.5 Commercial Software ...................................................................................... 2
  1.1.6 Strategy ............................................................................................................ 2
  1.1.7 Business Model ............................................................................................... 3
  1.1.8 Licensing .......................................................................................................... 3
  1.1.9 Strategy vs. Business Model ............................................................................ 3
  1.1.10 Intellectual Property (IP) ................................................................................ 4
  1.1.11 Research and Development (R&D) ................................................................. 4
  1.1.12 Community .................................................................................................... 4
  1.1.13 Networking .................................................................................................... 4
  1.1.14 Corporate Venturing ...................................................................................... 5
  1.1.15 Openness ....................................................................................................... 5
  1.1.16 Skill-set Analysis ........................................................................................... 5
  1.1.17 Maturity Analysis .......................................................................................... 5
  1.1.18 Productization ............................................................................................... 5

1.2 Background ......................................................................................................... 6

1.3 Purpose .............................................................................................................. 8

1.4 Delimitations ..................................................................................................... 9

1.5 Disposition ........................................................................................................ 9

2 Thesis Plan ........................................................................................................... 11

3 Innovation Strategy ............................................................................................ 13

3.1 Background ..................................................................................................... 13

3.2 Internal Components ....................................................................................... 13
  3.2.1 Business Model .............................................................................................. 14
  3.2.2 R&D Management ......................................................................................... 15
  3.2.3 IP Management ............................................................................................. 16
  3.2.4 Innovation Roadmap ....................................................................................... 17

3.3 External components ....................................................................................... 17
  3.3.1 Corporate Venturing ..................................................................................... 17
  3.3.2 Communities ................................................................................................. 17
  3.3.3 Networking ................................................................................................... 19

3.4 Firm-specific components ............................................................................... 20
  3.4.1 Innovation Processes ...................................................................................... 20
  3.4.2 Innovation Culture ......................................................................................... 20
  3.4.3 Firm size ........................................................................................................ 20
  3.4.4 Complexity .................................................................................................... 21
# Table of Contents

3.5 Summary ........................................................................................................ 21

4 Open Innovation .................................................................................................. 23
  4.1 Introduction ................................................................................................... 23
  4.2 “Openness” of the Innovation .......................................................................... 23
  4.3 Open Source as Open Innovation .................................................................... 24
  4.4 Summary ........................................................................................................ 28

5 Open Source Software ........................................................................................ 29
  5.1 Licensing ........................................................................................................ 29
  5.2 Open Source: Firm Perspective ........................................................................ 30
  5.3 Open Source in Europe ................................................................................... 32
  5.4 Open Source in Sweden ................................................................................... 33
    5.4.1 Open Source Sweden ...................................................................................................... 35

6 Open Source Strategy .......................................................................................... 36
  6.1 Focus on Products: The Licensing Strategy ....................................................... 37
    6.1.1 The GNU GPL or Copy-left License ................................................................................... 38
    6.1.2 The Lesser GPL (LGPL) License ......................................................................................... 39
    6.1.3 The Berkeley Software Distribution (BSD) License ............................................................. 39
    6.1.4 Dual and Multi Licensing ................................................................................................... 39
    6.1.5 Versioning ........................................................................................................................... 40
    6.1.6 Choosing the Right Licensing Strategy ................................................................................ 40
  6.2 Focus on Services/Deployment: Open Source Business Models .......................... 41
    6.2.1 The Distributor (Support Seller) Model .............................................................................. 41
    6.2.2 The Non-GPL (Added-value Provider) Model .................................................................... 41
    6.2.3 The Pure Open Source (GPL) Model .................................................................................. 42
    6.2.4 The Third-Party Support Seller Model ................................................................................ 43
    6.2.5 The Platform Provider Model ............................................................................................. 43
    6.2.6 The Software-as-a-Service (SaaS) Model ............................................................................ 43
    6.2.7 The Accessorizing Model ..................................................................................................... 43
    6.2.8 The Loss Leader Model ....................................................................................................... 43
    6.2.9 Hybrid Models .................................................................................................................... 44
  6.3 The R&D Management Strategy ...................................................................... 44
  6.4 The IP Management Strategy ........................................................................... 45
  6.5 The Networking Strategy ................................................................................. 46
  6.6 Forming the Open Source Strategy ................................................................... 46
    6.6.1 Skill-Set Analysis ................................................................................................................. 48
    6.6.2 Maturity Analysis ................................................................................................................ 49
    6.6.3 Open Source Adoption: Productization .............................................................................. 50
  6.7 Summary ........................................................................................................ 50

7 Methodology ....................................................................................................... 52
  7.1 Deductive vs. Inductive Research ..................................................................... 52
  7.2 Qualitative Data Collection ............................................................................. 52
  7.3 Validity .......................................................................................................... 54
  7.4 Reliability ....................................................................................................... 55
7.5 Empirical Findings ........................................................................................................ 56

8 Analysis ............................................................................................................................. 62

8.1 Individual Analysis of the Firms .................................................................................. 62

8.1.1 Firm A ......................................................................................................................... 62
8.1.2 Firm B ......................................................................................................................... 63
8.1.3 Firm C ......................................................................................................................... 64
8.1.4 Firm D ......................................................................................................................... 65
8.1.5 Firm E ......................................................................................................................... 66
8.1.6 Firm F ......................................................................................................................... 67
8.1.7 Firm G ......................................................................................................................... 68
8.1.8 Firm H ......................................................................................................................... 69

8.2 Overall Analysis of the Firms .................................................................................. 70

8.2.1 Firm Background with Open Source ......................................................................... 71
8.2.2 In-house components of the open source strategy ...................................................... 72
8.2.3 Business model components of the open source strategy ........................................ 74
8.2.4 Innovation characteristics of the open source strategy .............................................. 75
8.2.5 Open source development scene in Sweden .............................................................. 77

8.3 Summary ...................................................................................................................... 78

9 Conclusions, Reflections and Future Research ................................................................. 80

9.1 Conclusions ................................................................................................................... 80
9.2 Reflections ..................................................................................................................... 81
9.3 Suggestions for Further Research ................................................................................ 82

10 References ..................................................................................................................... 83

11 Appendix: Interview Structure ....................................................................................... 88
Sammanfattning

Öppen programvara har handlat om myter fram till nyligen, många människor har ju trott att det inte är tillförlitligt, eftersom open source-projekt är skrivna av en liten grupp av amatörer i ”sin väns garage”. Sådana myter har tillbakavisats med den framväxande framgången och ökade populariteten av öppen källkod, men många ansåg fortfarande öppen källkod och proprietära programvaror som fiender som aldrig kunde samexistera i en företagsplattform. Således har vissa företag valt att hålla fast vid sin tradition av kommersiell mjukvaruutveckling, medan nya företag som baserar sin företagsstrategi enbart på öppen källkod har dykt upp. Den tidigare gruppen har förlorat tid och pengar på diagnos och felsökning efter buggar i mjukvaran, medan den senare gruppen har visat sig ha svårt att hitta ekonomiskt stöd och ta marknadsandelar i konkurrens med den tidigare gruppen. Inte någon av grupperna har funnit absolut framgång i sin verksamhet, följaktligen har en kompromissmodell vuxit fram inom mjukvaruindustrin, vilket resulterade i en tredje grupp av företag som varken arbetar helt med öppen källkod eller helt med kommersiella modeller, utan istället jobbar med en hybrid som möjliggör integrering av dessa så kallade ”två fiender”. Sådana företag har övervunnit fallgröpar i båda tillväxtagångssättet och lyckats kombinera fördelarna med dem. Den växande framgången för integrationen av öppen källkod ledde till att allmänheten blev uppmärksam på den potentiella betydelsen av öppen programvara. Stora kommersiella IT-företag har också börjat integrera öppen källkod i sina kärnstrategier. Trots det utbredda tvivel och motstånd under de tidiga stadierna av öppen källkodrevolution, har det nyligen skett en ”förändring av perception” inom mjukvaruutvecklingsindustrin för att integrera strategier för öppen källkod i sina affärsmodeller.

Det här uppsatsen fokuserar på öppen källkodutveckling i Sverige genom att undersöka svenska företag som utvecklar mjukvaruprodukter, antingen baserade på komponenter av öppen källkod (hybridmodell) eller utvecklar produkter helt med öppen källkod. Efter en analys av marknaden på basis av affärsmodeller för dessa företag, undersöker uppsatsen ytterligare olika strategier för öppen källkod som bedrivs på dessa företag samt deras sammansättning. Vår uppsats slutar med en analys av scenen för öppen källkod i Sverige, vilket vi använder för att bestämma kännetecken av mjukvaruindustri i Sverige.

Våra slutsatser visar att strategier för öppen källkod delar de flesta av dess komponenter med strategi för innovation (vissa komponenter visar sig dock vara specifika för öppen källkodstrategi), vilket bekräftar sambandet mellan öppen källkod och innovation. Därför måste de företag som arbetar med öppen källkod uppmärksamma innovation och skapa lämpliga strategier för innovation och öppen källkod om de vill bli framgångsrika inom mjukvaruindustrin.

Nykkelord: Öppen källkod, proprietär programvara, affärsmodell, öppen programvara, strategi för öppen källkod
Open Source Software has been all about myths until recently, with many people believing that open source isn’t reliable because the open source projects are held by a small group of amateurs in their friend’s garage. Such myths have been refuted with the emerging success and increased popularity of open source, but still many considered open source and proprietary software to be enemies, which can never co-exist in the corporate platform. As a result, some firms have decided to stick to their tradition of commercial software development, while new firms, which base their corporate strategy solely on open source, have emerged. The former group of firms has suffered from losing time and money in fixing the errors and bugs in the software, whereas the latter group has found difficulties in finding financial support and market share in competition among the former group.

Neither group has found absolute success in their business, as a result a compromise model has emerged in the software industry, which resulted in a third group of firms that work neither with pure open source model nor with pure proprietary models, but instead with “hybrid” business models which allows integration of these so-called two enemies. Such firms have overcome the pitfalls of both approaches, while combining the benefits of them. The growing success of open-source integration attracted further public attention on the potential importance of open-source software (OSS). Indeed, major large commercial IT companies have started to integrate open-source software into their core strategies. Despite the widespread doubts and resistance during the early stages of open source revolution, there has recently been a “change of perception” in the software development industry towards incorporating open source strategies into their business models.

This master thesis investigates the open source software scene in Sweden by examining Swedish firms that develop software products either based on open source components (hybrid model) or just open source products (pure open source model). After analyzing the market based on the business models of these firms, further details of the open source strategies pursued by these firms have been analyzed. Our study then finalizes the investigation with an analysis of the open source development scene in Sweden, which we use to determine the characteristics of the software industry in Sweden.

Our conclusions reveal that open source strategy shares most of its components with innovation strategy (with some components found to be specific to open source strategy only), confirming the link between open source software and innovation. Therefore, firms that work with open source software have to pay attention to innovation and form an appropriate innovation and open source strategy if they want to be successful in the software industry.

**Keywords:** Open source software, proprietary software, business model, open source strategy, open source development
1 Introduction

1.1 Definitions

This section summarizes the most important terms used throughout this thesis which would give the reader an early clarification of the terminology used in the open source world.

1.1.1 Open Source (OS)/Open Source Software (OSS)

The term “open source” was coined in 1997 to refer a category of software where the source code is freely distributed (West & Gallagher, 2004). Open source software (OSS), or just open source in short-hand, refers to the type of software whose source code can be viewed, modified and redistributed to others for free, while acknowledging the original author’s contribution. This strongly contradicts with proprietary software where developer firms aggressively protect their software source code.

1.1.2 Free Software

In popular usage, the term “open source” actually overlaps (and largely subsumes) the “free software” category, however, the “free” software contains IP restrictions intended to force sharing of any derivative works, while other forms of “open” software (such as the Apache license) allow private commercialization of related innovations (West & Gallagher, 2004). Nonetheless, open-source software is still “free” in that no license fees are charged for use or redistribution of binaries or source code, and users are free to modify the source, create derivative works, and distribute those works (Hecker, 1999). Unlike open-source software products, free software products can prevent access to source code, even though the access to the software is still free. Yet, firms can still pursue open-source strategies related with these products, especially through by-products such as product maintenance and support.

1.1.3 Free and Open Source Software (FOSS)

The “Free and Open Source Software (FOSS)” concept covers both free software and open source software which, despite describing similar development models, have differing cultures and philosophies. Free software focuses on the philosophical freedoms it gives to users while open source focuses on the perceived strengths of its peer-to-peer development model. FOSS is a term that can be used without particular bias towards either political approach (Feller, Fitzgerald, Hissam & Lakhani 2005).

1.1.4 Proprietary Software/Closed Source

Software that is covered by copyright along with contract law, patents, and trade secrets that provide them the legal basis as owners of the software to establish exclusive rights on the product. Common examples of such exclusive rights are the restriction of the inspection of source code, modification of source code, and redistribution by the firm. Such firms determine the specific terms of use in an end-user license agreement (EULA) and usually limit the number of computers the software can be used and prohibit the user from installing the software on additional computers. Restricted use is sometimes enforced through a technical measure, such as product activation, a product key or serial number, a hardware key, or copy protection. They can also distribute versions that remove particular features, or ver-
sions which allow only certain fields of endeavor, such as non-commercial, educational, or non-profit use. Variation and extent of such restrictions vary by the EULA. This scheme is often referred to as *closed source*, the source code in this development model is regarded a trade secret of the company and is “hidden” from the users. The source code thus may include defects or malicious features which would compromise sensitive information which can be acceptable and recoverable in personal use, but rather critical and dangerous in business operations.

### 1.1.5 Commercial Software

Software that is designed for sale to serve a commercial need for the firms that develop, sell, support, and customize the software. Commercial software is usually proprietary software with “having profit as a chief aim”\(^1\); but in some instances it may be public-domain software\(^2\), i.e. software that lacks copyright protection. That is to say, commercial software can be either free or proprietary.

Commercial software is incorrectly used interchangeably with proprietary software. Phrases such as "Free Software", "Open Source Software", or "Proprietary Software" are totally independent of whether or not the software is commercial or not. Software can be licensed with a specific license, and yet have profit as a chief aim, such as with Red Hat Software. There is also going to be proprietary software which is given away for free (Internet Explorer and Netscape are two popular examples) where profit is not the aim of the software, but other motivations such as market control or advertising or Internet Portals\(^3\).

### 1.1.6 Strategy

Strategy is the direction and scope of an organization over the long-term: which achieves advantage for the organization through its configuration of resources within a challenging environment, to meet the needs of markets and to fulfill stakeholder expectations (Johnson, 2006). However, it is *not* tactical decisions such as setting prices or choosing suppliers.

Strategies exist at several levels in any organization - ranging from the overall business (or group of businesses) through to individuals working in it. Corporate Strategy ("Mission Statement") is a set of management decisions, for enabling an organization to achieve and sustain superior overall performance and returns. It is a core responsibility of senior executives and encompasses a range of critical activities, from defining and refining corporate vision to strategic performance measurement and management\(^3\). Business Unit Strategy entails knowing what business you are (or should be) in, understanding the current and future sources of competitive advantage in that business, and then defining a plan to capture and sustain an unassailable relative advantage over competitors\(^4\). Finally, Operational Strategy is concerned with how each part of the business is organized to deliver the corporate and business-unit level strategic direction, focusing on issues of resources, processes, people etc.

---

1.1.7 **Business Model**

Business model is the plan implemented by a company to generate revenue and make a profit from its operations\(^5\). The model includes the components and functions of the business, as well as the revenues it generates and the expenses it incurs.

A business model draws on a multitude of business subjects such as economics, entrepreneurship, finance, marketing, operations and strategy. The business model itself is an important determinant of the profits to be made from an innovation\(^6\), determining the revenue streams of the innovation for the firms.

A business model performs two main functions: creating value through a series of activities from the resources to the end product, and capturing a portion of that value, through establishing unique resources and assets for companies originated from these activities (Chesbrough, 2006).

1.1.8 **Licensing**

The success of open source development depends on the developers willing to work for the open source producers “for free”. Firms need free-software developers that would contribute their work to the firm, and to the developer “open source community”, without demanding or receiving money in return. This is formalized in the company’s choice of an “open-source license”, which specifies the terms and conditions under which the company’s open-source products can be used, modified, and redistributed (Hecker, 1999). An open source license can be of GPL, LGPL BSD type. It can also be combined with a proprietary license, forming a “dual license”.

1.1.9 **Strategy vs. Business Model**

Business model and strategy are often used interchangeably by mistake. A business model draws on a multitude of business subjects such as economics, entrepreneurship, finance, marketing, operations and strategy. Therefore, the business model is a part of the corporate strategy and it is a reflection of the firm’s realized strategy. Casadesus-Masanell and Ricart (2009) found that there is one-to-one mapping between the two concepts only in simple competitive situations. However, they differ when there are important contingencies upon which a well-designed strategy must be based. Consistent with this notion, strategy refers to the contingent plan as to what business model to use and it is a high-order choice that has profound implications on competitive outcomes.

Chesbrough and Rosenbloom (2002) compare the concept of the business model to that of strategy, identifying the following three differences:

1. **Creating value vs. capturing value** - the business model focus is on value creation. While the business model also addresses how that value will be captured by the firm, strategy focuses on sustainable competitive advantage.

---


2. **Business value vs. shareholder value** - the business model converts innovation to economic value for the business. However, the business model does not focus on delivering that business value to the shareholder.

3. **Assumed knowledge levels** - the business model assumes a limited environmental knowledge, while strategy depends on a more complex analysis that requires more knowledge of the environment.

### 1.1.10 Intellectual Property (IP)

Intellectual property (IP) is a form of legal entitlement which allows its holder to control the use of certain intangible ideas and expressions that are creations of the mind: inventions, literary and artistic works, and symbols, names, and images. The term intellectual property reflects the idea that once established, such entitlements are generally treated by courts, especially in common law jurisdictions, as if they were tangible property. The key forms of intellectual property are patents, copyrights, trademarks, and trade secrets. Teece (1986) also considers complementary assets among the IP portfolio of firms.

### 1.1.11 Research and Development (R&D)

R&D refers to the systematic activity combining both basic and applied research, and aimed at discovering solutions to problems or creating new goods and knowledge. R&D activities are performed by a team of professionals working to transform a product idea into a technically sound and promotable product. R&D may also result in ownership of intellectual property such as patents.

Corporate R&D departments are found in both large and small companies and are generally responsible for product development and testing. Every corporation has a corporate R&D management strategy regarding product development, which will be covered in more detail later in this thesis.

### 1.1.12 Community

Open source communities consist of people who contribute to the public good of open source software by writing code for the project. Contributors also have access to computing equipment operating on a common standard and they can distribute their creations widely and essentially without cost via the Internet so that everyone can immediately obtain, test, and observe the value of freely revealed new software code (von Krogh & von Hippel, 2006).

### 1.1.13 Networking

Networks are a source of new business partners to commercialize new products ideas or prototypes which would otherwise stay ‘on the shelf’ (De Jong, Vanhaverbeke, Kalvet & Chesbrough, 2008). Firms utilize strategic alliances or form external networks in order to gain such knowledge or utilize complementary resources to exploit external sources of

---

7 [www.wordiq.com/definition/Intellectual_property](http://www.wordiq.com/definition/Intellectual_property)


knowledge (Chesbrough, 2005). This approach is particularly popular in technology intensive sectors, and open source software (OSS) is one of them.

### 1.1.14 Corporate Venturing

Corporate venturing are entrepreneurial efforts that lead to the creation of new business within the corporate organization. It can be internal or external. Internal Corporate Venturing (ICV) is a way for firms to continuously scan their existing assets and resource structure for new business opportunities and commercialization potential, such as spin-offs, and spin-outs. External Corporate Venturing (ECV), on the other hand, is new business creation activity through organizational modes such as corporate venture capital, alliances, or acquisitions (Keil, 2001).

### 1.1.15 Openness

Openness is the proportion of engagement in open innovation activities to the total innovation activities of the firm. Too much openness can negatively impact the long-term innovation success, because it could lead to loss of control and core competences. On the other side, too much closeness does not serve the increasing demands of shorter innovation cycles and reduced time to market.

### 1.1.16 Skill-set Analysis

Benefits of the open source depend on the level of the skill-sets of the firm. Each skill level enables an IT department to handle open source of different degrees of maturity and is described in terms of the following dimensions (Woods & Guliani, 2005). Thus, it is beneficial for firms that work with open source software to pursue a skill-set analysis to keep track of their skills and whether they require further skills to be able to successfully cope with problems that may arise in the future during development or integration of the open source software.

### 1.1.17 Maturity Analysis

Detailed analysis that helps firms working with open source software in carefully selecting the right open source product; otherwise they would eventually fall into “open source traps”. A maturity analysis discipline helps firms to overcome such traps, helping them choose the best alternative from various open source projects before green-lighting the implementation.

### 1.1.18 Productization

*Productization* refers to making software work for the general case and making it as easy as possible to use. It requires a huge amount of work, about double or triple the amount of work it takes to complete the original features and turn a program into a product for firms to pursue productization. Productization helps avoid the skills gap of firms and accelerates the learning process of firms, through examining the source code (Woods & Guliani, 2005).

---

10 [http://www.googol.se/backnet/upload/files/googol_wp_internal_corporate_venturing.pdf](http://www.googol.se/backnet/upload/files/googol_wp_internal_corporate_venturing.pdf)
1.2 Background

The software industry has evolved at a rapid pace to become one of the largest industries in the world economy, attracting a mass-market of consumers for business as well as recreational software (Dahlander, 2004). Software developers write computer software in the form of source code, and document the source code with brief written explanations of the purpose and design of each section of their program so that other developers can read and understand the source code clearly either to modify it in order to integrate that software in their own applications, or to test the software for potential “bugs”, or errors in the software code.

Firms in the software industry capture and secure economic returns from their software products by two main approaches today. First, they can follow licensing arrangements based on copyright law, where the software license provides individuals, groups of individuals and other firms competing in the same market with the legal rights to use a piece of software often in return for a licensing fee (von Krogh & von Hippel, 2003). Second, they can choose to protect the software’s “source code”, a bunch of instructions executed by computers to carry out the software program’s purpose.

Looking from another point, we see that the software industry has long been dominated by corporations (Microsoft, Apple, IBM, HP, Oracle, Novell etc.) which develop proprietary software. The term proprietary means "privately owned and controlled", thus the software can still remain proprietary even when source code is made publicly available, if control over use, distribution, or modification is retained. In this case, a separate license is required for another party to use the software in addition to the EULA. Shareware is a common example of such case, where the owner of the closed source software encourages redistribution at no cost, but which the user must pay to use after a trial period. The fee usually allows use by a single user or computer.

A common pitfall of the proprietary software is the dependency on the future versions and upgrades for the proprietary software package, which in turn creates vendor lock-in, which can give vendors monopolistic power. However, supporters of commercial proprietary software claim that requiring users to pay for software as a product increases funding or time available for the research and development (R&D) of the software. The fact that proprietary software tends to create greater commercial activity over free software, especially in regard to market revenues, is another main reason why proprietary software development has been dominant for long time.

Closed source still dominates the software development scene, but in the last few years, with the emerging success of open source projects like Linux, KDE, and Apache, corporate strategy has undergone a transformation. A considerable number of corporations have recognized that closed and open source projects can complement each other, and thus have started increasingly adopting open source strategies, even though they strongly rejected in the first hand. Even the commercial giants such as Apple, HP and IBM have realized that integrating an open source strategy into their overall corporate strategy would indeed bring them benefits that would not be possible otherwise.

The phenomenon of open-source software has emerged as a reaction against the perceived threat posed by the rising proprietary software industry, which has traditionally employed...
business models relying on object code licenses that used a combination of trade secret, copyright and contract law to severely limit the legal ability of users to copy, modify and redistribute the source code. Open source licenses, however, are consciously designed to use the legal machinery of proprietary commercial software licenses in order to encourage, rather than to prevent widespread distribution and modification of the source code.

Open source (OS) is a phenomenon of increasing significance for organizations today because of its capability to offer effective business solutions and new business opportunities. Apart from their involvement in conventional OS component adoption, many firms are currently getting involved in open source software (OSS) development projects considering the fact that it can bring competitive advantages to themselves (Refer to chapter 5.2. for such advantages). There is in particular a strong European interest in Open Source (or Libre), with an ITEA report suggesting that 70% of OSS developers live within the EU, and several EU funded projects investigating the phenomenon (Lundell, Lings & Lindqvist 2006).

Proprietary software developers prefer to license or sell the source code of their software products, thus they have a tendency to restrict the access to the source code strictly to protect it against their competitors and contractors. A direct result of this is that only “insiders to the firm” possess the information required to modify and improve that proprietary code further (von Krogh & von Hippel, 2003), reducing any possible contribution from outsiders who may possess expertise knowledge and better the product in various ways.

An “open-source” software product means that the source code for that product is freely available under liberal licensing terms, with no licensing fees. Others are free to take that software, make changes to it and use or distribute the resulting modified versions as they see fit. By making the right products open-source and selecting an appropriate business model for it, a company can ultimately benefit from the product far more than just compensating short-term losses of profits coming from no longer being able to sell the product in the traditional proprietary model.

When Netscape first made the Navigator Web browser available for unrestricted download over the Internet, many questioned how Netscape could possibly make money “giving the software away” (Hecker, 1999). However, this strategy is now considered to be a pioneering successful innovation that was the key to Netscape’s rapid growth in that era.

Many other companies than Netscape have benefited from the current interest in open-source development. For example, Red Hat Software, a distributor of the open-source Linux operating system, received funding from Netscape and Intel, where Linux vendors potentially help other software vendors such as Netscape that provide software for Linux and hardware vendors such as Intel that sell processors on which Linux runs (Hecker, 1999). Similarly, Novell distributes OpenSUSE, an entirely free Linux distribution and contributes various open source projects such as Apache, MySQL and many others.

These examples indicate presence of a specific type of companies working with open source software, companies which solely base their strategy on open source business models. Such examples show that companies can choose to make their source code freely available and accessible, yet still serve its own business interests as a proprietary organization. Their open source strategy determines how these interests will be served in commercial manner.
Since the late 90s, the open-source community grew—from 200,000 registered participants on SourceForge in 2001 to 1,200,000 registrants in 2006 working on over 110,000 projects; but more importantly, well-accepted open-source products have extended beyond just the operating system (Linux), moreover into databases (MySQL), applications servers (J-Boss), customer relations management (Sugar CRM), and even TiVo (Maxwell, 2006). There are now an incalculable number of open-source software programs such as Apache, Perl, Sendmail and many more.

The scope of open source has grown beyond basic development tools to become a top-to-bottom infrastructure for computing of all stripes, including development environments, databases, operating systems, web servers, application servers, and utilities for all types of data center management. Open source now encompasses a huge variety of end-user applications, such enterprise applications as Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM), tools such as portals and data warehouses, and integration tools for messaging as well as for web services (Woods & Guliani, 2005).

The growing success of these open-source products attracted further public attention on the potential importance of open-source software (OSS). Indeed, major large commercial IT companies, including Apple, IBM, Hewlett Packard, Sun Microsystems, Novell, and Computer Associates, have now integrated open-source software into their core strategies. Google uses open-source software for its core business of searches; Yahoo uses it in its core business of directories. Despite the widespread doubts and resistance during the early stages of open source revolution, there has recently been a “change of perception” in software development industry towards incorporating open source strategies into their business models.

Moving to an open-source model implies sharing the product strategies of these companies with their external developers and letting them influence those strategies; including their competitors. Although this may be thought to pose a threat to the company, this indeed results in greater public support for firm strategies, especially via communities, helping to counter those of competitors. Besides, releasing the source code does not necessarily imply or require making all internal information publicly available; firms can indeed continue to closely control confidential details of business plans and the like.

On the other hand, some software producers choose to circumvent rivalry with the open source software movement and search for ways to cooperate with it (von Krogh & Von Hippel, 2003). These firms develop derived products based on open source components by utilizing open-source software and customizing it for their own use without making their customized products generally available, thus merging open source business models with their own internal innovation processes. These companies do a considerable amount of in-house work that they do not then share with others, but they also support considerable shared work in open-source development.

1.3 Purpose

Our aim with this thesis is to explore the “open source scene in Sweden” by examining a number of Swedish firms that work with open source software and observe the characteristics of the software industry in Sweden going through these firms. We try to diagnose the major constituents of the open source strategies pursued by such firms at various levels. This can be seen from the research questions that be addressed throughout the thesis:
Research Question-1: What are the different types of components of an innovation strategy for a firm?

Research Question-2: Considering open source as a form of open innovation, how do these components reincarnate in open source strategy?

Research Question-3: What are the open source-specific components of an open source strategy?

Research Question-4: How do the components of open source strategy vary among the firms that work with open source software in Sweden?

1.4 Delimitations

Open source software is deeply rooted in communities. There have been many community-specific studies (West & Lakhani 2008; Fleming & Waguespack 2005; Dahlander, Frederiksen & Rullani 2008) on what motivates software developers in communities to contribute to the community without any economic returns or how communities communicate and collaborate within themselves. However, in our study we are not focusing in communities themselves, but rather how they can contribute to firms in development of open source or complementary software. We are also not focusing on government procurement on open source development, which can be also an influence in the development process.

Our study focuses on business strategies used in open-source development, and thus does not investigate proprietary strategies, but rather uses them as a basis of comparison with open-source strategies.

1.5 Disposition

The rest of the thesis has been constructed as follows:

Research Plan: The research plan that was used to carry out this thesis work is summarized, including all the steps followed to conclude the study in chronological and graphical format.

Methodology: A description of the techniques employed to acquire data from the Swedish companies that work with open-source development.

Instead of having a singular dense and large chapter for our frame of references, we have instead divided our theoretical background into three separate sections in order to give a better depict of the open-source concepts to the reader. By this means, we have aimed to apply the popular “Divide-and-Conquer” strategy in software development to our research to make the reader follow the concepts more clearly without losing the track.

Innovation: A general introduction to the concept of innovation followed by the Open Innovation: Highlights of open innovation and underlining of how open innovation differs from traditional innovation notion.

Open Source Software (OSS): From this section on, our study digs further into the notion open source software (OSS) and how it differs from open innovation concept or what kind of traits it inherits from it.
**OSS Strategies:** From the concept of OSS, we land on which specific strategies are employed in companies that develop open source software.

**Method:** We explain the data collection methods used in our study which have been touched in the Methodology section in detail here and the systematic approach that we used to analyze our data to investigate our research questions.

**Analysis:** An analysis of the empirical data for investigating open source strategies of companies working with open source development in Sweden is carried out in this section.

**Conclusions:** Conclusions are eventually derived from the analysis to summarize the whole study and results from the analysis section are discussed.

**References:** The final section in our thesis lists all the references that we have used for our research.

**Future Work:** Recommendations of how possible directions based on the existing findings and ideas of this study can be used for further research is mentioned in this section.

In order to improve readability, chapters in the thesis will be linked to provide the reader with a storyline. We give an introduction before entering the details of the theoretical framework in some chapters to give an overview of what will be discussed in that particular chapter to the reader. Likewise, a summary is presented at the end of various chapters to wrap up the important points that were explained in the corresponding chapter.
2 Thesis Plan
Our starting point of our thesis was the innovation concept. We knew that we would work on innovation as a topic for our thesis, because we have had interest for innovation strategies in particular. Since we are both coming originally from an engineering background, we had the incentive to work on a topic which could combine engineering with the innovation strategy concept, and upon sometime of brainstorming, we came up with open innovation, which has become quite popular lately. We have narrowed our focus further from open innovation down to open source software, which has become a hot topic since the start of the 21st century.

Our “divide-and-conquer” strategy can be observed from the figure above, where the three tracks of open innovation, open source and innovation strategy are researched simultaneously, instead of making a deep and long research on just open source strategy, and this breaking down of open source strategy concept continues until we start collecting data from our methodology after our theoretical framework has been finalized. At that point, we now have a single track and switch to analyze the data coming from the methodology chapter and discuss the findings. We finish the thesis with the conclusions drawn from our analysis.

As depicted from the figure above, we start our research from the uppermost abstract level of innovation. Therefore we will start our theoretical framework first by exploring innovation concepts, in particular those related to innovation strategy and then we will pass through open innovation very briefly in order to land in open source strategy at the end. After covering these three layers, we then wrap them up under “open source strategy” and try to address how these concepts are held especially in open source strategy.
3 Innovation Strategy

3.1 Background

“Innovate or die”; with markets becoming less and less local, companies are being advised to compete globally based on their ability to innovate successfully (Angel, 2006). Innovation is rapidly becoming the “X-factor” that strengthens business strategies, offering perhaps one of the few sustainable competitive advantages for firms. Hence, it is widely regarded as a key ingredient in business success; firms spend considerable resources to foster an innovative culture and introduce innovations, with varying results (Rogers, 1994). But what is innovation indeed?

Innovation, in its broadest sense, refers to the entire process by which technological change is deployed in commercial products, however such “commercial” definition of innovation actually incorporates formally protected intellectual property (IP) such as patents or copyrights that is difficult to imitate, or reflects tacit knowledge that is easily imitated and at best provides a transient competitive advantage (West & Gallagher, 2004). Innovation is a systemic change process, which consists of both the elements of the invention of an idea for change and its application and diffusion in practice (Könnölä, 2007).

This traditional “commercial” point of view of innovation has led to three traditional processes of innovation: invention, translation and commercialization. These elements specify the success of the innovation, which is crucial for survival of firms in the market. Elements associated with the firm, its environment, or the innovation itself have been found to differentiate successful innovation from the more common failures (Saren, 1984).

Just few companies place strong strategic emphasis on innovation and build their self-identity around newness. Most firms still focus their self-identity on sameness and lack of innovation.

The cost and time dedicated to the innovation processes turn up to be huge if the firms do not pursue perfectly executed innovation strategy that is well-aligned with the corporate vision, the core mission and values as well as their future technology and suppliers of the firm. Lack of alignment between the products and innovation processes increases the cost and risk for the firm. The organizational structure should be aligned as well; otherwise the firms will be in even deeper trouble.

3.2 Internal Components

The “clarity” induced by the innovation strategy is critical for success, because innovations are quite risky as they are new to the firm. Many firms are actually afraid to place clear limits and expectations on their innovation activities and fail to pronounce the corporate strategy “clearly”, which creates distractions. Leaving the product development team without an understanding of its mission and how the innovation efforts supports core business strategies or needs is almost criminal, and will usually result in failure.¹¹ Companies with clear

growth and innovation strategy in existing or new markets will grab the lion’s pie of the profits in the market from their innovations at the end\textsuperscript{12}.

If firms can lock up the key resources to assure competitive advantage, the path to profiting from innovation becomes clearer. Innovators must develop a business model consistent with both the value of the intellectual property (IP) and the innovator’s position in the value network and the core technology must be incorporated into the product (Chesbrough, Vanhaverbeke & West 2006).

Innovation strategy can be proactive or reactive. A \textit{proactive} innovation strategy depends on the quality of creative genius. Such strategy is not appropriate for large bureaucratic organizations where “structural inertia” is high and prevents the firm from implementing efficient innovative activities. In such organizations like Motorola and HP, the main goal is to create structures and reward systems that stimulate the creative genius and reward effort as well as results and have a tolerance for failure, along with a strong focus on the key innovation that will change the competitive structure of the industry.

A \textit{reactive} innovation strategy, on the other hand, requires more emphasis on process innovation rather than product innovation. Because such innovations are easier to implement, results are more emphasized and are viewed in terms of commercial success. Thus, reactive innovation culture is less supportive of the creative genius and more suitable for progressive approach of “step by step” development. It is thus natural to say that proactive innovation strategies are more suitable for inventive and radical innovation whereas reactive innovation strategies are more suitable for imitative and incremental innovation. Besides, reactive innovation strategy requires more attention to the competitors in the market than the proactive innovation strategy.

Today, most companies pursue a hybrid innovation strategy instead, a strategy which falls in between pure proactive and pure reactive approaches. The choice of the innovation strategy depends on the corporate strategy, existing innovation strategy, the history of the firm and the current resources.

Each of these elements is briefly explained in the following sections, except the business model. Business model will be covered in depth later in this thesis with the emphasis on open innovation and open source software. Licensing, which is a special form of intellectual property which is important for open source software, will also be covered in detail later.

\section*{3.2.1 Business Model}

Software business challenges are interconnected in two ways. First, most if not all are functions of constrained resources, only few companies have enough people, money, or time to do everything that needs doing, especially when competing against larger companies with greater resources. Second, a strategy exists to address all these challenges at once: turning some (or in exceptional cases all) of a company’s software products into open-source ones (Hecker, 1999). Thus, the major strategic question for firms becomes how much of their resources they shall allocate to open source development.

\textsuperscript{12} http://www.1000ventures.com/business_guide/crosscuttings/vision_mission_strategy.html
A business model performs two main functions: creating value through a series of activities from the resources to the end product, and capturing a portion of that value, through establishing unique resources and assets for companies originated from these activities (Chesbrough, 2006). The ability of open innovation to enable firms to integrate knowledge inside and outside the organization has resulted in an increasing number of firms that has started to use this approach, not only for small enterprises which is a must since they usually lack the knowledge to fully complete the innovation process, but also for large corporations, pushing them away from their traditional R&D approach to a collaborative connect and develop (C&D) process. Major information technology companies, including IBM, Hewlett Packard, Sun Microsystems, Novell, and Computer Associates, have now integrated open-source software into their core strategy. Google uses open-source software for its core business of searches while Yahoo uses it in its core business of directories. Many companies are utilizing open-source software and customizing it for their own use without making their customized products generally available (Maxwell, 2006). The desegregation of the innovation process, requiring internal and external knowledge, has opened room for the emergence of new business models and types of firms (Chesbrough & Appleyard, 2007).

A central concern to open innovation is how to best use the internal R&D capabilities of the firm to maximum advantage (West & Gallagher, 2006). Those capabilities can be used for selecting the appropriate balance between “commerciality” and “openness” of the open source products. There can be a pure proprietary model where innovations are solely internally commercialized by using the absorptive capacity to identify external innovations and integrate them internally for generating economic returns for the firm, or a pure open source model that does not produce direct economic benefit, but indirectly generates economic returns through spillovers or sale of related goods and products. Successful firms may combine a variety of these approaches using a hybrid business model. Various business models that are used in open source development will be discussed later in this thesis in detail.

### 3.2.2 R&D Management

**R&D management** is the process and techniques used to control the amount of money and effort invested in R&D projects; it is simply the management of R&D organization within a firm. R&D management implies the design of the R&D processes and it ensures transfer of newly acquired know-how and technology to other business units in the firm which work with innovation. Firms must dedicate resources to R&D as part of a long-term management strategy and the allocation of resources is determined by the R&D management strategy of the firm.

Firms have an increasing need for reliable mechanisms to direct R&D toward effective innovation and accumulation of long term technological strength. R&D managers often have difficulty deciding on the technical content of the longer term component of their R&D portfolio, requiring a transparent and readily accessible set of decision criteria in order at the very least to enable the R&D programme to have some perceived legitimacy within the wider realms of the company (Coombs, 1996).

Characteristics of R&D management include innovation, long-term management and technological uncertainty. The R&D management concept represents structural approach to manage R&D projects inside the firm and assumes that the source of technology is found inside the firm itself. This is done by integrating the R&D into the business operations. A
successful R&D management is one where the entire company helps manage the R&D process inside the firm, not just the R&D managers.

R&D management can be considered as the intersection of innovation management and technology management. Typical activities include basic and fundamental research, new product development, project management, product life-cycle management and R&D portfolio management. Nevertheless, it does not include licensing, innovation management, IP management and corporate venturing which are independent activities that can be carried out without the presence of R&D inside the firm. The main purpose of R&D management is to determine which projects to pursue and which not to. A clearly defined R&D strategy translates to a clearly defined project which has a higher probability of success in terms of new product development, and thus is crucial for successful innovations.

3.2.3 IP Management

Intellectual Property (IP) is the know-how that comes from creativity, it is the knowledge of how to do something better, which may be worth a fortune if firms can successfully commercialize the IP while at the same time preventing their competitors from copycatting others from doing the same thing. Legally protected intellectual assets such as IP are critical for business success for firms. Firms should have as strong IP portfolio if they want to achieve their growth plans. The IP portfolio includes trademarks, copyrights, trade secrets, patents and licenses obtained for the technology. Active IP portfolio management and systematic analysis of existing patents and their potential are important issues for the innovation strategy of the firm.

The decision whether to protect IP or not (or to license the product or not) depends on what the IP is worth, thus “value assessment” of the IP portfolio is important. Firms have to value the profit potential in their products, or similarly the savings or benefit from commercializing their inventions. A special area of expertise of professionals within the accounting profession should thus deal with valuation issues.

IP is the lifeblood of the enterprise for high-technology industries such as software. Thus, the top management should develop a corporate culture which understands the importance of IP. Engineering managers should be looking at licensing out that IP which is not commercially critical to the enterprise, thereby generating additional profits for the company as well as looking at licensing in technology which might tie in to their own\(^1\). IP fuels the profitable core competency on which the firm relies and delivers core competencies which fuel long-term sustainable competitive advantage (Iron Mountain, 2004). It is thus important for top management to have an understanding of the protection of firm’s most critical IP assets. The primary locus of value for many corporations has been found in their intellectual property rights, which can help a company gain competitive advantage in various ways. They can provide a temporary technological lead, or “incumbency”, as well as protection of brand names or help form an industry standard (Reitzig, 2004).

In open innovation, proactive IP management becomes a critical element since IP flows in and out of the enterprise on a regular basis. Firms can outperform their competitors by carrying IP information strategically. Binding human resources to a corporation is also impor-

tant for companies seeking to maintain IP advantages since the tasks associated with the execution of a proper IP strategy are many. In addition to tiling for and enforcing IP rights, companies must attend to licensing, technology forecasting, potential alliance targets, information provision and consultancy regarding the choice of research trajectories (Reitzig, 2004).

3.2.4 Innovation Roadmap

The roadmap approach refers to a structured and time-based representation of alternative futures of technological, industrial, policy and social developments and their dynamic linkages (Könnölä, 2007).

Innovation roadmaps provide clear innovation pathways that identify the key technologies, systems and people required to address the challenges facing the software industry. The objective of the innovation roadmap is to develop the innovation strategy; to stimulate innovation and strategic investment that will ultimately lead to sustainably developed projects within the industry (Victoria, 2006).

From a corporate technology management point of view, the most important and most often used type of roadmap is the product–technology roadmap, which incorporates both the technology-level and the product-level view and it is a major planning instrument in many technology-oriented industries such as software (Lichtenthaler, 2008). To be most effective, the innovation roadmap needs to be fully integrated into the firm’s corporate and business strategy (which in turn to be integrated with R&D and IP management strategies).

Road-mapping leads to effective project portfolio development and management, and it provides a common language for innovation and builds bridges between technologists and business managers within the firm, and with the firm and the suppliers and the customers. Roadmaps are often constructed as time-based charts with the multiple layers that usually include technology push and pull perspectives (Könnölä, 2007).

3.3 External components

3.3.1 Corporate Venturing

Open innovation implies that enterprises can choose alternative entrepreneurial strategies to commercialize internal knowledge, and also to benefit from external knowledge (De Jong, et al. 2008). Corporate entrepreneurial activities include corporate venturing, intrapreneurship, and spinning off new ventures. Corporate venturing enables the recovery of innovations that were initially abandoned or that did not seem promising, and is usually done by large enterprises, which enterprises may create corporate venturing programs to invest in start-ups and other businesses to keep an eye on potential opportunities (De Jong et al. 2008).

Tactics that embody an open innovation approach include exploiting knowledge spillovers, consulting with venture capitalists, while also using both inbound and outbound licensing of key technologies (West & Gallagher, 2004).

3.3.2 Communities

The role of communities is especially important for firms that innovate in open source software (OSS) projects. A significant proportion of these firms are in a symbiotic relation-
ship with the OS community, supporting both through participation in existing projects and the release of new software under OS licenses (Lundell et al. 2006).

Linux is arguably the most successful example of OSS development. However, there are many others that are still facing sustainability problems and cannot receive any significant support from individual software contributors. This reveals a second important issue: the supply of such contributors is not infinite, and the vast majority of projects suffer from a lack of contributors thus open-oriented projects must compete for contributors (Chesbrough & Appleyard, 2007). The motivation of the individual contributors is critical here. A third important issue is how the open invention or coordination project is led and how its agenda evolves. This is decided within the community itself.

A final strategic concern comes from looking at open initiatives from the perspective of firms. If firms cannot find ways to profit from their innovation activities in open initiatives—through deployment, hybridization, complements, or self-service (which will be described in detail later in this thesis), they cannot sustain their participation in those initiatives over time. Migration from the pure form of open invention to a more hybridized form of open and owned invention is one way that open-oriented firms can control their own destiny (Chesbrough & Appleyard, 2007). The challenge is being able to manage the mix to avoid alienation of the community, which could precipitate a product war where an open alternative is created to displace the portion that is protected by IP.

OSS development communities centrally involve innovating software users. User-innovators benefit from their own expected use of an innovation. In general, no one has to pay them to innovate, and users don’t have to make investments that can only be recouped if others adopt what they have developed. (von Krogh & von Hippel, 2006)

No formal quality control programs exist and no authoritative leaders monitor the development in a community. It is thus surprising that open-source development achieves smooth coordination, consistency in design and continuous innovation while relying heavily on electronic media (Yamauchi, Yokozawa, Shinohara & Ishida, 2000).

Dahlander and Magnusson (2005) propose a typology of three different basic relationship-strategies used by firms to inter-relate to their communities: (1) symbiotic (2) commensalistic and (3) parasitic. The two extreme strategies are the symbiotic approach, where the firm tries to co-develop itself and the community and the firm management is directly involved in community development; and the parasitic approach where, as can be understood from its name, the firm only focuses on its own benefits, without taking into account that its actions might harm the community. The intermediate way to inter-relate to the community is the commensalistic strategy, where the firm thrives on communal resources that are continually replenished, while keeping the direct involvement in the development of these communal resources to a minimum.

There are various other aspects of communities that have been studied in literature such as motivation of contributors, self-organization of communities or community culture; however we are neglecting such detailed issues in order not to lose our main focus in our study. We only consider firm-community relations and how firms can utilize the power of communities in implementing their open source strategy in our thesis.
3.3.3 Networking

One area of considerable recent research is the role of networks in promoting firm innovation. In particular, SMEs may rely more heavily on external knowledge networks as an input to innovation than do large firms (Rogers, 2004).

Innovation can stem from a firm’s internal investments in R&D; however firms are increasingly active in acquiring important inputs from a variety of external collaborators. Relevant network partners include customers, competitors, suppliers, consultants, engineers, industrial associations, government and private laboratories, universities and other public research organizations, governments and non-profit intermediary organizations and other countries. Users are also potential network partners that firms increasingly collaborate with. Enterprises may proactively support their users to further develop their products by offering toolkits and other materials to trigger their innovative efforts (De Jong et al. 2008).

Innovative output of all firms rises along with an increase in the amount of R&D inputs, regardless of the nature of the collaborators of the firm (i.e. whether it is another firm such as its competitors or a university etc.). However, R&D expenditures made by private companies play a particularly important role in providing inputs to the innovative activity of large firms, while expenditures on re-search made by universities serve as an especially key input for generating innovative activity in small enterprises (Acs & Audretsch, 1994).

Alongside the company’s own strategic research programme, another source of input to the key technological capabilities of the company is external linkages to public science, and to the technology assets of collaborators. The role of managing these sources of technology is growing as companies find it harder to cover the whole spectrum of relevant technologies internally (Coombs, 2006). Firms use formal relationships, such as licensing agreements, alliances, joint-ventures, etc. as well as informal relationships (non-contractual personal relationships) to source technological expertise outside their boundaries (Dahlander & Gann, 2007).

Cooperation between firms has the effect of internalizing the strategic externality between them, which, at least for high spillovers, leads to higher R&D and welfare the need to engage in R&D to absorb external knowledge further reduces the effective spillover coefficient between rival firms. This means that an increase in external knowledge has an extra strategic effect, over and above its obvious direct effect (Leahy & Neary, 2007).

The nature of collaborators that firms use as a source for their R&D and technology knowledge depends on the size of the firm. In terms of generating innovations, corporate R & D is more important for large firms, while spillovers from university research laboratories are more important in producing innovative activity in small firms. Small firms innovate through exploiting knowledge created by expenditures on research in universities and on R&D in large firms. Apparently large firms are more adept at exploiting knowledge created in their own laboratories, while their smaller counterparts have a comparative advantage at exploiting spillovers from university laboratories (Acs & Audretsch, 1994).
3.4 Firm-specific components

3.4.1 Innovation Processes

The process of innovation is defined as the development and implementation of new ideas by people who over time engage in transactions with others within an institutional context (Van de Ven, 1986). Four inputs to innovation are culture, infrastructure, process and resources and integration of all four is the key to successful innovation (Gaynor, 2004).

In order to incorporate external capabilities and knowledge into a company’s innovation process, the organization needs to be adaptive and open for change (Sousa, 2008). Thus, innovation processes should be capable of integrating external knowledge and firms have to carry on a systematic analysis of ideas and projects that are emerging from this integration procedure.

3.4.2 Innovation Culture

In order to boost their overall innovation, firms start with learning to tap into the creative potential of all the employees and their knowledge about customers, competitors, and processes, and the key is to establish the right organizational “climate”. Beyond this, many organizations also need to learn how to make themselves more attractive to more diverse and unconventional talent (Leavy, 2005).

HP and IBM are called innovative not only because they have developed innovative products over the time, but also they’ve encouraged and developed the culture that fosters innovation at the same time. The “culture” of a firm is determined by the management practices the organizations follow building. “Innovative culture” begins with management at all levels but building a culture that fosters innovation is both complex and simple (Gaynor, 2004). For making innovation to happen, first thing to be asked should be “Is innovation is on the organization priority list?”.

Firms with established innovation culture (autonomy, empowerment of employees, open ideas and business models and processes) indeed have a higher rate of success in their innovations. (Gassman & Enkel, 2007).

3.4.3 Firm size

Size is one of the most important elements of an innovation strategy since it can affect the structure and innovation processes of firms. While large organizations have more slack resources for new projects and diversification, greater challenges & more opportunities for promotion and growth among their employees, and more control over the external environment, they also are more bureaucratic and less flexible, are unable to change and adapt quickly, and tend to have impersonal work environments (Damanpour, 1996).

Large firms have apparently stronger cash flows to fund innovation and access to a wider range of knowledge and human capital skills than small firms, allowing higher rates of innovation (Rogers, 1994). While the Schumpeterian position has generally been interpreted as asserting that large firms are more innovative than their smaller counterparts, this is not yet validated in the software industry in particular.
Previous studies support the modified Schumpeterian hypothesis that the relative innovative advantage of large and small firms is determined by the extent to which a market is characterized by imperfect competition. Industries which are capital-intensive, concentrated, and advertising-intensive tend to promote the innovative advantage in large firms whereas small-firms tend to re-capture their innovative advantages in industries in the early stages of the life-cycle, where total innovation and the use of skilled labor play a large role, and where large firms comprise a high share of the market (Acs & Autretsch, 1987).

On the other hand, there are some factors that suggest small firms can have an advantage. For example, small firms find it easier to adjust employee incentives to provide optimal innovative effort, or allow less rigid management structures that allow key employees to devote time to innovation-related, not management-related, tasks (Rogers, 1994).

3.4.4 Complexity

Complexity is another issue than can influence innovation processes of firms. It has been handled in various ways in previous studies, but basically there are two main points of view that circulate in the literature: structural complexity and complexity of products.

Structural complexity has been defined and measured in different ways, for example as the number of locations at which work is performed, the number of jobs or services performed, or the number of hierarchical ranks performing different tasks. In complex organizations, coalitions of specialists in differentiated subunits increase the depth of the knowledge base which, in turn, increases the development of new ideas (Damanpour, 1996). However, such complex organizations also possess “structural inertia” that inhibits innovation in the firm.

Structural complexity of the firm may also mean the nature of the organization in the firm, for example multinational firms may possess an advantage over domestic uni-national firms since they have a larger diffusion area for knowledge spillovers and a larger variety of collaborators to form larger networks.

Complex products, on the other hand, carry three basic and closely related characteristics which set them apart from mass produced goods and deeply influence coordination patterns. First, they are high cost hierarchical goods, made up of many customized, interconnected elements (including control units, sub-systems and components). Second, they are produced in projects involving more than one firm and frequently many collaborating organizations. Third, there is a high degree of user involvement, through which business needs feed directly into the innovation process (rather than through the market as in the standard model) (Hobday, 1998).

3.5 Summary

Innovation strategy of firms were found to be composed of various components. To sum up, innovation strategy can be considered to consist of three classes of factors: internal, external and firm-specific.

Internal components are the ones which lie within the company and thus are under the control of the company, such as the business model, R&D management, IP management and innovation roadmap. Firms do have the ability and power to adjust these for the sake
of higher economic returns and competitive advantage, thus aligning their innovation strategy with their corporate strategy.

External components, on the other hand, are the ones that reside outside the company, yet firms do still have control over them through their innovation strategy. These include corporate venturing, communities and networking.

There exists a final group, which are the firm-specific components. These are rather static compared to internal and external factors, and it is thus more cumbersome to modify. Such components include innovation processes, innovation culture, firm size and complexity regarding the organizational structure and products that the firms develop.

This classification can be more clearly seen from the figure below, which shows the theoretical framework of our study considering innovation:

*Figure-2: The framework of our study for innovation strategy (Authors’ construction)*
4 Open Innovation

4.1 Introduction

External technology sourcing of firms is becoming more important for a number of reasons such as shortening technology life cycles, emerging technologies with the potential to disrupt market leaders’ positions, sharing costs and risks associated with the technology, globalization of the R&D activities and increased rivalry between firms in their product markets (Vanhaverbeke, Cloodt & Van de Van 2007). Competitive advantage no longer comes from the closed innovation model, but instead from inbound open innovation, which is the practice of leveraging the discoveries of others: companies need not and indeed should not rely exclusively on their own R&D (Chesbrough et al. 2006).

Open innovation has thus been proposed as a new paradigm for the management of innovation. In the traditional closed innovation systems, firms only discover, develop and commercialize technologies internally (Chesbrough, 2003). It is the antithesis of the traditional vertical integration model where internal R&D activities lead to internally developed products that are distributed by the company itself. Open innovation assumes that internal ideas can also be taken to market through external channels, outside the current businesses of the firm, to generate additional value. This approach places external ideas and external paths to market on the same level of importance as that reserved for internal ideas and paths to the market (Chesbrough, 2005).

Firms have to alter their usual metrics for managing innovation when they are dealing with open innovation; they have to develop new skills and organizational routines to tap into external sources of technology for utilizing the open innovation model, as implied by Cohen and Levinthal (1990)’s “absorptive capacity”. Development and improvement of the absorptive capacity is the heart of open innovation for innovative firms (Chesbrough et al. 2006). One of the major contributions of the open innovation approach is the perception that the locus of knowledge and the locus of innovation need not necessarily be the same (Gassman & Enkel, 2004).

Past work has shown that there is a great deal of difference in the innovation strategies of small and large firms. Van de Vrande, de Jong, Vanhaverbeke & de Rochemont (2009) find that the responding SMEs have increasingly engaged in many open innovation practices during the past 7 years and that but medium-sized firms are on average more heavily involved in open innovation than their smaller counterparts.

The future of innovation is not about outsourcing all internal innovation activities, but about following a flexible innovation strategy to allow companies to create more and better innovation by combining various strategies, such as reintegrating new businesses, scanning and integrating new technologies, commercializing patents, connecting external sources to the internal innovation process and launching new collaborations during the required period (Enkel & Gassman, 2004).

4.2 “Openness” of the Innovation

Today, firms not just invest on pure open innovation, but rather invest simultaneously on closed as well as open innovation activities. The question of the degree of openness that
firms engage in becomes important here, which is basically the proportion of engagement in open innovation activities to the total innovation activities of the firm.

The optimum degree of openness cannot be determined in advance, but rather dynamic, depending in the firms themselves. Too much openness can negatively impact the long-term innovation success, because it could lead to loss of control and core competences. On the other side, too much closeness does not serve the increasing demands of shorter innovation cycles and reduced time to market. The future lies in an appropriate balance of the open innovation approach, where the company or the institution uses every available tool to create successful products and services faster than their competitor and at the same time fosters the building of core competencies and protects their intellectual property (Enkel, E., Gassmann, O., & Chesbrough, H., 2009). Indeed, internal and external sources of knowledge are complements and they have to be combined to improve the innovative performance of companies (Chesbrough et al. 2006). Chesbrough & Rosenbloom (2002) have noted that a critical option for firms is to choose between different forms of openness in developing the firm’s business model.

Dahlander and Gann (2007) have made a classification of open innovation strategies with respect to the degree of the “openness” of the strategy, depending on how much “inbound” and “outbound” the innovation processes are. The three most important classifications are “selling outbound” and “sourcing inbound” strategies. They define “selling outbound” as how firms commercialize their inventions and technologies through selling or licensing out resources developed in other organizations. Similarly, the “sourcing inbound” innovation to how firms can use external sources of innovation.

Chesbrough et al. (2006) claim that firms scan the external environment prior to initiating internal R&D work and if there are useful ideas or technologies, firms use them. Dahlander and Gann (2007) base the third type of category on this point, defining “acquiring inbound” innovation as acquiring input to the innovation process through the market place.

**4.3 Open Source as Open Innovation**

In recent years, more and more technical products have become “smart” in the sense of containing microprocessors and software. Accordingly, software development constitutes an ever increasing part of overall new product development (NPD), even more so since embedded systems are highly heterogeneous and thus require more software adaptations than standard computers. In this situation, efficiency and effectiveness of software development become increasingly important (Henkel, 2006).

Software development projects are generally based on Internet-based networks or communities of software developers. The term “e-entrepreneurship” has been coined to address the discovery and exploitation of business opportunities in the internet economy, which has emerged in the last decade. One important and intriguing aspect of e-entrepreneurship is the formation of new ventures in the domain of open source software (OSS). OSS not only poses a threat to incumbent software vendors and challenges the prevalent paradigm of software development, but also opens up opportunities for new ventures (Gruber & Henkel, 2004).

In the last decade, stronger global competition led to the labor sharing and cooperation between firms’ innovation processes, outdating the “do-it-yourself” mentality in technology and R&D management (Gassman, 2006) and giving the way to “outside-in” thinking ra-
ther than depending solely on internal R&D activities and shutting down the doors entirely
towards the outside world. The approach of “closed innovation” has pushed corporate
R&D organizations into difficulties in the case where internal research generate spillovers
that cannot be internally commercialized, and in the majority of these cases the research
projects ‘sit on a shelf’ waiting for further development or implementation. As a result, the
benefits of the innovation accrue not to the firm that financed its development, but to oth-
er firms who are able to capture the benefits of the innovation instead (Chesbrough &
Crowther, 2006). The best known contemporary example of such spillovers was observed
in Xerox Corporation during the 1970s.

In 1970s, Xerox corporation was running its business successfully, growing fast to enter the
Fortune 500 list. With a dominant share of the booming copier market, they became high-
ly profitable. In 1970, Xerox established the Palo Alto Research Center (PARC) in Palo Alto,
California. PARC has produced many of today’s technologies (the mouse, the Ethernet
protocol, graphical user interfaces and Post-Script are just some of these), however it has
been heavily criticized for its inability to capture value for Xerox. The main problem came
from the innovation process management inside the company, because Xerox was stuck
with the closed innovation paradigm. Indeed, most of the technological achievements that
PARC has made over the years gained recognition and became commercial when key
PARC researchers left to other companies or started new ones, both of which has rejected
to use the strict vertical integration and closed IP management policy that Xerox had used
for decades (Chesbrough, 2003). Typical examples of such resulting products are the Apple
Macintosh, based on the user interfaces developed at PARC; and the Microsoft Word pro-
cessor, based on the Bravo software developed in PARC. Xerox allowed these key em-
ployees to leave, because the management did not see any potential and commercial bene-
fits from these technological developments inside PARC.

The lesson learnt from Xerox and its PARC research center was significant for the software
development that followed, starting from the 1980s. By the 1980s, “outside-in thinking”
has started to emerge, which has since become the cornerstone of open innovation, and
builds on external sources of innovation.

With the “outside-in” approach, opening the firm’s boundaries to external inputs in an in-
ternally-managed way enables companies to realize radically new product innovation as
knowledge is growing faster and clusters of highly specialized knowledge are globally dis-
persed (Gassman, 2006). Indeed, external sources of knowledge and innovation have be-
come increasingly relevant the technological success of open source software, such as Linux
and Apache, has played an important role in spreading open innovation thinking.

Open-source software is a great example of open innovation because of the shared rights to
use the resulting technology as well as the collaborative development of the technology
(West & Gallagher, 2006). Thus, open source as an open innovation strategy has two key
components: shared rights to use the technology, and collaborative development of that
technology using donated labor. “Open source” software (OSS) includes source code that
can be modified and redistributed to others, while acknowledging the original author’s con-
tribution. In comparison, proprietary firms aggressively protect their software source code.
Apart from providing valuable software for free, open source projects also offer informal
development collaboration, helping participants to improve the code they contribute and to
develop it further even when internal resources would not be sufficient for this (von Krogh
& von Hippel, 2003).
OSS may be produced by an individual, a group, a company, or a combination of all of the above. This ecosystem of people, users, and commercial entities with a common interest in the software are collectively called "the community" (Cote & O’Grady, 2007). One of the most important distinctions between closed and open source software is the presence of the community. The important resources are not directly controlled by firms, but partly reside within communities that co-exist with the firms. The relationships that firms have to these communities influence their way of doing business while information and support is spread among the people involved in the community; and more importantly, innovations are shared within this community (Dahlander & Magnussen, 2005).

OSS dates back to the 1970s, including university-based research on BSD Unix during the 1970s and the ‘free software’ movement launched by Richard Stallman, a programmer at the MIT Artificial Intelligence (AI) Laboratory in 1984. Stallman felt distressed by the general trend in the software world towards development of proprietary software packages. As a result, he founded the Free Software Foundation (FSF) to counter the trend towards proprietary development of software packages. Stallman proposed that for-profit business models should treat software as a professional service rather than as intellectual property. Similarly, many companies have used software originating in the free-software community as the basis for commercial products, and in some cases have contributed to the development of free software through donations of money, hardware, or their employees' time (Hecker, 1999).

The basic license developed by Stallman to implement this idea was the General Public License (GPL), or commonly referred to as “Copy-left”. Basic rights transferred to the GPL-licensed software include the right to use it at no cost, the right to study its “source code,” to modify it, and to distribute modified or unmodified versions to others at no cost (von Krogh & von Hippel, 2003). GPL guarantees that anyone can modify and redistribute programs provided this does not hinder others from modifying and redistributing them. The availability of the source code in open source software enables users with special requirements to tailor programs, skillful users to eliminate bugs they encounter in use, and innovative programmers to enhance programs even outside a formal project (Yamauchi et. al., 2000).

The word “free” has traditionally led commercial software producers to think of “no revenue,” and customers of those companies to think “no support”. Thus, most firms in the commercial world saw free software as irrelevant, and free-software developers as idealistic and naive (Hecker, 1999). However, by making the source code freely available, open source firms can still serve its own business interests as a for-profit organization.

Prior to the success of open source software projects, it was assumed that a requirement to contribute one’s innovation to a commons would lead inevitably to the destruction of incentives to innovate, due to free-riding by others on the product of the innovator’s labor. (von Krogh & von Hippel, 2006). The surprising success of OSS projects such as Linux, Apache, Sendmail, or Jabber has startled the software development community as well as outsiders such as academics researching this phenomenon and management of software companies that has been traditionally developing commercial software. The publicly and freely available software packages of open source have reached wide diffusion as well as a quality comparable or even superior to that of commercial substitutes despite the fact that
they were, at least in earlier phases of their history, not supported by any commercial company (Gruber & Henkel, 2004).

In addition to free access to source code, open source is unique in its ability for users to contribute to the project itself. While that process is by no means required, as a user you may be in the position to contribute back to the project in the form of bug fixes, documentation, testing, or even code for entirely new features there is an important advantage to contributing back fixes and updates (Cote & O’Grady, 2007).

There is still a misunderstanding between “Open Innovation” and “Open Source”, which are occasionally misused interchangeably with open source methodologies for software development. Both share the concept of greater external sources of information for creating value, however open innovation explicitly incorporates the business model as the source of both value creation and value capture. Value capture enables companies to sustain their position in the industry value chain over time. While open source shares the focus on value creation throughout an industry value chain, its proponents usually deny or downplay the importance of value capture (Chesbrough, 2005).

If firms cannot or don’t want to develop sufficient absorptive capacity themselves, they may utilize strategic alliances or form external networks in order to gain such knowledge or utilize complementary resources to exploit that knowledge (Gassman, 2006). These networks typically include suppliers, customers, universities and similar academic institutions, government and private laboratories, competitors, or other countries. This approach is especially popular in technology intensive sectors like the open source software (OSS) industry.

Open source’s low cost has contributed to the widespread adoption of sophisticated development platforms and tools, including operating systems such as GNU/Linux and FreeBSD, databases such as MySQL, application servers such as JBoss, optimizing compilers such as the GNU Compiler Collection, integrated development environments such as Eclipse and KDevelop, build managers such as Make and Ant, and version control management systems such as CVS. Today, even small programming shops with a couple of developers can use sophisticated tools that once only large and well-funded development efforts could afford (Spinellis & Szyperski, 2004). Open-source software has changed this situation: we can now access millions of lines of code (of variable quality), which we can read, critique, and improve, and from which we can learn.

While open source software may be free to download and use, this does not typically equate to free in terms of usage and deployment: production deployments are very rarely free, and typically involve expenditures for support, services, training, and even project management. Though this change in fee structure can be difficult for traditional procurement organizations to adapt to, many open source buyers have success by simply treating the support and service fees as the license in contractual terms (Cote & O’Grady, 2007).

Open Source is only Open Innovation if it has a business model. AOL’s exit strategy with Mozilla is a good example of open source being not open innovation, which (like many of the Xerox PARC spinoffs) reflected the failure of the sponsor to create a viable business model, leaving the foundling innovation abandoned to whoever is willing to nurture it (West & Gallagher, 2005).
4.4 **Summary**

We have discussed the concept of open innovation in this particular chapter and explained that open source is indeed rooted in open innovation, with an increasing number of firms being engaged in the “outside-in” approach, and opening their boundaries to external inputs in an internally-managed way enables companies to realize radically new product innovation.

The most important element that pertains to firms that follow the outside-in approach is the degree of openness that they implement in the firm, which determines how much the firm opens its boundaries to external sources of knowledge. A firm may choose to pursue a purely open innovation strategy; however this would bring drawbacks considering that its competitors would have access to the firm’s internal resources as well, which could lead to intellectual property infringement. In this case, firms should choose an appropriate license that would not prevent from utilizing the benefits of open innovation and protect the intellectual property portfolio of the firm as well, as we will see in the following chapter.

We can thus add open innovation to the framework we presented for innovation strategy back in section 3.3.4., and use it as a basis for building the framework for open source strategy step-by-step:

*Figure-3: Innovation framework of our study with open innovation added (Authors’ construction)*

More components will be discussed in the following sections which will eventually lead us to a final framework to be used for open source strategy.
5 Open Source Software

5.1 Licensing

A distinctive characteristic of open source projects compared with traditional proprietary software development projects is the way intellectual property (IP) rights are handled (Tuomi, 2000). In the traditional software business model, companies provide all (or almost all) of the value to customers, and they realize revenues and profits in return through traditional software license fees, whereas in an open-source business model much of the value provided to customers will not be provided solely by the firm, but rather by other developers who are attracted to working on the firm’s open-source products and who will contribute to the firm’s resources as opposed to its competitors’ (Hecker, 1999).

Open source software is a public good: its use is non-rival, and it involves a copyright-based license to keep private intellectual property claims out of the way of both software innovators and software adopters—while at the same time preserving a commons of software code that everyone can access (von Krogh & von Hippel, 2006).

Licensing is undoubtedly one of the more controversial aspects to open source software, but the actual risks are often poorly understood. For example, in many cases, open source customers have no intentions of distributing the software itself, which minimizes concerns around even the most restrictive open source licenses they might encounter (Cote & O’Grady, 2007). However, without open source license, it would be very risky to build a system that so critically depends on a resource that is produced outside the community (Tuomi, 2000).

Various open source licenses have been published for use with open-source software; some work better than others for particular business models. All share the fundamental trait that the property rights to its use are placed in the public domain, making software “free” to users both by being no-cost and by minimizing restrictions on use and redistribution. They differ in the extent to which they allow public domain property to be mixed with private property rights. The historical trends have been to tolerate a hybrid of both (Kogut & Metiù, 2001). Since companies cannot use traditional software licenses and license fees with open-source software, they must find other ways of generating revenues and profits based on the value they provide to customers. Doing this successfully requires selecting a suitable business model and executing it well (Hecker, 1999).

Open source licenses represent free spillovers that can easily serve as external innovations for firms in their own products, whereas restrictions of free software licenses assure that the shared innovation remains shared, while limiting incentives for further commercial investment to develop and enhance the technology (Chesbrough et. al., 2006). From here we define two groups of companies in the technology industry that develop software products. The first group sticks to the traditional business model and develop commercial software products, but base those products on open source development. They take the advantage of open source licenses like Apache and BSD which allow them to use these open source components to develop their own proprietary software. The second group of companies, which stick to GPL and GPL-like licenses, focus on pure open source software development and utilize the full power of outside-in thinking via communities and networks such as universities or other companies which share the same ideology of being against development of
proprietary software using open source and believe that this is against the nature of open source ideology.

5.2 Open Source: Firm Perspective

From an open source customer’s perspective, one of the major benefits of open source is that users of open source can perform self-service evaluations and Request for Proposals, or RFPs: running and testing the software, talking freely with other users in the community about their experience, and discussing project needs of firms with the community (Cote & O’Grady, 2007).

Entering the software industry as a vendor of a new proprietary system would require considerable capital just for matching the present development level of incumbent systems. This barrier to entry is strongly reduced for firms basing their market entry on open source software. Also the barrier erected by extensive continuing R&D efforts of incumbents is much easier to surmount for OSS (Gruber & Henkel, 2004).

When the software product is proprietary, the platform provider and the application provider invest only in their own product to maximize their profit. However when it is open source, there is no platform provider firm, but the users invest in the platform to maximize their user surplus and their development reputation, which depends on the success of the platform measured by its adoption (Economides & Katsamakas, 2006).

In recent years, the requirements on software products have become more and more demanding, forcing software firms that offer proprietary software products to stay on top of these developments. A small firm entering this industry would have to replicate the R&D effort of incumbents to join the game in the first place, then would have to work continually to stay abreast of further technical developments, and needs to have the capacity to do error correction (“bug fixing”) and testing of its developments to make sure it does not ship flawed code to its customers (Gruber & Henkel, 2004). All of these factors require extensive access to human and financial resources, which small firms lack, putting them in a very disadvantageous competitive position against larger companies. These “liabilities of smallness” are much less relevant for software firms dedicated to open source. Such firms can thus focus on those features that their customers demand, instead of expanding effort just to stay up-to-date with respect to “must have” developments.

Proprietary software producers have increasingly gotten involved in open source software nowadays. Such firms make limited use of openness to win adoption of their technologies, either by opening portions of their technology or providing partly open access to key technologies (West & Gallagher, 2005). Just because firms no longer charge traditional software license fees doesn’t mean that they can’t sell the software in some form; firms can indeed implement indirect price discrimination, selling the software in a traditional retail package with CD and hard-copy manual included, while still allowing users to acquire the software at no charge through other means (for example, downloading it over the Internet) (Hecker, 1999). Firms can also derive revenue from product support, maintenance and services, depending on the open-source strategy.

Pursuing an open source strategy will lower the cost of production for the software producers and as a result increases the profit margin. There is a service revenue stream also possible for firms, depending on their business model.
Large companies especially care about product performance rather than the product development style. Open-source products have started being used in a wider range of companies, especially large software producers such as IBM, Oracle, Novell or Apple because of their increased performance over the years that caught up and even surpassed their commercial alternatives. OSS has been increasingly adopted by the software firms because of their technical merits and their ability to meet stringent requirements of the firms. Examples of notable adoptions include Amazon and Yahoo’s use of Perl, Orbitz’ use of Linux and Apache and Google’s usage of Linux. Another important feature of open source products is their huge amount of support backed up by the community (even for free) and the relative ease of configuration, user-friendliness and management compared to commercial software. Even large firms have difficulty in managing their huge and complex products, whereas community-driven open source products receive support from an unlimited number of contributors around the world, compared to the small and narrow in-house staff of commercial firms. Thus, there is a higher chance of solving potential bugs and errors and can lead to faster and better improvements on the product.

Since open-source products are usually available for free as an online download, firms can treat it as a low product risk. They can simply download the product and even if they decide not to implement a new product based on it, they will not pay anything since the original product is free. Firms also need to evaluate the total cost of ownership (TCO) of corporate alternatives with open-source products. TCO is the total cost of purchasing, installing and maintaining the open source product for firms. If TCO is lowered with open source products, it is of course beneficial to green-light open source integration. TCO is furthermore sensitive to the nature of the organization and should be evaluated by each organization.

Because of such advantages over the traditional commercial software, open source success stories are well known and increase every year. For example, the city of Munich chose OpenOffice.org, an open source suite of desktop applications, over the commercial software giant Microsoft which had constantly sought the contract aggressively, sticking a finger in the eye of commercial software development. Similarly, Amazon.com dumped Sun hardware and software in favor of Linux, the most popular open source operating system. Just like Linux, Apache, open source web server architecture, has become the most popular web server in the world and Perl, a robust scripting language, now runs huge, highly scalable sites such as Ticketmaster. Large financial companies, who used to strictly resist against the open source revolution, are now creating massive clusters of Linux machines for crunching numbers in complex portfolio analysis. This is just the tip of the iceberg.

Open source is being adopted by entire governments around the world. China, Brazil, Thailand, Peru and Turkey—are all adopting open source software officially and are spending millions to improve the software and encourage its adoption and contributing to the open source community (such as “Pardus”, a Linux distribution developed by the The Scientific and Technological Research Council of Turkey (TUBITAK) backed by the Turkish government, which currently ranks 42th in popularity among all Linux distributions).

All of these movements in the software industry are clearly nothing but a sign of a “change of perception” from the traditional commercial software development towards open source software, from small firms to large corporations and even governments. The opportunity provided by open source is too large to ignore for any organization that seeks to support its operations with software development. Companies, large and small, have taken to open
source as a way to increase collaboration, reduce development costs, provide a friendly platform for their products, and sell services (Woods & Guliani, 2005).

A major advantage of open source software is that it can be easily customized and adapted for a specific audience of users, in making it simple and functional. OSS firms can create a specific development environment, an application for email services to the public or a customized computer workstation with custom functionality. The products are highly reliable and simple to administer and maintain. Current cost is very low and the total cost much lower than the corresponding traditional proprietary alternatives.

For governments and state institutions, OSS can lower the cost of the apparatus of the state’s own requirements while it increases competition in the market while enhancing democracy to the citizens through increased availability and service levels. Open software and open standards are key elements to making the public administration. At the same time, it increases the chances of new small businesses to set up, which in turn leads to local and regional growth.

A number of free products like Microsoft Office contender OpenOffice and Google’s highly-successful Android operating system have questioned that view and shown that profit-driven projects are not always the best choice out there.

Despite all these benefits, there is still hesitation, especially at the government level, about switching to open source software. This is mostly due to the close ties with the large corporations that the governments are buying the commercial software from. In the world of politics, where tax incentives are offered to technology firms to keep them on side, it does not bode well to cut contracts that are extremely profitable to these commercial giants. Another reason appears to be a general fear of change, even if the change is clearly significantly better and provides many long-term benefits.

The European Commission came under pressure from Ministers of the European Parliament (MEPs) recently when it attempted to renew a contract with Fujitsu for expensive Microsoft software, which will be used on 36,000 Commission-operated PCs, with a hefty bill of €48.9 million ($67.8 million)\(^\text{14}\). Many believe this is something that troubled Europe simply cannot afford. Such economic difficulties has led to increasing concerns about commercial software and as a result countries all around the world has decided to re-consider their open source strategy, green-lighting for increased open source adoption.

### 5.3 Open Source in Europe

Both nationally and at EU level, there have been a number of initiatives to promote the distribution and use of open source software in public administration. A number of reports have been produced on behalf of the European Commission. The project FLOSS (Free/Libre Open Source Software) was announced in June 2002, for instance. FLOSS has been funded by the IST (Information Society Technology), and the project aims to meet demand for information on open source software.

IDA (Interchange of Data between Administrations) started in autumn 2002 a pilot

\(^{14}\) http://www.techeye.net/software/europe-squares-up-to-us-in-open-source-race
project, which includes preparation for possible migration and the introduction of a
solution based on open source software for the authorities in the German state of
Mecklenburg-Vorpommern, lowed by a study group with Swedish participation. The
program is operated by the European Commission and aimed at promoting use of ICT in
the exchange of information between administrations in the EU. “e-Europe” is an initiative
launched by the EU Commission in 1999 to drive IT development in Europe.

There are at least 20 countries around the world, where the governments have taken a
position on the use of open source software. Some countries, such as in South America with
Peru at the helm, has chosen to legislate on the use of open source software, while others,
like Britain and South Africa, has a softer line with the recommendations and policies for
open source software. Germany is perhaps the most active country in Europe with a
number of pilots for the introduction and use of open source software.

The Danish Board of Technology has produced a report, ”Open source software – in the e-
government”. The report includes an economic analysis of open source software and notes
that the transition to an electronic management in Denmark will require large investments.
The report states that the Danish authorities can save up to about 4.5 billion a year in the
short term (4 year) by switching from proprietary to open source software. It is primarily
on workplace computers, office software, operating systems and unique business system
that the biggest savings can be obtained in combination with an extended lifespan of
desktop computers and servers. Conditions in Sweden would be similar to the Danish.
Theoretically, therefore, saving potential in Sweden could be the one presented for
Denmark. There have been little cost comparisons between open and commercial software
in Sweden, though.

OpenForum Europe (OFE) is a not-for-profit industry organization which was originally
launched in 2002 to accelerate and broaden the use of open source software among
businesses, consumers and governments. OFE’s role has since evolved and its primary role
now is to promote the use of open standards in ICT as a means of achieving full openness
and interoperability of computer systems throughout Europe. OFE is a registered interest
group with the European Commission that devotes much of its time to explaining the
merits of openness in computing to politicians and legislators across Europe.

OFE works closely with the European Commission, European Parliament, national and
local governments both directly and via its national partners. It fully supports the European
Commission’s Digital Agenda, which aims to create a flourishing digital economy in
Europe by 2020. OFE corporate members are Google, IBM, Oracle, Red Hat and Deloitte.
OFE has national partners from across Europe, representing many tens of thousands of
SMEs. It also has a partnership with the Free Software Foundation Europe (FSFE) and
collaborates with the Foundation for a Free Information Infrastructure (FFII).

5.4 Open Source in Sweden

Public administration in Sweden is not a big user of open software, with the most common
users being universities and higher education, having a long tradition of open source

16 http://www.openforumeurope.org/
software. Many universities rely on open software for the operation of servers, websites, e-mail systems and so forth. Linköping University, for example, makes use of most of the open software programs listed in the FOSS report, utilizing a UNIX-based IT infrastructure in education. Apart from universities, many other institutions have decided to go for open source. SMHI (Sveriges Meteorologiska och Hydrologiska Institut) has opted for OpenOffice/StarOffice as the default office application while the Swedish Pensions Agency PPM (Pensionenmyndighet) uses the Linux operating system. Many other authorities have announced their interest in open source software in associated with the need to renew their environment on workplace computers. AMS (Arbetsmarknadsstyrelsen) conducted an evaluation of StarOffice as office applications and found that there is a large potential savings of a switch. Some clinics in Malmö Allmänna Sjukhus has begun using Linux as an alternative to Windows, and OpenOffice / StarOffice as an alternative to MS Office. It uses a solution with thin clients based on Citrix.

In 2007, Sweden’s leading consultancies in open source, Redpill, became a strategic partner of SJ (Svenska Järnvägar), introducing EnterpriseDB for its integration platform that connects SJ’s systems and processes. The recipe of success for EnterpriseDB, or the “Oracle killer” includes the low price, its performance and its compatibility that greatly facilitates the transition from Oracle database. The product won the prestigious “LinuxWorld Product Excellence Award for Best Database Solution” in 2007, for the third consecutive year of competition. One of the main causes of SJ’s election was the price 17.

In addition to SJ, Redpill also received an order from the Swedish product company, Cambio Healthcare Systems, with Cambio’s system of clinical care support and patient management Cambio COSMIC hosted by the open source application server JBoss from Red Hat 18. The agreement covers access to education, support and expertise of the transition to the application server JBoss. Seven counties (län) in Sweden have selected Cambio COSMIC. Two of these, Uppsala County Council and County Council of Kronoberg and Capio’s installations in Sweden at St George’s Hospital in Stockholm and Lundby Hospital in Gothenburg, has already migrated to JBoss.

Sveriges Kommuner och Landsting (SKL) is actively working to promote the use of open applications. Together with municipalities, Osby, Eslöv, Motala, Värmdö, Sundsvall and Örnsköldsvik and Region Värmland, SKL has established a national center of excellence for open source software and open standards. The aim is to create new opportunities for sharing information, experience and software 19.

Looking at pure costs, Sweden lacks the advantages to compete at the level of continuously fast growing countries of Far East, namely India or China. However, when moving up in the engineering class or the highly innovative production talent, Sweden indeed ranks right up there. However, according to Sun’s open source expert Danes Cooper, Sweden is too far behind in open source. While the rest of Europe is lagging behind, but countries such as England and Germany are better 20.

18 http://www.cisionwire.se/redpill/sakerhetskritiskt-sjukvardssystem-anvander-oppen-kallkod
19 http://www.skl.se/web/odell_positiv_till_oppen_kallkod_inom_staten.aspx
20 http://www.idg.se/2.1085/1.23912
5.4.1 **Open Source Sweden**

*Open Source Sweden* is an association aiming to safeguard the interests of Swedish suppliers of open source, inspired by OpenForum Europe. The association has been formed at the initiative of some of the leading Swedish companies in open source. Membership is available to all companies that work in open source software, with products, developments or services. The association is operated in accordance with annual action plan adopted by the Board. The practical work is carried out partly by membership meetings held each quarter, and by different working groups on a voluntary basis, operates various issues and activities.

Open Source Sweden is an industry association that supports the interests of Swedish open source companies that seeks to maintain the interests of Swedish companies dedicated to supplying open source products, development and services. As stated by the official website, their mission is “to stimulate a healthy market for Software through the development, provision, and support of products and services based on open source software and open standards”. The association is a member of the European organization *Open Source Business Organizations of Europe (OBOOE).*

21 [www.opensourcesweden.se](http://www.opensourcesweden.se)
6 Open Source Strategy

As the case of OSS demonstrates, the fact that many important inputs to the open innovation processes are public does not mean that this fact hinders open source innovators from capturing economic returns. Since the open source community protects the commons from being depleted by commercial firms, firms that attempt to appropriate returns from OSS ought to use different strategies to appropriate returns than in private goods (Dahlander, 2004).

Open business models define the requirements for open source architectures and systems; where the business model utilizes both external and internal ideas to create value, while the internal mechanisms claim only some portion of that value (Chesbrough, 2005). A study by Forrester Research in 2006 analyzing the corporate market found that 60 percent of major businesses plan to implement some open-source software in the coming years (Maxwell, 2006).

Green-lighting for an open-source strategy not only boosts ability of firms to capture value, but also enables them to create even more value via their software products. A good example of this is the software and hardware giant IBM, which has been popular for its strict and bureaucratic proprietary business model in software for a long time. This situation changed dramatically by the late 90s for the company, when IBM began to embrace Linux and construct its own business model around the Linux code. This new business model was distinctly different from the earlier proprietary software models. The specific portion of IBM’s funding of Linux allowed its internal programmers to optimize the code base to run very effectively with IBM’s other hardware and software products (Chesbrough & Appleyard, 2007). Mainstream software and hardware products are not the only revenue sources for IBM; the company also makes quite a lot of money on complementary hardware and software items. Thanks to switching to the open source model, the company can now capture much greater value via participation in the open source community. However, apart from increased value, implementing open source development will also bring extra costs to the firm. Here, the open source strategy of the firm will determine if the benefits of using open source will exceed the costs and vice versa.

There are many different ways in which open source ideas can be adopted by business, and influence the way in which companies do business. A number of different business models have been observed, ranging from the use of open source infrastructure products to basing a company’s entire business model on Open Source (Lundell et al. 2006). The key question concerning these business models for the firms is, can firms use open source profitably at your organization and if so, how can they do so?

The casual use of the terms open source and free software among programmers and in the press has led to a misconception that the software’s creators have otherwise volunteered not to exercise their copyrights (or to file for patents) (Woods & Guliani, 2005). However, firms can rely on various approaches to appropriate adequate returns to survive. *Table-1* shows how firms can try to appropriate returns divided into the possible ways of creating revenues (Dahlander, 2004 and Chesbrough, 2005). These open business models are not mutually exclusive, they may evolve over time, and companies frequently pursue more than one simultaneously, forming hybrid models. The self-service model in particular omits an
explicit value capture mechanism, which raises the question of whether this last model is sustainable over time.

<table>
<thead>
<tr>
<th>Category</th>
<th>Strategy</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Products</td>
<td>Licensing</td>
<td>Licensing the right to use the software (adding a proprietary part to the open code or by allowing the customer to use the source code how they wish)</td>
</tr>
<tr>
<td></td>
<td>Multi-Licensing</td>
<td>Distributing open source software under multiple different sets of terms and conditions, with a combination of GPL and non-GPL licenses</td>
</tr>
<tr>
<td></td>
<td>Versioning</td>
<td>Multiple versions of a technology such as a public free version and a private commercial version are offered</td>
</tr>
<tr>
<td></td>
<td>Hybridization</td>
<td>Proprietary innovation investments are made that rely on intellectual property ownership for add-ons (proprietary extensions)</td>
</tr>
<tr>
<td>Services/ Deployment</td>
<td>Black-box / Complements</td>
<td>Bunching several pieces of OSS in a hardware solution.</td>
</tr>
<tr>
<td></td>
<td>Consultancy</td>
<td>Consultancy work based on an area of expertise, either a product of the firm or a community established project</td>
</tr>
<tr>
<td></td>
<td>Education</td>
<td>Education based on an area of expertise, either a product of the firm a community established project.</td>
</tr>
<tr>
<td></td>
<td>Support</td>
<td>Support based on an area of expertise, either a product of the firm or a community established project</td>
</tr>
<tr>
<td></td>
<td>Self-Service</td>
<td>A user community creates a software application for its own needs</td>
</tr>
</tbody>
</table>

Table-1: List of Open Source Strategies for generating revenues of firms

6.1 **Focus on Products: The Licensing Strategy**

One key innovation that contributed significantly to the growth of open source software was the development of software licenses that prevented corporations from simply taking open source software and embedding it into their products. These open source licenses ensure that software developers can control the terms under which others can reuse the software they contribute to open source projects. When Stallman designed the first open source license, the GPL, he indeed designed it so that any software that uses it as a license is open to modification, and so that any larger software project that incorporates code issued under the GPL is automatically bound by the GPL and cannot be thus commercialized. However, if code is protected under a different license such as BSD and the BSD license, developers can then use that open source code as a foundation for proprietary products. Understanding the overlap and incompatibilities among different licenses is therefore critical for firms
that seek to harness an open source project or mix-and-match open source code (Woods & Guliani, 2005).

The licensor of open source software is always free to license the software under some conditions, which means that a potential licensee who prefers to accept the software under a different license than an open source one—and who is willing to pay for that “advantage”—may contact the licensor to determine if the software is also available under a different license (Rosen, 2004).

Supporting services are not the only way that an open source developer can make money. They can also make money from licensing the open source products, by picking an appropriate open source license for their products. The Open Source Initiative (OSI)\textsuperscript{22} approves licenses as “open source” based on conformance to a set of criteria known as the Open Source Definition (OSD). Currently, more than 60 licenses are approved as open source by the OSI. Of course it’s impossible to cover all of these licenses, thus we will only talk about the most popular licenses that are used in the software industry shortly.

It is important to mention that the “GPL-ness” of the open source license determines what focus firms should have for generating revenue streams for their open source products. If the GPL-ness is strong (copy-lefted) then the firms should better make money off services, and if the GPL-ness is weak (copy-righted), then it is better to make money off products.

6.1.1 The GNU GPL or Copy-left License

The fundamental and most commonly used license, the GNU GPL, or the “Copy-left” license, is administered by the Free Software Foundation (FSF)\textsuperscript{23}, founded in 1985 to promote the development of the GNU operating system and free software in general.

Copy-left is a license that permits people to freely copy, modify, and redistribute software as long as they do not keep others from also having the right to freely copy, modify, and redistribute the software. Copy-left provisions in a license require that anyone modifying the software can distribute only their modified versions under the terms of the open source license they originally received with the software\textsuperscript{24}. A firm can actually sell a “copy-lefted” open source product, but it must also offer the source for free, either with the product or available for free to anyone on request.

GPL or Copy-left creates the agreement in which company or a person may have this free software on condition that any derivative works that they create from it and distribute must be licensed to all under the same license. For a work to be made available under the GPL, the component parts must be available to all subsequent licensees. For any software containing components licensed by others, the GPL can’t affect the licenses to those component parts (Rosen, 2004).

The term “viral licensing”, which the Free Software Foundation rejects, was invented to describe the effects of the GPL—new code based on GPL-protected work is “infected” by the terms of the license, with no allowances for the size of the code involved or its relative importance to the derivative work. Thus, mixing copy-lefted code with proprietary software or attempting to distribute a closed, proprietary work based on such code is a violation of the

\begin{itemize}
\item \textsuperscript{22} http://www.opensource.org/licenses
\item \textsuperscript{23} http://www.fsf.org/
\item \textsuperscript{24} http://c2.com/cgi/wiki?Copy-left
\end{itemize}
GPL that the lawyers of Stallman’s Free Software Foundation and the GNU community will enforce (Woods & Guliani, 2005).

### 6.1.2 The Lesser GPL (LGPL) License

LGPL was designed as a compromise between the strong “copy-left” GNU GPL and “permissive” licenses such as the BSD or Apache licenses. It is a less restrictive version of the GPL for use with code libraries that are used in precompiled binary form and do not require access to source code, where the majority of the primary GPL’s conditions still stand (that the source code of modified libraries must be made freely available) but the LGPL allows open source applications to fit more snugly into a preexisting product development process (Woods & Guliani, 2005). The LGPL places copy-left restrictions on the program itself but does not apply these restrictions to other software that merely links with the program.

### 6.1.3 The Berkeley Software Distribution (BSD) License

BSD License has been published for the Berkeley Software Distribution (BSD, or Berkeley Unix), which is a UNIX operating system derivative developed and distributed by the Computer Systems Research Group (CSRG) of the University of California, Berkeley. There exist numerous descendants of the BSD today, such as FreeBSD, NetBSD, OpenBSD or DragonFly, all of which are under the BSD license.

BSD allows for uncontrolled borrowing of source code for other software projects or commercialized versions of BSD itself, without any condition for the sharing of source code, modified or not (Woods & Guliani, 2005). The BSD license uses variations of an open source license that is short, simple, and generally GPL compatible and still allows proprietary products to be built and sold from BSD code without requiring the source code to be given away, thus it is GPL-incompatible. Apple, for example, used the BSD license to incorporate code from FreeBSD into Mac OS X.

### 6.1.4 Dual and Multi Licensing

Dual licensing is the practice of distributing open source software under two different sets of terms and conditions. One option is a proprietary software license, which allows the possibility of creating proprietary applications derived from it, while the other license is a copy-left license, thus requiring any derived work to be released under the same license. The copyright holder of the software typically provides the free version of the software at little or no cost, and profits by selling proprietary licenses to commercial operations looking to incorporate the software into their own business.

Dual licensing also provides license compatibility, allowing code from differently licensed free software projects to be combined, or to provide users the preference to pick a license. Examples of these include GNU General Public License (GPL) and GNU Lesser General Public License (LGPL); Perl, which is dual-licensed under the GPL and Artistic License, and Ruby, whose license contains explicit GPL dual licensing.

However, confusion may arise when someone outside the firm or a competitor creates additional source code, using the less restrictive license. Since the firm with the official code is not the copyright holder of the additional code, they may not legally include this new work in their more restrictively licensed version.
Sometimes firms may choose to use the commercial license initially but then decide to alter their business model and open source their product, because for example the firm’s focus might have shifted away from software development and towards the consultancy market. Anyhow, for a specific type of business model, dual-licensing can be an important part of a company’s marketing armory.

Current implementations of the dual licensing strategy include MySQL AB’s database, Oracle Corporation’s Berkeley DB and Qt Software’s Qt development toolkit. Apart from dual licensing, firms can also pursue a multi-licensing strategy, combining more than two different open source licenses in their licensing strategy. A popular example is the tri-licensing model of the Mozilla Corporation, which combines the Mozilla Public License (MPL), the General Public License, and the Lesser General Public License (LGPL) to license certain software in an effort to address the issue of incompatibility with other open source licenses.

6.1.5 Versioning

Apart from multi-licensing or dual-licensing, firms can also decide to offer two versions of the product, an open source version that offers basic functionality and a proprietary version that offers an enhanced feature set. This allows people who require only the baseline version of the product to use the open source version while customers with more complex requirements can purchase the proprietary version. Often, some of the additional features or customizations of the proprietary version make their way into the open source version. The proprietary version also helps pay for the development of the open source version in this strategy. Typical examples include SugarCRM and Sendmail.

6.1.6 Choosing the Right Licensing Strategy

For many commercial companies, selecting the proper license to use often faces on one or both of the following issues (Rosen, 2004):

1. How can we make money from distributing this software under an open source license? In other words, can our license help us sell free software?

Licensor can make money on what the open source license doesn’t grant, thus it is often more rewarding to consider the exclusion from license rather than the open source grants of license for gaining profit out of it.

The most important factor is trademark or brand identity. Trademarks are excluded from all open source licenses. By marketing software under a trademark, the licensor can sell perceived benefit even though the software might be available somewhere else for free without the trademark. Customers are often willing to pay for brand-name software, especially if it comes with support while firms with copyright notices in valuable software make a noticeable benefit because of their professional reputation being enhanced by those contributions. They essentially sell themselves and their expertise, rather than their software. This is particularly important for firms with distributor business model.

2. How can we prevent others from making money unfairly from our open source software?

A protection and appropriation means independent of the type of licensing is provided by complementary assets (Teece, 1986). In the present context, this may be a brand name, as
for distributors such as Red Hat and SuSE, a developer community committed to the respective firm and its software, proprietary software specific to the OSS in question; or hardware to which the respective OSS is tailored (Henkel, 2006).

All licensees are free to copy and create derivative works without payment of royalties to the licensor, and so a licensee can make as many copies of such software as possible without returning the licensor any benefit, which results in a “free rider” problem in open source development (Rosen, 2004). If after considering open source models a company still wants to prevent free-riders, it should consider adopting one of the non–open source licenses.

6.2 Focus on Services/Deployment: Open Source Business Models

6.2.1 The Distributor (Support Seller) Model

Service is also important for open source products as sale of software alone is insufficient to sustain. Once a software product has been released, it is supported with patches and bug fixes as needed. Service and maintenance costs lie roughly between 15% and 25% of the original licensing fee in annual support costs of commercial software. In this sense, distributors of OSS play an important role, by the providing free access to the source code and the software product developed by the distributor firm. For example, in the case of LINUX, leading distributors include Red Hat, Caldera and SUSE.

Distributors make money in three ways (Krishnamurthy, 2003):

i. **Providing the product on CD rather as an online download:** Most people are not comfortable with downloading the product from a web site. Therefore, there is money to be made selling the product in CD form.

ii. **Support services:** Apart from just selling the product, there is still money to be made in services such as support for installation, answering technical questions. The model is based on the premise that the creators of a software are the best suited to provide support because they are the creators.

iii. **Upgrade Services:** By acting as application service providers, distributors can help their clients get the latest version of the product seamlessly and updates on the product.

6.2.2 The Non-GPL (Added-value Provider) Model

Some companies take software distributed by open source projects and use it in their own open source products, which they then distribute. For instance, operating system like Linux and a web server like Apache can be used in a commercial product like a cell phone or a television recorder (Rosen, 2004). This case conforms to the “complements” part of the focus on services and deployment, where firms sell a PDA, cell phone, or other device at a profit that runs an open source application software suite or operating system. The Non-GPL model in this case becomes the “Widget Frosting Model” since the OSS product provides high-quality, low-cost software components that reduce the cost of the solution without reducing the revenues of the firm.

Software producers can also incorporate the source code of an existing product in a larger code base and create a new product, or they can also take an entire open source product
and bundle it with existing products, where the source code for the derived product does not need to be disclosed since the license is not GPL (Krishnamurthy, 2003). Firms must be extremely careful here in terms of licensing strategy, as explained before. They should either incorporate GPL-incompatible licenses or pursue the multi-licensing approach. Problems would arise with GPL and GPL-variant licenses.

As an example, Microsoft has incorporated the code from BSD in its products and has not released the source code to any interested party, but it had to do was to acknowledge that it benefited from BSD’s code. In this case, the source code for the original product is still available to end users via communities.

### 6.2.3 The Pure Open Source (GPL) Model

Unlike the Non-GPL model, software firms are furthermore forced in the GPL model to make the source code for the derived product available to the end user for free. The release of the source code in this model accelerates innovation due to more rapid feedback and input. However, it does expose the inner workings of the company’s product to the users (Krishnamurthy, 2003). Successful GPL models are usually based on large and very important applications. Since firms cannot capture economic returns from the product itself, they need other ways to get revenues to survive. Similar to the distributor model, there are a number of possible ways to achieve this:

i. **Consultancy or Support Services**: Either for the product that the firms are producing or the in-house skill set of the firm that is utilized to produce the product.

ii. **Charging for plug-ins and similar proprietary extensions**: This is a popular approach, however it carries a potential danger of someone else or a competitor deciding to distribute a similar one for free, so this approach is not likely to be sustainable.

iii. **Charging on a subscription basis for upgrades and add-ons**: Same risk as having others redistribute.

iv. **Spinoffs to vertical marketplace**: Contributors moving from a proprietary licensing scheme to a pure GPL will probably sell fewer copies, but is their business really all about selling proprietary extensions? If this is not the case a vertical marketplace covering other business areas would help the firm create revenue streams.

v. **Networking**: IT firms and individual developers involved with the product could start a business partner network.

vi. **Re-coding proprietary extensions**: Favored or popular proprietary extensions could be re-coded through allocating funds raised by networks or product-specific communities (such as OpenSourceMatters[^25] for Joomla!), if feasible.

vii. **Education**: Firms with pure GPL model can charge their customers for giving training and education on installing, configuring and managing the product.

[^25]: http://opensourcematters.org/home.html
6.2.4 The Third-Party Support Seller Model

Third party service providers simply don’t care where the firm got the code or where they got the product from. If the product that firm is using meets a broad set of criteria, they will fully support it. They have just single revenue stream—service (Krishnamurthy, 2003).

This model is essentially the same as the distributor model with the exception that the firm offering the support services is not the owner (copyright holder) of the supported open source software. The firm need not even be part of the core development group of the OSS product. The model is often used when the OSS product is either not owned by any one developer or development company, or when the owner of the source code is not able to provide professional support services (e.g. when he is an individual rather than a corporation).

6.2.5 The Platform Provider Model

The company bundles several open source products into a complete solution or platform and provides quality-assurances that the selected products work together. Firms such as Red Hat and Novell that provide commercial Linux distributions use this model. This model is usually combined with the Third-Party Support Seller Model, because it is much easier to support and bug-fix a complete solution (platform) as it implies greater control over the operating environment. Furthermore, the model enhances the value delivered to the customers if firms can source the platform and related support services for the same supplier (Daffara, 2007; Krishnamurthy, 2005).

6.2.6 The Software-as-a-Service (SaaS) Model

SaaS models are those where the OSS product is used to create a web-accessible application service. Sophisticated application functionality is provided on a low-cost basis. The SaaS model does not require any IT administration or server hardware resources, thus is beneficial for customers in terms of maintenance. For SaaS, firms need to develop the software services and must be able to build a strong brand if they want the model to be successful (Koenig, 2004). The best known company that uses this model is Google.

6.2.7 The Accessorizing Model

In this model, firms sell physical accessories for the OSS products. Most important of these accessories are technical books and manuals. The central value here for customers is the access to first-rate technical information about the product. The often chaotic nature of OSS documentation makes these books of great value to users, especially if the firm is among the core developers of the product (Hecker, 1999).

6.2.8 The Loss Leader Model

In this model, a software company provides an OSS product as loss leader and market demand for the proprietary product, thus OSS is not essential for the loss leader strategy to capture market share. The proprietary product offers additional high-value functionality. The idea is that customers who need these functions are prepared to pay for it. The OSS nature of the loss leader helps widespread distribution and helps building a brand for the company and the proprietary products within the software professional community (Hecker, 1999; Daffara, 2007).
6.2.9 Hybrid Models

The existence of commercial subjects using OSS guarantees the future survival of the software. Thus, using hybrid business models relax the constraints surrounding open-source. For example, a company might use both traditional licensing and open-source-like licensing “side by side” for the same product, differentiating between users or between different types of use, or it can license source widely to any and all users, and even allow “evaluation” licensing at no charge, but still charge “right-to-modify” license fees and restrict redistribution of modified versions in some way (Hecker, 1999). These business models are not pure open-source models, though.

Hybrid models have been characterized by firms producing open source software, giving it away to customers free of charge and shifting the value from licensing agreements to additional services such as packaging, consultancy, maintenance, updating and training (Bonaccorsi & Rossi, 2003).

Firms that have been working with commercial models have decided to go with variants of the hybrid model for their own interests. Software giants, in particular, realized the “inevitable coexistence” of open and closed innovation under the same roof, and thus have decided to implement an alternate strategy, via making their products compatible with open source platforms, releasing the code of some programs and even taking part in Open Source projects. Apple, for example, has released the source code of the Mac Server OS X, while Sun produces the Open Source suite StarOffice in order to compete with Microsoft’s Office. Red Hat provides NASA with software for space missions, puts together and compiles the source code of the Linux operating system obtaining a program ready for installation, which is distributed on a CD at less than 100 dollars together with a set of complementary applications (for instance the Gimp program for managing and changing images) and a complete documentation. Red Hat also sticks to the pure open model, releasing the “Fedora” Linux distribution for free download, mainly for end-users. This is a good example of firms that do not stick to a single model, but pursue different models for different target customers and applications. The case of IBM that has been illustrated before is another example.

Hybrid business models are made possible by the proliferation of licensing agreements that follow the dictates of Open Source while not following the more extreme GPL format. Apart from GPL, which is the most famous, yet the most restrictive license, there are a wide set of licenses. Some of them, such as the BSD (Berkeley Software Development), allow the appropriation of software modifications. Moreover, other licenses, which cannot be considered Open Source given that they protect copyright, provide the release of the source code that particular categories of users are allowed to use (Bonaccorsi & Rossi, 2003). This is the case of the Apple Public Source License by Apple and of the Sun Community Source License of Sun.

6.3 The R&D Management Strategy

West and Gallagher (2006) categorize R&D management strategies for open source software into four groups:
### R&D Management Strategies in OSS

<table>
<thead>
<tr>
<th>R&amp;D Management Strategy</th>
<th>Examples</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pooled R&amp;D</td>
<td>Mozilla Firefox</td>
<td>Donate IP to the open-source project while exploiting the common benefits of all contributors to facilitate the sale of related products</td>
</tr>
<tr>
<td>Spinouts</td>
<td>Jikes, Eclipse</td>
<td>Release more value from their technologies by situating them outside the firm, while at the same time maintaining an ongoing corporate involvement</td>
</tr>
<tr>
<td>Selling Complements</td>
<td>Apache, Apple Safari</td>
<td>Providing installation, training and support services as complements to a free core product, as would the many other firms that sell services for ‘free’ software.</td>
</tr>
<tr>
<td>Donated Complements</td>
<td>Half-life, Counter Strike</td>
<td>Make the money off of the core innovation but seek donated labor for valuable complements</td>
</tr>
</tbody>
</table>

**Table-2: R&D Management Strategies in OSS**

### 6.4 The IP Management Strategy

In traditional closed business models, firms handle internal intellectual properties (IP) as backwater activities, protected by legal specialists because any external IP resource is considered risky and unreliable (Chesbrough, 2006).

To license their intellectual property, innovators must first gain some form of legal protection for it. In most subject matters, the relevant form of legal protection is the patent grant. An alternative form is copyright, which is a low cost and immediate form of legal protection – it is applicable to many forms of original writings and images (von Hippel, 2002). Furthermore, open source software can co-exist with commercial software that is protected by user licenses, copyright and trade secrecy. Thus IP management becomes a critical issue for firms pursuing hybrid models.

Several leading IT firms have contributed their own patents to the open-source software development process. Novell, Computer Associates and IBM, among others, have helped to create a “patent commons” by contributing thousands of patents from their own large patent arsenals— arsenals that were created to generate revenues from licensing and provide either offensive or defensive weapons in intellectual property battles (Maxwell, 2006).

Software is better positioned than most other fields with respect to the feasibility of establishing a modest level of legal protection that can support the licensing of intellectual property. This is due to the legal status of software as “writings,” which can be protected by copyright. Licensing of copyrighted software is widely practiced by commercial software firms. When one “buys” a copy of a non-custom software product one is typically actually buying only a license to use that software rather than buying the intellectual property itself. Licensing is also the basis of free and open source software practice (von Hippel, 2002).
Table-3 below summarizes major IP strategies used in open source firms (Dahlander, 2004):

<table>
<thead>
<tr>
<th>IP Strategy</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Focus on IP rights</td>
<td>Patents and Copyrights that grant creators exclusive right to reproduce, prepare derivative works, distribute, perform and display the work publicly</td>
</tr>
<tr>
<td>Secrecy</td>
<td>Keeping secrets within the firm, primarily by closing the code</td>
</tr>
<tr>
<td>First-mover advantages</td>
<td>Network externalities</td>
</tr>
<tr>
<td>Complementary assets</td>
<td>Distribution, marketing in conjunction with the innovation</td>
</tr>
</tbody>
</table>

Table-3: IP Management Strategies used in OSS

6.5 The Networking Strategy

The table below shows major partners that open source developers collaborate with during the product development process.26

<table>
<thead>
<tr>
<th>Type of Partner</th>
<th>Main Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Integrator, SI</td>
<td>Integrates system security solutions.</td>
</tr>
<tr>
<td>Independent Software Vendor, ISV</td>
<td>Makes or sells software, designed for mass marketing or niche markets.</td>
</tr>
<tr>
<td>Independent Hardware Vendor, IHV</td>
<td>Makes or sells hardware, designed for mass marketing or niche markets.</td>
</tr>
<tr>
<td>System Developer, SD</td>
<td>Contributes to the overview of projects on application level and focuses on software development processes.</td>
</tr>
<tr>
<td>Original Equipment Manufacturer, OEM</td>
<td>Sells/uses the firm technology as their own brand or component of their own product.</td>
</tr>
<tr>
<td>Internet Service Provider, ISP</td>
<td>Sells/uses the firm technology as their own brand or component of their own product.</td>
</tr>
</tbody>
</table>

Table-4: Classification of potential partners of open source firms

6.6 Forming the Open Source Strategy

Whether open source will work at any company depends on both the capabilities of the company and the maturity of the open source software. Not all the users of open source are equal, compared to the massive IT budgets of large corporations such as Amazon, Google, Yahoo! and IBM.

26 http://www.primekey.se/Partners/Becoming/Types+of+partners/
The difference between the successful open source implementation, in which the value of open source is realized for a company, and the unsuccessful one, in which the struggle to use open source is not worth the effort, amounts to knowing your problem, knowing the software, and knowing yourself (Woods & Guliani, 2005). Thus, companies should approach open source development in terms of open source maturity, in-house skill sets and open source-specific costs. Any firm that intends to adopt or integrate open source should not only have the adequate resources to do so, but also a belief in in-house skills building and tendency to take increased responsibility for the corporate IT infrastructure. An open source project which is immature or ill-suited to the in-house skills may not be realized at the beginning of the product development process, but would be disastrous as the process evolves.

If a firm wants to use or is already using open source, it must learn to evaluate the software’s maturity and the level of support provided by the community that surrounds the project in order to understand the risks associated with open source development.

The open source strategy needn’t rely strictly on open source platforms (such as Linux or BSD), it can also be an OSS application running on proprietary platform such as Microsoft Windows.
Firms should not only select the right open source strategy, but they should also apply this strategy to right problems. They should create innovation processes or maybe modify the established processes in order to manage the introduction of OSS to the organization.

Finally, open source comes with unique risks that are not present in commercial software. Firms thus need a governance structure to manage the OSS development and avoid potential chaos. The firm size and the way the firm uses the OSS have an influence on these structures.

6.6.1 Skill-Set Analysis

Benefits of the open source depend on the level of the skill-sets of the firm. Four different levels are present: Beginner, Intermediate, Advanced and Expert. For most firms, reaching the intermediate skill level is enough to run most of the projects, unless they want to be leaders.

Each skill level enables an IT department to handle open source of different degrees of maturity and is described in terms of the following dimensions (Woods & Guliani, 2005):

i. **Open source development tools**
   What knowledge of open source development tools will be required?

ii. **Hosting**
   What hosting capabilities will be needed?

iii. **System administration**
    What system administration skills will be required?

iv. **Operations**
    What operations capabilities and skills will be needed?

v. **Open source infrastructure**
   What open source infrastructure (e.g., Apache, MySQL etc) must be understood?

vi. **Programming languages**
   What programming language skills will be required?

vii. **Open source community skills**
    What sort of skills for getting help through open source community channels will be needed?

Firms that are in the beginner level first need to understand how the components of the OSS project work, by looking at the source code. Determination of the skill level of the firm is important, because it helps firms introduce product complexity gradually. Otherwise, firms that have recently started open source development and are thus in the beginner level would face turmoil as the development process goes on.

Firms should form a team responsible for the open-source efforts of the firm, usually within the IT department. They should furthermore be careful in having their own developers be “customers” of that infrastructure, in principle no different than any other developers for optimum efficiency of the development process.
It should be noted that it is impossible even for a large IT department to be skilled in a small fraction of OSS, thus firms should “lock-in” to a specific technology or set of components, languages and projects in their open source development strategy. The in-house skill level is important even when firms engage in consultancy support from other firms, because the consultancy firms help them access their own skill-sets.

The decision to use open source should be made carefully in the context of how critical a system is to business operations. The more important the system, the more skills are required to support an open source project. The importance of the IT system to the firm is classified into four different levels of importance: Experimental, Low-priority, Operational and Mission-Critical. The two extreme cases are the experimental and mission-critical systems. Experimental systems are not expected to work in the first place and are the least critical systems; whereas mission-critical systems must work to keep the business running where downtime is not acceptable and failover plans should be designed.

Taxonomy of the IT system is also important, whether the system being deployed with the OSS product is stable, flexible or dynamic affects the implementation and thus requires different levels of skills for the firms.

6.6.2 Maturity Analysis

Firms have to spend time gathering information and understanding the open source software that they are using in their products. In other words, evaluating open source thoroughly enough to reduce risk is an important work for software firms.

If some of the key elements of maturity are lacking in the OSS, problems during the development will be hard to address, help will be hard to find and customization & configuration of the product will be difficult. If the elements of maturity are present, however, the software will be much easier to handle for the firms.

Many firms are so blindly attracted to open source that they pass over adequately considering the issues involved with open source maturity, simply because it is exciting, it represents an interesting opportunity for the technologist and a way to save money for the business.

Firms should be careful in selecting the right open source product; otherwise they would eventually fall into “open source traps”. Avoiding such traps requires discipline. A maturity analysis discipline would help firms to overcome such traps, helping them choose the best alternative from various open source projects before green-lighting the implementation, and even during implementation process (Woods & Guliani, 2005):

   i. **Organizational Structure**
      
      Firms should provide the necessary infrastructure for external developers (newsgroups, source code repositories with revision control, special bug-reporting systems).

   ii. **Vitality of community**
      
      How big is the community? How active are the forums on the product? How many downloads happen, and how often? How frequently is the project referred to on major search sites such as Google?

   iii. **Quality of end-user support**
The most crucial elements in terms of saving time when using an open source project are active forums, well-maintained FAQs, and documentation that are available through a search engine, and are generally the biggest time-saving feature of an open source project.

iv. **Momentum (Frequency of Releases)**

How often new releases of a program become available might help or hinder its effective use.

v. **Quality of code and design**

Have the authors organized the code in a way that invites understanding, and that reveals at least some organization? Is the code modularized? How are the modules grouped together? Has a naming convention been rigorously adhered to?

vi. **Integration with other products**

Is the OSS project aware of inter-dependencies between different software ecologies? Are the new releases of the product proactively tested with other open source software products?

vii. **Quality of project site**

Is the official project site concise and well organized?

viii. **License type**

How is the access to source code maintained? Does the license allow firms to distribute the product or modify the product and re-distribute?

ix. **Potential for commercial conflicts**

The OSS code might infringe on the intellectual property of a commercial company, putting the OSS developers as well as users at risk of potential legal action by the commercial suppliers.

### 6.6.3 Open Source Adoption: Productization

*Productization* means making software work for the general case and making it as easy as possible to use and it requires a huge amount of work which can take double or triple the amount of work it took to complete the original features and turn a program into a product (Woods & Guliani, 2005).

For highly skilled IT departments, the cost of overcoming the productization gap is small. From this perspective, the barrier to wider adoption is not the lack of productization, but the lack of skills in those who desire to use open source, or the skills gap.

Productization can be a revenue stream for firms with at least intermediate level of skills. It can help avoid the skills gap of firms, through installation scripts, configuration tools, administrative interfaces and diagnostic consoles. It can also help firms accelerate their learning process, through examining the source code, reading the API and architecture documentation and similar manuals & technical documentation. However, it is a boring and long process which is broadly ignored by most open source developer firms.

### 6.7 Summary

In chapter 3 we discussed various components of the innovation strategy, and classified them as being internal, external and firm-specific, explaining each class of components in
Later, in chapter 4, we added the point of open innovation to the discussion and showed that openness of the innovation activities in the firm is another important element in shaping the overall innovation strategy. Finally, in this chapter, we have seen how various factors explored in chapter 3 and 4 reincarnate specifically in open source strategy, not for any firm, but for firms that work with open source software.

We have seen that R&D management, IP management, networking and business model components from the innovation strategy again come into consideration when one tries to determine the components that are specific to open source. We have seen that depending on whether the firm focuses on product development or services/deployment, licensing or the business model comes into consideration for determining the open source strategy. We have also explained how these components from innovation strategy can be specifically implemented in an open source strategy.

On the other hand, some new elements were explored that did not show up during the discussions for innovation strategy. Licensing, versioning and open source business models are examples of such open source-specific components that make up a firm’s open source strategy.

Following in the footsteps of Figure-3 and Figure-4, we conclude with the following framework of open source strategy, which we will use during analysis. The open source strategy framework in Figure-5 shows clearly the layers of an open source strategy: the innovation strategy, openness coming from open innovation side of open source and open source-specific elements such as skill-set analysis, maturity analysis, adoption (productization), the IT infrastructure and integration (platform/application based).

Figure-5: The resultant theoretical framework showing the layered structure of open source strategy, along with the components that make up the open source strategy at each layer (Authors’ construction)
7 Methodology

7.1 Deductive vs. Inductive Research

Research is a systematic process of collecting and analyzing information in order to either increase or get an understanding of a phenomenon that is desired to be understood and thus is under study. The systematic nature of the investigation is essential for the efficiency of the research process and thus research should be carried out using proper research methods to collect and analyze data for the optimal use of opportunities and available resources and efficiency of the research process. Research methods refer to systematic, focused and orderly collection of data in order to obtain information from them to solve the special research question (Ghauri & Granghaug, 2005).

In research, we often refer to the two broad methods of reasoning, as the deductive and inductive approaches. In deductive research a conceptual and theoretical structure is developed and tested by empirical observation where as in inductive research theory is developed from the observation of empirical reality. Thus, deductive research is referred also to as “top-down” or “waterfall” research, whereas the inductive research is referred to as “bottom-up” or “hill-climbing” approach, as can be seen from the figure below. In our study, we are combining both approaches; we are using the deductive approach to extract useful information from our interviews and literature review, whereas we use the inductive for building our layered model (as will be seen in chapter 6.7.) from the strategy components that we discussed in chapter 3 and 4.

![Figure-6: Deductive vs. Inductive Research](image)

In any kind of research process there are some basic stages: identifying the research topic, formulating the research problem, determining how to conduct the research, collecting the data, analyzing the data and drawing the conclusion (Collis & Hussey, 2003).

7.2 Qualitative Data Collection

The two main data collection methods in a study are the qualitative method and quantitative method. The qualitative method can provide more details and understandings when an event or social process is difficult to study with quantitative method. This research method is suitable to study social and human behavior science and study the organizations, groups
or individuals. Qualitative method of collecting data furthermore allows the researcher to explore the attitude of brands, trends and behaviors. Using this method in some cases such as cross-cultural research provides better understanding of a given context, values and attitudes. Data collected from small samples could be enough, and it is more flexible and unstructured compared to the quantitative method. By employing small samples for data collection could be enough to explain several and different aspects of the research problem. Therefore, qualitative method is the most useful when the purposes of the study require in depth insight into the phenomenon (Ghauri & Granhaug, 2005).

In our study, we therefore have decided to go with the qualitative research since we want to investigate the different kinds of innovation strategies used in software firms; in particular their strategies regarding open source development, thus we are not concerned with any quantitative data. We carry out interviews and/or questionnaires with these firms to have an insight of the structure of their innovation strategies to be able to identify the issues stemming from the literature and to see if firms address these issues in their own strategies and which issues they give more importance to from their own point of view. Qualitative research is especially effective in obtaining culturally specific information about the values, opinions, behaviors, and social contexts of particular populations; thus choosing it as a foundation for our research will help us obtain firm-specific information about the values and opinions of firms regarding open source strategy.

Interviews require real interaction between the researcher and the respondent and it can be done via email, telephone or in person. Open-ended questions and probes yield in-depth responses about people’s experiences, perceptions, opinions, feelings, and knowledge in an interview (Patton, 2002). Given that personal contact is mandatory with interviews, it is often easier to obtain a higher response rate using this technique than others such as surveys or questionnaires (Williamson, 2002). To be able to carry out the interview efficiently, the researcher should know the respondent’s background, values and expectations (Ghauri & Granhaug, 2005).

There are two main types of interviews, structured and unstructured. In structured interview each respondent is asked exactly the same questions, with the sequence of questions also being same. It is simply a survey questionnaire administered by interview in this sense. Structured interviews are used where it is important to be able to compare results across respondents, with flexibility for individual expression within a more formal structure (Williamson, 2002). Unstructured interviews, on the other hand, refer to face to face verbal exchanges in which the interviewer tries to obtain information or opinion of interviewee by asking question where respondents are free to answer the according to their own thinking so the answers are not constrained by only a few alternatives and in the case of ambiguity the interviewer can ask for further elaboration (Ghauri & Granhaug, 2005). Unstructured interviews are useful for exploring a subject or gaining insights into key people to collect extensive data especially in case studies, and for an effective strategy for countering memory failure or respondent resistance (Ruane, 2005).

---

There exists also a third approach, which is a compromise between these two, called the *semi-structured interview*, which includes a standard list of questions but allow the interviewer to follow up on leads provided by participants for each of the questions involved (Williamson, 2002). They are actually closer to the unstructured than the structured approach.

An extremely popular variation on the one-on-one personal interview is the next best thing to “being there” – the telephone interview, which is a “technology-dependent” technique that sees the interviewer questioning respondents by phone and recording their answers (Ruane, 2005). This method is a relatively fast option and it is much more economical than personal interviews and they are also less intrusive or threatening than in-person interviews of “opening the door to a stranger”.

A good interview requires considerable thought and preparation, not only with regard to the questions to be asked, but also in relation to a wide range of issues which may have effect on the overall quality of the data collected and thus the quality of the research (Williamson, 2002). The first important issue in any type of interview is to introduce the research and the purpose of the study to respondents (Ghauri & Granhaug, 2005). The interviewee should be given full freedom to express personal answers while the interviewer is controls the situation in order to get relative information in a certain time. The efficiency of the interview strictly depends on the mutual relationship between interviewer and interviewee.

Interviews could be carried out in various ways, with the two popular ways being phone and face-to-face interviews. Phone interviews have the advantage of saving a lot of time since the interviewer does not need to travel and meet the respondents face-to-face, thus a large number of different people could be reached in a relatively short period of time. It also eliminates any discomfort for the interviewees that some of them would feel less uncomfortable disclosing personal information over the phone than face to face. On the other hand, face-to-face interviews enable researchers to adapt the questions if necessary, clarify doubt and ensure that the responses are well-understood by repeating or rephrasing questions. Furthermore, the researcher can pick up nonverbal cues from the respondent, any discomfort or stress the respondent experiences can be detected through facial expressions and body language which in turn will give the interviewer a clearer indication of the candidates true, honest feelings/ emotions. The interviewee is also more likely to provide the interviewer with more information in terms of ‘open-ended’ questions as they may feel more comfortable speaking just face-to-face. Yet, they are extremely time-consuming, especially if the researcher is supposed to carry out a large number of interviews.

### 7.3 Validity

According to Yin (1990), internal validity certifies if the empirical findings could be valid for the research question. Internal validity also considers the fact if the researchers have been investigated what they were supposed to. Internal validity can be improved by using multiple sources. However, it is not easy to reach a concrete measurement of validity, but looking deeply at interviews ‘questions show that how that the researchers are investigating the right area and the interviews really are providing the information needed to reach the aim of the thesis.
In our thesis we used multiple sources of data. Both documentation and interviews were used to collect data. The interviews’ questions were reviewed by our supervisor. We also used a recorder during the interview to secure that we did not miss any information or misunderstood any answers.

Yin (1990) furthermore defines external validity of the study is defined as generalizing the result of the study. He also mentions that a researcher should be careful with generalizations when a qualitative research is conducted and when it comes to case studies it is impossible to not be critical against generalizations since the research only focuses on single case or a number of cases. In addition, in order to obtain high external validity there must be a high internal validity.

This thesis is based on of 8 open source software companies in Sweden. In this study, it is hard to generalize the findings but within reasonable limits conclusion can be drawn. However, generalization of findings is not the main goal of this thesis. The purpose as it is mentioned in purpose section is not to draw statistically valid conclusions, but rather to understand, get insights to open source strategies.

### 7.4 Reliability

Reliability of the research means that if an investigation is repeated exactly in the same way, it should generate the same results (Yin, 1990). Reliability measures the concordance, consistency or repeatability and accurateness of outcomes of the data collected (Haas, 1991). In qualitative investigation, it is hard to achieve high reliability since the respondents in the interviews might have changed their behaviors or ideas in different situations. Since the study has been done in a certain situation any change in condition may causes to new results. On the other hand, in theory part the same result should be created if the same research was conducted in completely different time periods. However, in our study in order to achieve high quality of research following actions have been taken during the data collection process:

- Interviews only persons which are very closely related to the research subject to ensure most accurate answers;
- Clear formulation of interview questions in order to prevent from misunderstandings and wrong answers;
- Use of recording device in order to accurately collect all the responses and later transfer the full meaning of the answers without losing precious information;
- Clarification of questions when asked by interviewers and during the interviews;
- Constantly checking the consistency and accurateness of the information with literature, theory.
7.5 Empirical Findings

In order to filter out a potential large number of samples and also to pick as relevant and appropriate firms as possible for our study, we formed a firm database of 25 companies which originates mainly from the member companies of the Open Source Sweden\(^\text{28}\) organization, with exceptions that also heavily or entirely use open source, but are not a member of the organization yet. The sample of firms selected for our database is listed in Table-4 with corresponding geographical location.

Some of these companies base their business strategy solely on open source business models, which we define as “pure” open source strategy; some combine open-source models with their own proprietary business models to form a “hybrid” strategy whereas others continue to follow the traditional “pure” proprietary model. There exist also companies that develop products with pure open-source strategy and also products with hybrid strategy, as well as companies that develop products with “non-pure” open source strategy by providing free access to the software but not allowing modifications or re-distributions of the source code.

Table-5: The preliminary firm database used in our study

<table>
<thead>
<tr>
<th>Firm No.</th>
<th>Firm Name</th>
<th>Open-Source Sweden Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Op5</td>
<td>Member</td>
</tr>
<tr>
<td>2</td>
<td>Celona</td>
<td>Non-member</td>
</tr>
<tr>
<td>3</td>
<td>Cendio</td>
<td>Member</td>
</tr>
<tr>
<td>4</td>
<td>Curalia</td>
<td>Non-member</td>
</tr>
<tr>
<td>5</td>
<td>Altrusoft</td>
<td>Member</td>
</tr>
<tr>
<td>6</td>
<td>Kentor</td>
<td>Member</td>
</tr>
<tr>
<td>7</td>
<td>Init</td>
<td>Member</td>
</tr>
<tr>
<td>8</td>
<td>Omicron Ceti</td>
<td>Member</td>
</tr>
<tr>
<td>9</td>
<td>Lokomo Systems</td>
<td>Member</td>
</tr>
<tr>
<td>10</td>
<td>Roxen</td>
<td>Non-member</td>
</tr>
<tr>
<td>11</td>
<td>Kerfi</td>
<td>Non-member</td>
</tr>
<tr>
<td>12</td>
<td>HiQ</td>
<td>Non-member</td>
</tr>
<tr>
<td>13</td>
<td>Enea</td>
<td>Non-member</td>
</tr>
<tr>
<td>14</td>
<td>Primekey Solutions</td>
<td>Member</td>
</tr>
</tbody>
</table>

\(^{28}\) http://www.opensourcesweden.se/medlem/medlemsregister
Of these 25 companies, 8 of them have given positive feedback, yielding a participation rate of \( \frac{8}{25} \) or 32%. The resulting firm portfolio is shown in the table below:

<table>
<thead>
<tr>
<th>Firm</th>
<th>Location</th>
<th>Open-Source Sweden Membership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firm A</td>
<td>Stockholm</td>
<td>Member</td>
</tr>
<tr>
<td>Firm B</td>
<td>Enebyberg</td>
<td>Member</td>
</tr>
<tr>
<td>Firm C</td>
<td>Kista, Göteborg, Malmö</td>
<td>Member</td>
</tr>
<tr>
<td>Firm D</td>
<td>Stockholm, Umeå</td>
<td>Member</td>
</tr>
<tr>
<td>Firm E</td>
<td>Tärsjö</td>
<td>Non-member</td>
</tr>
<tr>
<td>Firm F</td>
<td>Täby</td>
<td>Non-member</td>
</tr>
<tr>
<td>Firm G</td>
<td>Solna</td>
<td>Member</td>
</tr>
<tr>
<td>Firm H</td>
<td>Stockholm, Öresund, Linköping</td>
<td>Non-member</td>
</tr>
</tbody>
</table>

*Table-6: The resultant firm portfolio for our study*
Furthermore, of these 8 companies, 3 of them (3/8 = 37.5%) are members of the Open Source Sweden organization. Two primary types of data sources were collected:

1. **Secondary resources** were gathered on all firms from homepages of the companies and also from Swedish company database Affärsdata\(^{29}\). All information was used to get an idea of the competitive environment, important milestones and the perception of outsiders of the firm. It formed a useful background to later steps and provided us with the possibility of comparison with other data sources.

2. **Semi-structured interviews** were carried out at the firms. Since the geographical location of the companies were quite scattered through Sweden, we have decided to do video-conferencing and phone interviews instead of face-to-face interviews. Video-conferencing was preferred nonetheless, since the interview could be emulated as face-to-face interviews, thus inheriting the advantages of face-to-face interviews, yielding healthier judgments.

One pilot interview was carried out to learn how to use the interview manual and test the relevance of the questions. Each interview lasted about 0.5–1.5 h and included a comprehensive number of questions about the open source strategy and innovation strategy of the firms respectively.

In addition to having the interviews taped, we have also documented the answers through extensive and careful note taking. A draft of the empirical observations was sent to the respondents to ensure that we interpreted the details correctly, and for them to give further feedback.

Data collection for the interview was carried out over a 5-week period, which spanned from 17\(^{th}\) of December, 2010 to 19\(^{th}\) of January, 2011. The survey started with screening questions, with the respondents being asked about the background of their companies and also their personal background and role in the specific company.

We have been careful about balancing the interview questions appropriately between open source strategy and innovation strategy, and provided the interviewees with necessary information since interviewees were coming from various backgrounds and had various roles in their companies, e.g. a CEO could possess much more knowledge within innovation but may not have enough knowledge for open source strategy (especially technical details such as the IT infrastructure of the company), whereas a developer or a CTO could be vice versa.

Since our major focus is on open source strategy, the next section of our interview that came right after the screening questions consisted of open source-specific questions regarding licensing, business model, communities, networking, in-house skill-set, IT infrastructure and productization. The final section was an effort to investigate the innovation activities of the companies under the radar, with various questions addressing internal corporate venturing (ICV), innovation culture, R&D and IP management, openness, core capabilities, product and structural complexity and road-mapping. Our semi-structured interview questions and the structure of the interview we have used during the data collection stage could be seen at the appendix.

\(^{29}\) www.ad.se
We sought senior-level perspectives within the selected organizations. The interviews were held primarily with either CEOs or vice presidents of the companies. If such option was not possible, we then diverted our attention to developers that actually were responsible for developing the open source products. Majority of the interviews were carried out through the phone.

We have experienced a few problems during collection of the qualitative data. First of all, the contact persons that we sent the interview requests in each of the twenty-five companies that were in our initial firm database mentioned that they were too busy in their own projects to be involved in our interview. Some also mentioned that they cannot participate in our interview because of confidentiality issues. This led to the resultant firm portfolio shown in Table-6.

After we had our resultant firm portfolio, we then diverted our ambition to scheduling the interview with all the eight contact persons. This formed another problem, because it was often difficult to contact them via telephone because of their busy schedule (based on the fact that most of them were CEOs or other senior-level executives), and when we contacted them via e-mail, the response time was quite long. We first suggested a specific date and time for the interview to our contact persons and waited whether they were available in that particular date and time. Most of the time they weren’t and they suggested a further date and time, which made the process even longer. It took about two months to finalize an interview schedule with all the interviewees.

As mentioned before, we recorded our interviews which would make data extraction easier and faster during the analysis, but upon completion of all the eight interviews, we realized that the recordings were extremely noisy. This was due to the fact that all interviewees except one chose to participate in our interview via telephone (with one interviewee being interviewed via video-conference). Thus, the speaker of the telephone unfortunately interfered with the microphone of the recorder, resulting in noise. We spent about two weeks further on computer trying to filter out the noise and extracting the answers from the interviews.

A third problem arose with confidentiality. All respondents but one had no objections to referring to their companies explicitly using the name of the company; however one interviewee requested us to hide this information in the analysis. Therefore, even though we could still refer to the firms explicitly, we chose not to. Instead, we shuffled the order of the firms and used ambiguous names such as Firm A, Firm B etc. during our analysis. A copy of the thesis was sent to the respondents along with the information which letter corresponded to their company, in order to check if they had further objections to any of the data used in the analysis.

A final problem was to find a common database for fetching firm-specific data such as turnovers, firm size and firm age. As mentioned before, we used Affärsdata for accessing these data; however these data date back to 2009. Thus, they do not reflect the current numbers.

We have randomly sorted the firms in our portfolio as in the following table in order not to disclose any sensitive information such as turnover. Only the domestic firm size of the firms is considered, in the case of firms with international operations. From now on, we will refer to the firms by their specific firm letters (A to H) instead of their actual names.
As can be seen from the tables below, the firms are all quite small (considering only the domestic operations) and the number of employee’s ranges from 2 to 61, with a mean of 16.37. The turnovers of the firms are also relatively low, ranging from as low as 834 kSEK up to 51.2 MSEK, with a mean of 17.221 MSEK.

<table>
<thead>
<tr>
<th>Firm</th>
<th>Foundation</th>
<th>Firm Size (Domestic)</th>
<th>Turnover (kSEK, 2009)</th>
<th>Role of the Interviewee</th>
<th>Product Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2006</td>
<td>5</td>
<td>3,484</td>
<td>CEO</td>
<td>Pure OSS, Open Source Components &amp; Customization</td>
</tr>
<tr>
<td>B</td>
<td>2000</td>
<td>2</td>
<td>834</td>
<td>Managing Director</td>
<td>Pure OSS &amp; Customization</td>
</tr>
<tr>
<td>C</td>
<td>2003</td>
<td>22</td>
<td>33,257</td>
<td>Software Developer &amp; System Designer</td>
<td>Open Source Components &amp; Customization</td>
</tr>
<tr>
<td>D</td>
<td>1981</td>
<td>61</td>
<td>51,212</td>
<td>CEO</td>
<td>Open Source Components</td>
</tr>
<tr>
<td>E</td>
<td>2003</td>
<td>4</td>
<td>1,969</td>
<td>CEO</td>
<td>Open Source Components</td>
</tr>
<tr>
<td>F</td>
<td>1997</td>
<td>2</td>
<td>1,052</td>
<td>CEO</td>
<td>Open Source Components</td>
</tr>
<tr>
<td>G</td>
<td>2002</td>
<td>11</td>
<td>13,460</td>
<td>CTO</td>
<td>Pure OSS &amp; Customization</td>
</tr>
<tr>
<td>H</td>
<td>1968</td>
<td>24</td>
<td>32,505</td>
<td>Senior VP &amp; Marketing</td>
<td>Open Source Components</td>
</tr>
</tbody>
</table>

Table-7: Elementary information about the firms in our portfolio

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Size</th>
<th>Turnover (kSEK, 2009)</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>16.375</td>
<td>17,221</td>
<td>15</td>
</tr>
<tr>
<td>Range</td>
<td>59</td>
<td>50,378</td>
<td>38</td>
</tr>
<tr>
<td>Minimum</td>
<td>2</td>
<td>834</td>
<td>4</td>
</tr>
<tr>
<td>Maximum</td>
<td>61</td>
<td>51,212</td>
<td>42</td>
</tr>
</tbody>
</table>

Table-8: Descriptive Statistics about the firms in our portfolio

It is important to note that even though the firms in our portfolio all work with open source software development, the degree of open source usage both in their product development processes and in their internal IT infrastructures vary from firm to firm. They also all serve different markets, using different approaches to appropriate returns. They are all
relatively young (mean=15) and with only Enea being the exception, they were all founded in the 1980s or later. As can be seen from Table-7, the age of the firms spans between 4 to 42 years.
8 Analysis

In the previous chapters, we have first discussed various constituents of the innovation strategy of firms, and then added the essence of open innovation to the picture and finally showed how these components showed up in an open source strategy. Some of these components were shared between innovation strategy and open source strategy, whereas some were specific to open source strategy. By doing so, we have answered the first three research questions that were listed in the purpose.

We have, however, not mentioned how these components are really used in Swedish firms that work with open source software. In this chapter, we will answer the last research question referring to this issue and explore the similarities and differences between these firms in terms of their open source strategy.

Since our research is on Swedish companies working with open source, a natural question arises about the software industry in Sweden, which we define as the “Swedish open source software scene”. Thus, we’ve added another point to our analysis which considers how the attitude of the Swedish software industry is towards open source software.

8.1 Individual Analysis of the Firms

8.1.1 Firm A

Firm A is an IT company with consultants within open source, developing IT solutions based on open source within virtualization, systems development and systems integration. The company is independent and provides never any third-party products.

The company offers turnkey solutions, expert advice and systems development, with the customer benefit mainly by focusing on skills, security, flexibility, cost efficiency and scalability, with an iterative approach. They consider themselves as specialists in integration and development of advanced IT solutions, particularly in the middleware and web-based open source solutions. The business idea of the company is helping customers to sustainably improve their operations and free up substantial financial resources, by combining the firm profile and expertise with the power of open source software, in development for more efficient IT support.

The company’s operations are characterized by openness and transparency, basing solutions and supplies to the greatest extent possible on open source, with an aim to contribute to greater understanding of how IT systems built using open source can lead to significant cost savings and synergies. Firm 1 is thus sensitive about encouraging open source development as much as possible, trying to help change how potential customers look at software development by showing how openness leads to mutual benefit, better functionality, vendor-neutral and that the customer gets greater control over their activities.

Three independent consultants started the company and with plan of a “new kind of company with a social and ethic theme” from the beginning. Since the firm had no funding initially, they kicked off their business with consultancy in object-oriented programming (OOP) and Java development. The interviewee mentioned that it was a “long struggle to reach current position”.
The company is doing open source based development, channeling their resources for own development within office automation products and also having a couple of pilot customers. Everything the company develops is 100% open source without any commercial version and the company gets its revenues from support and development. An exception to this is when the company is contracted as consultants of software developers. In this case, open source products/technologies that they work with depends entirely on the customer financing the project. OSS is here used to cut costs, but the return to the community becomes almost negligible, thus transforming the company from symbiotic to commensalistic. However, the company has recently been engaged in an OSS project which includes proprietary customizations specific to the customer, which would allow them to contribute much more to the community.

Firm A develops web-based and middleware-based object-oriented application development in various sectors and provides SMEs with office automation solutions, including hardware and software. The firm also provides Linux applications “bundled” on top of Linux, which is planned to be introduced next year.

The company also wants everything to wind up in public domain, thus all the products they develop is GPL or as close to GPL as possible (GPL-based).

The company has an expert knowledge, with at least fifteen to twenty years of experience in open source. However, they do not use any skill-set analysis to keep track of their in-house skills, and the interviewee mentioned that such analysis approaches are “a lot of unnecessary bureaucracy and administration for a company as small as ours”. The firm instead desires to develop a lot of personal talent through allocating 10% of their time in various collaborative efforts to develop the skill-set of the company.

8.1.2 Firm B

The founders of Firm B have experience from Internet and the IT-industry and the company is privately held and owned by the founders and staff.

The company defines its mission as to enable IT departments and data centers to reach better service control, visibility and quality. In order to fulfill such mission, the company offers productivity tools that enable IT-departments to efficiently deliver services in a modern production environment.

Firm B does not produce any proprietary software; instead it “uses other open source components, mix them together and develop new own software”. All product development follows a plan that is continuously updated based on input from customers, based on an innovation roadmap. The company offers unique software solutions that give service managers and IT Managers control over IT services operations.

The core product is an open source configuration management database (CMDB) aimed at small and medium sized businesses which keeps track of software and hardware assets and their relations. Therefore the company is doing application-based development that can run on various platforms.

Today, mainly GPL licensing is used in the company. In addition to GPL dual licensing including GPL and proprietary licensing is also used in the company, though rarely. The revenue streams for the company are providing service and support and licensing fees.
The firm had 10 to 14 of persons dedicated to R&D activities in product development before, but the original company has been sold to a large Swedish corporation, leaving only 2 persons from former management in a new company, which will continue OSS development. Thus they currently do not have much R&D activity going on in the company.

The company combines number of different components from different companies and develop complex systems. The area of systems’ usage is also complex. The products are not just installed and run rather they are integrated with other components.

Firm B is not interested in cooperating with universities at all, but rather “more interested to focus on things that give commercial benefits right away” rather than cooperating with academic institutions as it would “take time to give commercial results” with such collaboration, as stated by the interviewee. Furthermore, the interviewee mentioned that “in larger firms it is easy to find time to spend on cooperation with universities but smaller companies cannot spend time on it”.

8.1.3 Firm C

Firm C develops user-friendly operating system for network monitoring, log management and data analysis. Their tools are based on open source, enabling the customers to easily monitor their entire IT infrastructure and quickly get an overview of their own IT systems. The company delivers the benefits of open source and security of professional support. Their products are based on a variety of GPL licensed software.

The company develops functionality that is typically added on top of one or more open source software, usually including the code integrated functions between several of each other independent programs in this way to get the added functionality and/or usability. The source code is supplied always to the customers and there is nothing to prevent the customers from modifying it. The company, however, is not responsible for product and feature guarantees and support.

The open source projects that firm C uses in their system extends in a broad range of proven products with a reputation for alpha releases of the software as we go through to verify that the code is secure and stable. As the basis of their appliance, the company uses CentOS distribution of Linux, which can be regarded as Red Hat Enterprise Linux, the free, or CentOS write for themselves.

The company uses innovation roadmaps of 1 to 3 years and their products are rather simple, however depending on customers’ demand and need simple systems are combined together and develop complex systems such as network systems and distributed systems. The interviewee believes that increasing complexity increases availability of services.

IP management activities in the firm are mainly focused on complementary assets and co-development whereas R&D ideas come from outside to the company. As of licensing, the company mostly uses GPL and in house developed licenses while revenue streams for company come from providing consultancy and training for open source products as well as update open source products.

The open source strategy of the firm is application based with a platform based core system. The company started with already-existing software, packaging them and providing the services to maintain the software products. Now, the company also is developing open source component and systems using these components. Company also produces special products
according to customers’ needs, such as graphical user interface (GUI) and distributed embedded systems (DES).

The interviewee underlined the importance of the community, mentioning that “10 years ago no one cared about community but today 80% of the innovation in open source software project is coming from communities”.

Firm C offers services for installation, deployment, customization, integration, remote assistance, training and support. The company also offers specially adapted packages for on-site upgrades or remote upgrades and arranges cost-effective customer tailored courses at any preferred location by the customers. The support team in the company is a team of highly skilled software engineers with several years of experience in system administration and system integration with a deep understanding of the software applications. The company offers their customers the possibility to subscribe to a hardware and software maintenance service. At recurring intervals, they supply their customers with updated versions of the software and plug-in packages.

Firm C considers their partners important, believing that they add significant value to their core products, being experts on specific systems/hardware/software and represented on local markets close to the users.

8.1.4 Firm D

Firm D is an IT-consultant company in system development and quality assurance, with customers using the new technology for enterprise development and product management in a creative and innovative way. The company was founded by four entrepreneurs as a consultancy company initially, but grew constantly but slowly in the last three decades.

The company is engaged in consulting, advisory, development, out-formation and production activities in the engineering, telecommunications and chemical industry, and defense and administration, together with other related business. Activities of the company are oriented towards computer, electronics and information technology, with the company also operating leasing, renting, preferably within electronics industry. However, the company does not develop open source products but uses them in projects.

Firm D has experience in embedded systems development, with the areas of expertise within distributed systems for vehicles, trains, airplanes and various machines; high-speed communication with routers and switches; analysis instruments for medicine and life science along with engine control and power electronics.

Apart from product development, the company also offers IS/IT services in the form of programming, system architecture, quality assurance, investigations and IS/IT management, with experience in industrial development projects with central and local government and a genuine technological interest.

The company works in open source/open standards and well-known proprietary environments (especially Microsoft platform), with development of web-based business systems to one of the major state institutions, integration solutions for leading insurance companies and banks, and support systems to streamline test operations of a large, multinational, telecoms company.
Expertise of the company lies in Java, Linux and Open Source development, with focus and techniques used in architecture and systems development in open source - such as case management, communication and version control, web framework, and building and persistence management.

Firm D has a special group which has the responsibility of following recent developments in open source software. The firm holds a meeting every several months to check out the new developments in the company. Since the company does not modify open source components it does not use any specific licensing strategy.

The openness of the innovation was heavily closed on the customer for this company and the interviewee mentioned that the degree of innovation culture depended also on the customers just like the openness. Interestingly, firm 4 was found to have its own IP and applications, but that are built on top of open source components owned by its customers.

8.1.5 Firm E

Firm E delivers IT solutions based on open source-licensed software. The company works with open source software such as Joomla, Drupal, Linux, Apache, MySql and PHP and has high competence and long experience in project management, programming, graphic design and education.

The company offers complete IT solutions to its customers. The company believes that open source results in better software as bugs can be detected and corrected faster, and that no single company can match the development of software carried out and tested by thousands of people in a worldwide Internet-connected community, where being able to exchange ideas as well as codes in an instant, regardless of where you are, is the ideal soil for creative collaboration.

The company has thus stuck to open source business model because of the ability to yield higher quality, greater security and more user-friendly and easily integrated software. Moreover, such development is both faster and at less cost than in individual companies. The products that 1st Solution develops for customers is based on skills, communication and commitment, using the Internet to keep customers daily update on how their projects are running.

Firm E considers itself as a network company, working closely with other companies to provide its customers the best service available. To work with its partners to work smoothly and efficiently, the company has developed a common information platform together with other participants in the network. Coordinating the resources of the partner companies enables 1st Solution to act more flexibly, adapting quickly to market and customer demands without losing control over the products and services provided.

In order to offer its customers the best solutions available, the company has gathered widest possible expertise over time. The company has extensive practical experience in the respective areas of expertise: Project Management, Graphic Design, Web Design, Programming, Education, E-learning and IT Security.

Open source structure of the company is both application-based and platform-based. The company uses open source software as base for their projects and for customers. Many of the company’s projects are open source projects, with the revenue streams coming from product renewals, services (roughly 40%) and consultancy assignment.
Revenue streams for company are providing consultancy and services for open source products as well as open source products renewal. According to interviewee 40% of the revenues stream of the firm is providing services for open source software.

8.1.6 Firm F

Firm F defines itself as “a small company with a large network and modern approach”, running projects by virtual participants in different continents with the aid of modern systems. The company offers start-ups, small and medium sized enterprises IT environment, with readily available staff and provides a number of standard products based on open source that can be adapted for different needs. The company also provides a number of other services by agreement with customer, from the technical and administrative services to education, if necessary, everything from an IT manager to the office staff with a special understanding of entrepreneurs, new companies and new technologies.

The business idea of the company consists of the following components:

- IT as a service
- Open source software
- Existing investments in its own hardware
- Extensive experience of developing and operating in IT

The company’s business idea is based on the benefits of IT as a Service and Open Source, while compensating for the disadvantages. The company has good control over the systems of open source and is working full time to develop, adapt and learn these systems.

Firm F can thus act both as an IT department or supplement the IT department. There is close cooperation and inconceivable unless it is based on open standards, open formats and open source. The company uses and continuously tests a variety of powerful software and can also respond to specific requests outside of the application portfolio of customers.

The company provides supportive services as follows:

- Training: Custom courses in different areas.
- Specialized infrastructure: Custom development and operating environments.
- Management: Services with special focus on border management and IT.

Business Development, effective IT policy, quality, customization of software, scanning of documents and updating of records are some examples of practical services that actually can be delivered via the Internet by the company.

The company provides infrastructure for customers who in turn are involved in extensive and specialized activities. This means that the company is indirectly used by many of today’s Internet users in Sweden.

The firm provides custom development and operating environments, services with special focus on border management and IT. Both as a provider of IT services through the internet and IT services through people, Firm F considers it important to monitor new technology and understand how it may affect customers’ operations. However, it does not have any maturity or skill-set analysis. Indeed, the interviewee from believes that in open source software projects, no one knows about the software until one runs the software and sees the
features and emphasized the importance of “trial & error” approach instead of holding a detailed skill-set analysis. In many cases the company is the main responsible for open source project.

Unlike the other companies that were interviewed, the company has special interest in entrepreneurs, new technologies and new businesses, seeking entrepreneur promotion projects such as venture capital firms, incubators, mentoring and entrepreneurial events as partners in its network. The firm thinks that, by providing customized services for them can, through long experience, the existing investment and open source, creating exciting business opportunities.

Open source structure of the company is platform based and the company uses open source products without modification in their projects, with a combination of various open source applications and Linux.

8.1.7 Firm G

Although based in the capital of Sweden, the company has grown out of our original offices and expanded in a controlled fashion. The company is operating on a global level - in over 20 countries - directly or through partners.

The business idea of the firm is to provide open source solutions and services in IT-security with focus on two main areas; Public Key Infrastructure (PKI) and signing solutions. The company defines its mission as to provide state-of-the-art solutions that integrate with customer application environments.

The company is the world’s leading provider of open source enterprise Public Key Infrastructure (PKI), which is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates. As EJBCA founders and its main developers, the company offers first-hand skill and EJBCA support to the customers. The company also develops digital signing solutions.

Firm G distinguishes itself by its eminent professional consulting services, useful training courses, free community support, as well as comprehensive technical product support. Highly specialized in PKI, the company continuously develops and maintains its successful open source PKI security software.

The company is an active figure in the open source PKI community, with the concept, work and functionality of the community making it easier for visitors and members to use the company’s software, and also receiving support from the company’s professional services and enterprise support. The company is the originator, main contributor and administrator of the EJBCA project at SourceForge.net. EJBCA, short for Enterprise Java Beans Certificate Authority, is an open source, Java Enterprise Edition (JEE)-based implementation of a certificate authority (CA). In addition to the EJBCA project, the company is also handling the SignServer project at SourceForge.net.

Support, training and professional services are optional for the company. Customers may choose to have the complete planning, installation and maintenance done internally by their organization, or they may prefer to get support, maintenance and training - partly or entirely - from the company itself, or one of the company’s certified partners instead.
With half of its business generated through strategic partners, the company is actively establishing partnerships with leading resellers, technology companies, and system integrators all over the world. The firm’s partners range from global technology companies to local consultancy firms. Approximately two thirds of the partners are either business partners, who use the company’s technology and is also a supplier of their own product or technology, or technology partners, who are leading suppliers of hardware and software that complement the company’s own products. By bringing together its offerings with the suppliers, the company can create added value and present custom-made solutions. The remaining third is both business and technology partners. A few of them are also training partners. Carrying out a Partner Training Program certifies a partner to sell and deliver our courses, including certifying participants. Anyone who is a distributor, reseller or system integrator can ask to become a business partner of the company.

8.1.8 Firm H

Firm H is a global software and services company focused on solutions for communication-driven products with experience in the development of software platforms with extreme demands on high-availability and performance. The company’s expertise lies within real-time operating systems and high availability middleware.

The company has a global customer base, with the majority of customers being manufacturers of telecom equipment, while others within medical technology, industrial automation, automotive, military and avionics industries.

The company was established by four students at the Royal Institute of Technology in Stockholm, Sweden, who were designing a solution for storing data in an air traffic control system at that time. During the early years of the company, the group became pioneers of real-time programming and development of operating systems, where the foundation of the company’s art of engineering and outstanding expertise lies now.

The corporate culture of the company has more or less remained the same throughout the years, with product development being at the heart of the company ever since the start. The company’s position could be defined as a world-leading supplier of solutions for communications-intensive products. Consultant portion of the operations grew sharply and a major acquisition in Romania increased the outsourcing capacity of the company.

The company defines its core values around three “guiding stars” of focus on customers’ success, passion for the art of engineering and emphasis on team players. The company thus has enthusiasm for innovation, which is considered to be a competitive advantage for its customers. In order to utilize the global network to full extent, the company also focuses on collaboration and transfer of know-how over geographical and organizational boundaries, and between the company and its customers. The company cooperates with various partners from all over the world, including universities and global vendors.

Open source structure of the firm is platform based. The company uses open source for customer platforms for application developing, providing services and training on Linux and also providing specific optimization for Linux. The company uses open source software as a platform for applications development for customers, and also provides optimization and training on Linux and Android platform. The company also develops proprietary software based on open source. For 20 years the main core of the company has been real time
operating system. Today, the company is also planning to develop operating systems which coexist with Linux due to fast emergence of Linux as an operating system with real time support also. Another open source product of the company is proprietary add-ons for debugging in Eclipse integrated development environment (IDE). The tool is called Optima tool which is an Eclipse-based platform with the company’s proprietary software.

The company uses GPL, BSD and LGPL includes license terms for customers. According to interviewee, the company carefully uses open source software products in their projects/products and makes sure that open source products have not been misused. Also the company informs the customers of their products along with open source software which has been used.

IP management activities in the firm are mainly focused on IP rights & secrecy including copyright and few patents. The interviewee mentioned that Intellectual Property of the firm is protected with copyrights and few patents for tools. Skills and experts are part of intellectual property of the company, combination of expertise along with products. The company based on expertise and not only product.

The firm has innovation roadmaps which are continuously updated, and a specific plan for product developments and part of the processes create innovation roadmaps and innovation strategies. For example part of the innovation roadmaps of the firm is challenge to meet the customers need and considering customer segment in marketing.

8.2 Overall Analysis of the Firms

Since open source strategy is a huge topic itself, we’ve decomposed it into three headlines in order to simplify our analysis:

1. Background of each firm with open source software, considering firm history with OSS, why the firm has come up with using open source in the first place and the evolution of the software industry through the existence of the firm.

2. In-house components of the open source strategy such as the IT system of the firm, productization efforts, structure of the software and analysis tools regarding maturity of the software and skill-set of the firm.

3. Business model components of the open source strategy such as community relations, networking activities, licensing, revenue streams, open source trap management methods, existence of a dedicated open source team and whether creativity is encouraged within this team (if present)

Furthermore, since innovation and open innovation are also huge topics with a large amount of intersecting concepts, we’ve merged these two areas under one area of innovation characteristics of the firm. This contains openness of the firm’s innovation activities, presence of an innovation culture in the firm, R&D and IP management strategies in the firm, complexity of products and the organization of the firm, presence of an innovation roadmap and whether the firm has recently been engaged in any corporate venturing activities (internal or external).
In each of these areas under examination, first a summary table is to be presented in order to give a clear figure of how each area differs between the firms, followed by a detailed analysis of the firms in the sample.

### 8.2.1 Firm Background with Open Source

In this section, we are investigating background of each firm with open source software regarding motivations for using open source and also evolution of the software industry regarding open source. Firms have given various motivations for working with open source software either developing the software (themselves or using it as a base for their own proprietary software) or providing services depending on their business model. They have also mentioned differences in how the software industry has evolved from the time they were founded and now.

The interviewee from firm A claimed that the industry was enriched with lots of application that are not benefiting from the unique business models introduced by open source software with the industry changing but rather very slowly. The position of OSS in the software industry, in the interviewee’s own words, changed from “extremely bad” to “not so extremely bad” in front of commercial opponents. The interviewee also discussed that so much money was wasted on proprietary software which could be bestowed more efficiently by open source, remarking that a point would be reached in the software industry when a breakthrough would come for open source. Furthermore, the respondent said it was very difficult for a small company to support sales and marketing efforts and to have certain revenue to manage the company and that “one cannot just sit and develop the software and wait for customers”. The only change that the firm faced was to be patient and accept the gradual change in the industry.

It had always been cumbersome to sell proprietary software for firm B and customers were skeptical about buying software from a small, unknown company according to the interviewee while using open source software would allow the firms to reach new customers. When the company launched open source software, it overcame those obstacles, according to the respondent. The company has developed open source components since then and has checked constantly the market to see whether it is ready for these products or not.

The interviewee from firm C claimed open source software to be a good way of attracting customers which costs a lot of money. In addition to this, the firm could develop open source software and release it to get feedback from community, which is a very good way to make sure that the software is working while the firm doesn’t need to go back and re-design it. In other words, getting feedbacks from community allows the firms to early testing the software. At beginning making open source software was not easy according to the respondent; not many companies agreed to use open source software but today more projects are based on open source software.

Firm D only uses open source components in projects and does not develop open source products. Sale of open source software systems based on UNIX started in mid 80’s was overtaken by Linux as the main platform for projects and as the internal IT system by 1997. According to the interviewee, open source software started growing in the mid 90’s as applications were developed in UNIX and then the company easily moved to Linux. In the respondent’s opinion, open source software industry is growing in several areas today such as mobile phones, most of which are based on open source software. Thus, the open
source software is very successful except for desktop (PC dominated by commercial). An interesting note was that the interviewee mentioned that even the Macintosh was based on the BSD, which is a derivative of the UNIX operating system and is thus open source software.

According to the interviewee from firm E, open source software products are superior to their commercial counterparts with many developers working on it and open source software allows doing much more with the budget that firm has compared with licensing proprietary software. Furthermore, being part of open source software community contributes to modules that can be reused by others. The interviewee also mentioned that open source software had started with Linux while nowadays it is more “acceptable”.

The company uses open source software modules and uses OSS as a base for proprietary solutions aimed at customers. Most of the projects of the company are open source software projects. In the interviewee’s opinion, open source software was not popular in the beginning like Microsoft and people had doubts about why they should use open source software and how to get profits from open source products and how to trust open source software. But things has changed in time and especially in the last three to four years open source software has started to become more popular as more people are using open source software nowadays.

The interviewee from firm F argued that an increasingly number of firms had started using open source software as basis for their commercial products in order to develop knowledge, share knowledge and have access to flexible knowledge. When the company was founded, open source software didn’t exist and the industry was dominated by commercial companies like Google and Apple. The firm was not involved in innovation and using open source software. The respondent mentioned that tremendous things have happened lately and that today, open source is the major factor and companies are using open source software for commercial products also.

The respondent from firm G mentioned that open source industry had changed dramatically. At beginning it was very difficult to have business based on open source but today’s even government started using open source. Furthermore, development time become has extremely shorter according to the interviewee. Finally, according to the respondent from firm H, Linux and other open source software are increasing in usage both in Sweden and internationally while commercial and legal aspects are becoming more and more important for companies.

8.2.2 In-house components of the open source strategy

<table>
<thead>
<tr>
<th>Firm</th>
<th>Skill Level</th>
<th>Maturity Analysis</th>
<th>Skill-Set Analysis</th>
<th>Taxonomy of IT System</th>
<th>Importance of IT System</th>
<th>OSS Structure</th>
<th>Productization</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Expert</td>
<td>No</td>
<td>No</td>
<td>Stable</td>
<td>Operational</td>
<td>Application-based</td>
<td>Yes, but seldom</td>
</tr>
<tr>
<td>B</td>
<td>Expert</td>
<td>To some extent</td>
<td>No</td>
<td>Dynamic</td>
<td>Mission-critical</td>
<td>Application-based</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table-9: Comparison of the in-house components of the open source strategy for our sample of firms
Table-9 (continued): Comparison of the in-house components of the open source strategy for our sample of firms

<table>
<thead>
<tr>
<th>Firm</th>
<th>Skill Level</th>
<th>Maturity Analysis</th>
<th>Skill-Set Analysis</th>
<th>Taxonomy of IT System</th>
<th>Importance of IT System</th>
<th>OSS Structure</th>
<th>Productization</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Expert</td>
<td>Yes</td>
<td>No</td>
<td>Variable flexibility depending on development</td>
<td>Partially Operational</td>
<td>Application-based with a platform-based core system</td>
<td>Yes</td>
</tr>
<tr>
<td>D</td>
<td>Expert</td>
<td>To some extent</td>
<td>No</td>
<td>Quite Stable</td>
<td>Mission-critical</td>
<td>Application-based</td>
<td>No</td>
</tr>
<tr>
<td>E</td>
<td>Expert</td>
<td>To some extent</td>
<td>Yes</td>
<td>Flexible (In progress)</td>
<td>Operational (In progress)</td>
<td>Application-based &amp; Platform-based</td>
<td>Yes</td>
</tr>
<tr>
<td>F</td>
<td>Intermediate</td>
<td>No</td>
<td>No (Trial &amp; Error)</td>
<td>Extremely Stable</td>
<td>Mission-critical</td>
<td>Platform-based</td>
<td>No</td>
</tr>
<tr>
<td>G</td>
<td>Expert</td>
<td>To some extent</td>
<td>Yes</td>
<td>Very Stable</td>
<td>Operational</td>
<td>Application-based &amp; Platform-based</td>
<td>Yes</td>
</tr>
<tr>
<td>H</td>
<td>Expert</td>
<td>Yes</td>
<td>Yes</td>
<td>Fairly Stable</td>
<td>Partial Mission-critical</td>
<td>Platform-based</td>
<td>Yes</td>
</tr>
</tbody>
</table>

As can be observed from the table above, all but firms but one (7/8=87.5%) classified their IT skill-set to be at expert level, with the only exception being firm F, which considered itself as having intermediate level skills in open source. This is particularly interesting since firm F was established in 1997 (13 years of existence in open source) whereas firm A, for example, classifies itself as being experts despite only 4 years of existence in the industry.

After learning about their skill-set level in open source, interviewees were later asked if they kept track of their skill-sets through a certain “skill-set analysis”. Almost all respondents gave a negative response, with only two firms confirming that they have a mechanism of skill-set tracking.

Unlike skill-set analysis, almost all of the firms were found to have maturity analysis within the firm (6/8=75%), although most of them (4/6=67%) said that they did not have a specialized analysis tool or mechanism for this purpose, but rather had a primitive approach which is summarized as a maturity analysis “to some extent”. For example, firm C asks the developers to define the projects and how long it will take to make the project instead of having such specialized group.

Considering the IT system infrastructure, most of the firms (5/8=62.5%) classified their infrastructure as stable, although with varying degrees of stability among the firms. Only one firm made the classification as dynamic, whereas two mentioned that they had a flexible infrastructure. The interview from firm C, in particular, commented that the flexibility of the system was variable, depending on the development. Regarding the importance of the system, the results were divided. Four firms (4/8=50%) told that their IT system was mission-critical while the remaining four firms mentioned that it was operational. Besides, one firm from each of these two groups described their infrastructure as “partly mission-critical” and “partly operational”. The interviewee from firm E emphasized that when company is working with open source product, IT system of the company could not be mission critical and
stable or dynamic but rather it should be operational and flexible and that their IT system still was still in progress to implement the flexibility and operationality.

As in the case with maturity analysis, six of the eight firms interviewed (6/8=75%) confirmed that they were allocating their time to productization efforts for the software that they develop or work with in addition to just making it work at the end of development. As for development, most of the firms were found to develop application-based open source software, that is software that can be run independently of the platform (for example a content management system can be run either on the proprietary Microsoft/Apple/IBM platform or on an open source platform such as UNIX/Linux/BSD just like itself. Two respondents in particular mentioned that they were developing both application and platform-based software solutions in their firms for their customers. Furthermore, one firm mentioned that although their products were application-based, the core system was platform-based, running on the UNIX platform.

### 8.2.3 Business model components of the open source strategy

Following in the footsteps of Dahlander and Magnusson (2005), we have classified the relationship of the firms with the community. As can be seen from the table above, half of the firms pursue a symbiotic relationship with the respective open source community, whereas the other half is found to pursue a commensalistic relationship. None of the firms in our study were found to have a parasitic approach. Firms can take the community’s support during development of the open source software, which is mentioned by the interviewee from firm D that “being part of community gives opportunity to firm to fix the problems by asking help from community”, even though the firm doesn’t develop any open source software itself.

#### Table-10: Comparison of the business model components of the open source strategy for our sample of firms

<table>
<thead>
<tr>
<th>Firm</th>
<th>Relationship with the Community</th>
<th>Networking Participation</th>
<th>Licensing Strategy</th>
<th>Revenue Stream(s)</th>
<th>Open Source Trap Management</th>
<th>Open Source Development Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Symbiotic (Gradually improving)</td>
<td>Personal Networks</td>
<td>GPL and GPL-derived</td>
<td>Service, Consultancy &amp; Support</td>
<td>None</td>
<td>No</td>
</tr>
<tr>
<td>B</td>
<td>Commensalistic</td>
<td>Service firms, Competitors</td>
<td>GPL and Dual Licensing</td>
<td>Service, Support &amp; Licensing Fees</td>
<td>Yes (Licensing)</td>
<td>Yes</td>
</tr>
<tr>
<td>C</td>
<td>Symbiotic</td>
<td>Sales partners, Resellers, Consultancy firms (Global)</td>
<td>Mostly GPL, but also in-house</td>
<td>Updates, Consultancy &amp; Training</td>
<td>Yes (Community-driven)</td>
<td>Yes</td>
</tr>
<tr>
<td>D</td>
<td>Commensalistic</td>
<td>Customers, Vendors, Universities</td>
<td>None</td>
<td>Usage &amp; Support</td>
<td>Yes (Licensing)</td>
<td>Yes</td>
</tr>
<tr>
<td>E</td>
<td>Symbiotic</td>
<td>Customers, Universities, Personal Networks</td>
<td>None</td>
<td>Product Renewal, Service &amp; Consultancy</td>
<td>Yes (Integration)</td>
<td>No</td>
</tr>
</tbody>
</table>
Table-10 (continued): Comparison of the business model components of the open source strategy for our sample of firms

<table>
<thead>
<tr>
<th>Firm</th>
<th>Relationship with the Community</th>
<th>Networking Participation</th>
<th>Licensing Strategy</th>
<th>Revenue Stream(s)</th>
<th>Open Source Trap Management</th>
<th>Open Source Development Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Commensalistic</td>
<td>Suppliers &amp; competitors (Entrepreneurial)</td>
<td>None</td>
<td>Usage, Service, Support &amp; Training</td>
<td>None</td>
<td>No (External via contracts)</td>
</tr>
<tr>
<td>G</td>
<td>Symbiotic</td>
<td>Consultancy firms, Resellers, Vendors</td>
<td>LGPL</td>
<td>Consultancy, Support &amp; Training</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>H</td>
<td>Commensalistic</td>
<td>Universities, Vendors (Global)</td>
<td>BSD, GPL, LGPL (Terms for the customer)</td>
<td>Service, Consultancy &amp; Training</td>
<td>Yes (Licensing)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

All of the firms were found to be active in networking, having various types of collaborators for benefiting from open innovation. Typical collaborators include competitors, consultancy/sales/service firms, vendors, universities, resellers and competitors, which matches with the literature. Universities and competitors were not found to be very popular, though. Another interesting note is that two firms (2/6=33%) underlined the importance of personal networks rather than firm-level networks.

Regarding revenue streams, most of the firms (5/8=67.5%) were found to generate their revenues from consultancy. Other popular streams include training, support and service, independent of whether the firm develops the software itself or not.

Open source software brings unique risks in addition to the unique benefits when a firm decides to work with it. The two most popular risks are integration problems, either with commercial or other OSS components, and licensing. When we look at the table above, we see that licensing indeed is considered to be a major trap for firms and that about one-third of the firms do have an attitude for solving “licensing” traps.

One of the firms works collaboratively with the community to solve this problem, while another thinks integration issues are more important, whereas the other third of the firms do not consider pursuing trap management at all for such problems.

Our final point in this section was whether firms have a dedicated open source development team and if they had, whether they were encouraging creativity within this team. Based on the interviews we found that two firms do not have any specialized team at all for developing OSS, owing to the small size of the firm, and a third firm acquiring external developers through contracts.

8.2.4 Innovation characteristics of the open source strategy

Since the roots of open source software are embodied in open innovation, it is obvious that open source firms should have a heavily open innovation strategy. As expected, most of the firms (5/8=62.5%) characterized their innovation activities as “heavily open”, with only firm 5 using an alternate classification of “mostly open”. To our surprise, a considerable amount of the firms (3/8=37.5%) interviewed classified their innovation strategies as “heav-
ily closed”, with one of these firms mentioning that the closeness of the innovation depending on the customer’s needs.

All interviewees except one confirmed that they had innovation culture within the firm, with the interviewee from firm C mentioning that presence of innovation culture makes it easy to bring new ideas to company. The interviewee from firm B informed that listening to customers to get new ideas helps them to solve problems better. Firm H was found to be also heavily involved in innovation with one of its core products getting reward in 2009 and 2010 respectively, for its innovative architecture.

Regarding R&D management, half of the interviewees admitted that their firm had no R&D activities. Two of the firms are planning to engage in R&D management in the near future; whereas the remaining three notified that they had a specific R&D management strategy.

All of the firms were found to have a specific IP management strategy, with most of the firms focusing on IP rights in terms of copyright and patents. Three firms were found to focus on trade secrecy as well. An interesting observation was that one-third of the firms emphasized the importance of complementary assets. Similarly, the respondent from firm H remarked that skills and experts are a vital part of the firm’s IP portfolio of the company, in combination of expertise along with the firm’s products. The firm’s IP is based on expertise, rather than just the products. Furthermore, the interviewee from firm A mentioned that the unique profile of the company and the virtual network of developers that the firm has is the most important intellectual property that the firm possesses, without mentioning trade secrecy at all.

When we look at the complexity of the products, we see that majority of the firms (6/8=75%) classify their products as complex, although the degree of complexity changing from firm to firm, or as specified by the interviewees, from “fairly complex” to “rather complex”. Firm B, for example, combines a number of different components from different companies and develops complex systems, with the area of systems’ usage also being complex. The products are “not just installed and run, rather they are integrated with other components” according to the interviewee. Only firm C has classified its products as being “rather simple”, but “depending on customers’ demand and need, simple systems could be combined together and complex systems such as network systems and distributed systems could also be developed”, according to the interviewee and the interviewee believes that increasing complexity increases availability of services. Of course, when the complexity of the products is high, integration problems arise, as mentioned with trap management. Firm D, for example, does a lot of integration between different systems due to the high complexity of its products.

Table-11: Comparison of the innovation characteristics of the open source strategy for our sample of firms

<table>
<thead>
<tr>
<th>Firm</th>
<th>Openness</th>
<th>Innovation Culture</th>
<th>R&amp;D Management</th>
<th>IP Management</th>
<th>Complexity of Products / Organization</th>
<th>Innovation Roadmap</th>
<th>Corporate Venturing</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Heavily open</td>
<td>Efforts in implementing</td>
<td>None, but planned</td>
<td>Complementary assets (unique profile &amp; virtual networks)</td>
<td>Quite complex / Flat</td>
<td>Yes, but not functional</td>
<td>None, but external venturing planned</td>
</tr>
</tbody>
</table>
Table-11 (continued): Comparison of the innovation characteristics of the open source strategy for our sample of firms

<table>
<thead>
<tr>
<th></th>
<th>Heavily open</th>
<th>Yes</th>
<th>None, but planned</th>
<th>Complementary assets &amp; Focus on IP rights</th>
<th>Complex/Flat</th>
<th>Yes</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Heavily open</td>
<td>Yes</td>
<td>To some extent</td>
<td>Complementary assets (Co-development)</td>
<td>Complex/Flat</td>
<td>Yes</td>
<td>Internal</td>
</tr>
<tr>
<td>C</td>
<td>Heavily open</td>
<td>Yes</td>
<td>None</td>
<td>Customer-specific Secrecy</td>
<td>Complex / Flat</td>
<td>No</td>
<td>Internal &amp; External</td>
</tr>
<tr>
<td>D</td>
<td>Mostly open</td>
<td>Yes</td>
<td>Customer-based</td>
<td>Focus on IP rights</td>
<td>Complexity depending on products / Flat</td>
<td>No</td>
<td>Internal</td>
</tr>
<tr>
<td>E</td>
<td>Mostly open</td>
<td>Yes</td>
<td>None</td>
<td>Secrecy</td>
<td>Rather simple / Flat</td>
<td>No</td>
<td>None (Entrepreneurship)</td>
</tr>
<tr>
<td>F</td>
<td>Heavily closed</td>
<td>Yes</td>
<td>None</td>
<td>Secrecy</td>
<td>Complex / Flat</td>
<td>Yes</td>
<td>Internal</td>
</tr>
<tr>
<td>G</td>
<td>Heavily open</td>
<td>Yes</td>
<td>To some extent</td>
<td>Focus on IP rights (Copyright)</td>
<td>Complex / Flat</td>
<td>Yes</td>
<td>Internal</td>
</tr>
<tr>
<td>H</td>
<td>Heavily closed</td>
<td>Yes</td>
<td>Decentralized global network</td>
<td>Focus on IP rights &amp; Secrecy (Copyright and few patents)</td>
<td>Fairly Complex / Complex</td>
<td>Yes</td>
<td>External</td>
</tr>
</tbody>
</table>

The interviewee from firm C remarked that they were developing simple systems, yet the complexity was depending on the customers, whereas the interviewee from firm F simply mentioned that they were developing “rather simple” products.

Regarding complexity of the firm organization, on the other hand, we see that all firms except one are simple, flat organizations. Only one firm has a complex organization coming from a decentralized global network.

Most of the firms were observed to have a roadmap for planning their innovations in the near future. Firm C has “1 to 3 years roadmaps”, for example. Firm G has innovation roadmaps for development of next generation products and a lot of R&D activities are going on for this purpose inside the firm. Firm H has innovation roadmaps which are continuously updated, with a plan for product developments and part of the processes creating innovation roadmaps and innovation strategies. For example, “part of the innovation roadmaps of the firm is challenge to meet the customers need and considering customer segment in marketing”, according to the interviewee. Firm A also has an innovation roadmap, but it is not functional at the moment. For firm B, all development follows a plan that is continuously updated based on input from customers.

Finally, all firms except one were observed to have been engaged in corporate venturing (internal or external)

8.2.5 Open source development scene in Sweden

The interviewee from firm A mentioned that people on the political platform that haven’t realized what IT really is about, thinking that IT is just a tool so it has to work. Even one
former prime minister of Sweden believed that there is no need to have an IT minister but what they missed according to the interviewee is that IT is indeed a catalyst for business, collaboration and things to start; it is one of the strongest driving forces for business. Furthermore, IT is much more about democracy, policy, globalization, collaboration; therefore one should see what is going on and what is needed.

The respondent believes that the politicians failed to capture the soul of IT and still see the IT from commercial eyes. It is particularly important here because if one wants growth in political establishment, then the opportunity to pay attention to IT should not be missed. The respondent gave an open source seminar that was held in Sweden once as an example, where only two politicians were present. In the respondent’s opinion, government is doing a lot in tax reduction in other sectors but is failing to encourage the IT industry likewise. Thus, people who are involved in IT sector should provide more possibilities to discuss the situation but unfortunately they don’t do that.

The interviewee from firm B discussed that technology for open source software is very technical and thus is sold mostly to IT departments. According to the respondent, open source software is mature in Sweden and strong proprietary vendors are pushing for their commercial solutions in order not to lose their lion’s share to open source counterparts.

The interviewee from firm C remarked that many companies prefer to use commercial products rather than open source products because whenever they face a problem they can immediately solve it. Besides, to make business out of open source software is not really hard at all; companies can make money from source code access and more code analysis.

In the opinion of the interviewee from firm D, open source has not been very successful in Sweden and not many firms are selling and modifying open source products efficiently. In fact, there is no much money in licenses according to the respondent. The interviewee from firm E, on the other hand, thinks that government should not be directly involved in helping with open source, but should rather provide good education only and help the companies to recruit people with qualified skills.

The respondent from firm F argued that large companies want commercial products like Microsoft and it is hard to use open source software, but now it is starting to move to open source products. On the other hand, the respondent from firm G briefly mentioned that open source organizations still have a lot to be improved.

Finally, the interviewee from firm H remarked that Linux and other open source software products are increasing in usage both in the international level and in Sweden with many companies using open source software in their products nowadays.

8.3 Summary

In this chapter we analyzed the qualitative data collected during our research and also tried to answer our last research question in the purpose of this thesis. We showed how we managed to combine the theoretical findings and empirical perspectives related to our study and analyzed how various elements that we had found in theoretical studies vary in Swedish firms that work with open source software. By analyzing the firms both individually and comparatively, we aimed to obtain a more objective picture of the subject, showing how the elements that were explored in the theoretical framework exist in reality. The structure of the chapter had the following format: first we discussed the findings related to each indi-
individual firm and then compared the results of the individuals analyzing in order to dedicate similarities and differences and have general analysis of the firms. At the end of this section we answered a further question that arose about the software industry in Sweden, which we defined as the “Swedish open source software scene” to get a general picture of how the attitude of the Swedish software industry is towards open source software.
9 Conclusions, Reflections and Future Research

9.1 Conclusions

The purpose of this thesis was to investigate the open source strategies in Sweden by examining a number of Swedish firms that work with open source software. In order to get better insights into open source strategies, we tried to build a logical chain of reasoning so that, four research questions were formulated.

The first question was about identifying the different types of components of innovation strategy within firms. After conducting a thorough and extensive literature review we have found that the major constituents of the innovation strategies used by Swedish firms are categorized into three main groups: internal - including business model, IP management, R&D management and innovation roadmap; external - including corporate venturing, communities and networking, and firm specific - including innovation processes, innovation culture, firm size and complexity.

The second question was about understanding how these components reincarnate in open source strategy. In doing so, we moved from innovation to open innovation and through open innovation we ended up in open source software and tried to address these elements, not from a pure innovation point of view, but rather how they look in terms of open source strategy and analyzed each component in detail regarding open source strategy. In order to simplify our analysis, we have grouped these components under three headlines:

- Background of each firm with open source software: firm history with OSS, why the firm has come up with using open source in the first place and the evolution of the software industry through the existence of the firm.

- In-house components of the open source strategy: the IT system of the firm, productization efforts, structure of the software and analysis tools regarding maturity of the software and skill-set of the firm.

- Business model components of the open source strategy: community relations, networking activities, licensing, revenue streams, open source trap management methods, existence of a dedicated open source team and whether creativity is encouraged within this team (if present).

Furthermore, since innovation and open innovation are also huge topics with a large amount of intersecting concepts, we’ve merged these two areas under one area of innovation characteristics of the firm. This contains openness of the firm’s innovation activities, presence of an innovation culture in the firm, R&D and IP management strategies in the firm, complexity of products and the organization of the firm, presence of an innovation roadmap and whether the firm has recently been engaged in any corporate venturing activities (internal or external).

We have seen that various components which are classified as being internal, external and firm-specific can also be observed in open source strategy for firms that work with open source software. We have found that components such as R&D management, IP management, networking and business model from the innovation strategy can also be seen in
open source strategy. Depending on whether the firm focuses on product development or services/deployment, licensing or the business model comes into consideration for determining the open source strategy.

The third step of our research was to identify the unique components of open source strategy in the Swedish firms which work with open source software. According to theoretical findings, licensing, open source business model, skill set analysis, maturity analysis, productization and integration (platform-based or application-based) have been considered to be the most essential when it comes to open source activities within firms.

The last research question unveiled how all these components of open source strategy vary among the firms that work with open source software in Sweden. The same question also revealed the similarities/differences between open source strategies of Swedish firms that work with open source software in terms of their own open source strategy components. Most of the components have been found to be fundamental pieces of the open source strategies of firms in reality. Innovation culture, innovation roadmap, networking, IP management, maturity analysis, corporate venturing, licensing, application-based integration, presence of a dedicated open source development team, relationship with the community, productization, high degree of product complexity and openness, stable IT infrastructure, whereas a low degree of organizational complexity and small firm size were all found to be a part of the open source strategy of the Swedish companies that were interviewed.

On the other hand, some of the components suggested by the literature were not considered essential by the firms while forming their open source strategy. R&D management and skill-set analysis are such components. Furthermore, firms were found to be split in half in terms of the importance of their IT infrastructure. An interesting observation was the emergence of personal networks instead of firm-level networks such as universities, competitors or customers. We have also found that the skill-set of the firm was actually not determined by the duration of existence and market experience of the firm, but rather by the individual experience of the employees in the IT department.

The results of this research have shown which components are critical in innovation strategies and how they are linked to the open source strategy. We have identified the essential constituents of open source strategies of Swedish firms and how they are implemented by firms. The findings of this research can be applicable in small or medium sized firms inside the Sweden which focus on open source software as part of their products or pure open source products.

Our study has revealed that open source strategy shares most of its components with innovation strategy (with some components found to be specific to open source strategy only); thus confirming the link between open source software and innovation. Therefore, firms that work with open source software have to pay attention to innovation and form an appropriate innovation and open source strategy if they want to be successful in the software industry.

9.2 Reflections

The software industry has evolved at a rapid pace to become one of the largest industries in the world economy, attracting a mass-market of consumers for business as well as recreational software (Dahlander, 2004) so research in this area is a challenging task. In our study
we mainly focused on open source strategies used by Swedish firms which work with open source software. In order to obtain an objective picture and provide a better understanding on open source strategies first we defined the main components of open source strategy and then we investigated how these components were actually implemented and used by these firms.

Since we have chosen to investigate the open source strategy of Swedish firms in a systematic fashion, the qualitative approach was selected to be the most appropriate. Furthermore, we have to say that our research findings have to be perceived as situational. The identified components of innovation strategy and open source strategy do differ significantly across organizations; therefore some findings may not be applicable or have a completely different effect in some organizations. However, many of these findings might be useful for better understanding and getting insights to open source strategies.

This research has mainly focused on qualitative interviews and extensive literature review as primary resources of information. In addition to this, less attention was given to statistical and quantitative data analyses in this research. We have to mention that if a quantitative approach had been used as well, different insights and conclusions could have been obtained with a higher level of detail. Using such approach is suggested as future work instead.

Overall, we believe that this research has achieved its goals and findings can constitute a valuable source of information, especially for firms which work with open source software.

9.3 Suggestions for Further Research

Because of limited time and the relatively low response rate for interviews, we have been able to investigate open source strategy of just eight Swedish firms out of twenty-five. Considering that entrepreneurship rate in Sweden is quite high, the number of firms working with open source is likely to increase sharply in the future, thus yielding much more companies to investigate. As a result, a potential further research is investigation of open source strategy in a national level in Sweden, covering all or almost all of the Swedish firms that work with open source software.

Another potential further research could be investigation of the effects of the open source strategy on economic returns of firms. To do so would require a detailed regression analysis; this could be done by the help of data coming directly from Skatteverket, the Swedish Tax Board. Since we had limited time, we were able to only use secondary data from Äffarsdata and ignored such economic effects of the open source strategy.
10 References


References


11 Appendix: Interview Structure

Part I. Introduction

1. Firm background
   a. History (with open source)
   b. Size (Total number of employees)
      (Operations)
   c. Overall organizational structure (De-centralized or centralized distribution, Dis-
      tribution of headquarters, size and operational function of headquarters and or-
      ganizational structure in each headquarter)
2. Role of the interviewee in the firm (CEO, CTO, IT or R&D department, development
   team etc.)
3. How did you come up with open source software development in your firm (Motiva-
   tions for ending up with open source software development)?

Part II. Open Source Strategy

4. What changed in the software industry between the time the firm first started and now?
5. Do you develop only open source products or open source products along with com-
   mercial ones (could be commercial version of the open source product, or a completely
   different commercial product based on various open source components)
6. Is your OSS strategy platform-based or application-based?
7. What is (are) the revenue stream(s) for the firm? (Distribution, marketing, branding,
   training, consultancy, support, customized development etc.)
8. Can you briefly talk about the licensing strategy in your firm?
9. How would you define the level of firm’s expertise /in-house skill-set in open source
   software (OSS)?
10. Do you perform a “skill-set analysis” within the firm, continuously or just before choos-
    ing the project?
11. Are you aware of the unique risks of OSS? Do you have a special mechanism, e.g. a go-
    vernance structure to manage the OSS development and avoid potential “open source
    traps” that may result in chaos in the firm? If not, how do you manage these risks?
12. How would you describe the taxonomy of your IT system? (Stable, Flexible or Dynam-
    ic)
13. How would you describe the importance of the IT system to your firm? (Experimental,
    Low Priority, Operational or Mission Critical)
14. Do you spend considerable amount of time before deciding to work with an OSS
    project in order to perform any “maturity analysis” of the open source project that you
    are planning to work with?
15. How would you describe your relationship with the open source community and participation in open source software projects? (What is the level of firm’s involvement in open source projects?)

16. Do you collaborate with any partners via networking? What type of partners do you collaborate with in networks (customers, competitors, suppliers, consultants, engineering staff, industrial associations, government and private laboratories, universities etc.)?

17. Do you have a special team responsible for the open-source efforts of the firm? If you do, do you encourage freedom and creativity for them? (e.g. let your developers be also the “customers” of the open source infrastructure)

18. Do you carry out productization of the OSS? (Productization means making software work for the general case and making it as easy as possible to use; it requires a huge amount of work and can take double or triple the amount of work it took to complete the original features and turn a program into a product.)

19. Could you briefly describe open source development in Sweden? Do you think it is getting enough support (from companies and government etc.)?

Part III. Innovation Activities

20. How would you define your core capabilities (core competencies)?

21. What is the proportion of open and closed innovation activities inside the firm? How does the firm keep the balance between these activities? (“Openness” of the open innovation)

22. Do you think your firm possesses a specific innovation culture? If not, are you planning to implement it in the future?

23. How would you define the complexity of your products and complexity of your organizational structure?

24. Can you briefly talk about your
   a. R&D Management (if there is closed innovation going on in-house)?
   b. IP Management (Complementary assets, Secrecy or first mover advantages)?

25. Do you use any specific innovation roadmap for planning innovation activities in the firm?

26. Has the firm been engaged in any venturing activity recently/ in the past (internal e.g. spinoffs, spinouts or external e.g. acquisitions)? If not, are you thinking to do so in the future?

27. Do you have any objections to referring to your firm explicitly by its name in our analysis in the thesis?