



OULUN YLIOPISTO  
UNIVERSITY of OULU

# Business models and motivation of companies that take part in open source software development projects

University of Oulu  
Department of Information Processing  
Science  
Master's thesis  
Pyy Korhonen  
10.9.2013

## Abstract

The target for this study is to find out what kind of business models and motivation profit oriented firms have for partaking in open source software development projects. I became interested in this topic when I worked in an open source software development project led by a company.

The research was conducted by going through relevant prior literature and by assessing the obtained results from case studies conducted on actual companies. The case studies are conducted based on a framework built on top of prior literature.

The main observations are that there exist many viable methods for obtaining value and revenues from open source software. The most flexible of open source business models is the service provider model. Firms also have many motivations for partaking in open source software development projects but financial aspects encompass the most important motivation. Open source communities and the developers in them can be great tools for profiting from open source software development.

The thesis contributes to the research field by providing an in-depth study on open source business models and firm motivation.

*Keywords:*

Open source, business models, business motivation

## Foreword

First I would like to thank my thesis instructor, Lecturer Juhani Warsta, for guiding me through the pitfalls of writing a thesis.

I would also like to thank my opponent, Docent Veikko Seppänen, for diligently reading through my thesis, informing about wording mistakes and suggesting improvements to word forms.

I want to thank my wonderful wife, Henna, for providing criticism and speaking for the importance of good planning. You are a difficult one to live with as you have such strong opinions on what is right. However, I appreciate those opinions and love you just the way you are.

I want to thank my parents, Taina and Veikko, for their encouragement and support. I'm relieved that I got the thesis finished before summer so I will be able to come and enjoy barbequing with you guys.

I want to thank my little sister, Sini, for being a relatively easy teenager to deal with. Your skills as an artist are almost at my level. You will never reach my musical skills, however. Desu.

I want to thank my little brother, Visa, for the great cooking he does and for his extreme critique of modern computer games and of Finnish beer culture.

I want to thank all my friends who have inquired me about how my thesis work is going and when it is going to be finished. It is fun meeting you guys every week for our weekly gaming night. Trust the die.

I can't thank all the people here who would deserve my thanks and appreciation but do not think I've forgotten you.

Pyry Korhonen

Oulu, August 11, 2013

# Contents

Abstract.....	2
Foreword.....	3
Contents .....	4
1. Introduction.....	5
1.1 Why I chose this research subject.....	5
1.2 Research question.....	6
1.3 Research methods .....	6
1.4 Thesis structure and expected results .....	7
2. Background .....	8
2.1 Open source software and licences .....	8
2.2 Firms and open source software development projects .....	9
2.3 Firm motivation for open source software development projects .....	11
2.4 Open innovation .....	14
2.5 Open source developers .....	15
2.6 Motivating open source developers .....	17
2.7 Connections and control.....	18
2.8 A summary of the background literature .....	19
3. Business Models .....	22
3.1 Firms and open source business models .....	22
3.2 Frameworks for analyzing software business models .....	26
4. Analysis .....	30
4.1 Research data.....	30
4.2 The case companies .....	31
4.3 Analyzing the case companies .....	40
5. Discussion .....	47
5.1 The analysis framework .....	47
5.2 Business logic and open innovation.....	50
5.3 The case companies and open source business models.....	51
5.4 Motivations.....	54
6. Conclusions .....	57
References.....	59
Appendix A. Interview Themes .....	61
Appendix B. The Case Companies .....	63
Appendix C. Interviews .....	65

# 1. Introduction

Open source software development has gained a momentum in the recent years and can now be considered a serious development method for commercial companies, as pointed out by Ågerfalk and Fitzgerald (2008). Just a few years earlier, in 2006, West and Gallagher wrote that it is very hard to understand what motivates firms and organizations to take part in open source software development as you essentially give away the end product for free. Giving away for free can be understood in the case of individuals who develop software as a hobby, but for a firm that would normally mean that the resources spent on the development process have been completely wasted. It is well known that firms and organizations are always looking for new ways to be more innovative, competitive and efficient and open source software development promises quite a few advantages on this field including reduced salary costs, reduced cycle time from follow-the-sun development, cross-site modularization of development work, a larger developer pool, a greater chance of new ideas, shared best practice and closer proximity to customers. (Ågerfalk and Fitzgerald, 2008).

The popularity of open source software development can also be seen in how more and more developers flock under its flag and this in turn has led to attention from academic researchers as described by Gary et al. (2009). These researchers try to understand how open source developers, firms and organizations act together and what their impact on the world at whole is. Gary et al. (2009) have also pointed out that the proliferation of new business models utilizing open source software development means that firms have yet to converge on and to understand the best ways and strategies to use this software development method.

## 1.1 Why I chose this research subject

Open source software development has grown increasingly popular in the recent years as described by Hars and Ou (2001) and Gary et al. (2009). As more companies try to utilize it as a business model new strategies for such utilization are conceived of and tested. I got a chance to work in such an open source software development project sponsored by a firm last year and that sparked my interest in trying to understand what could motivate firms to do it. If the firm does not get any value from the actual software under development then where is the value? It was not at all clear for me where the value for the firm was generated in the project that I was involved in but I had quite some hunches. Maybe they will sell something complimentary to the open source product or related services? Who has decided on this development and business model and what motivation is behind this decision? Why is this open source software development method better here compared to the traditional one of doing the work inside the firm itself?

Behlendorf has stated in his article in 1999 that firms, organizations and teams need to have extremely good evaluation and analysis skills and tools in order to make sure that open source software development really is a worthwhile pursuit. Researching this sounded really interesting and could benefit myself and those firms that are already

using or planning to use open source software development and for this reason I chose this subject.

## 1.2 Research question

In this thesis I am interested in companies that sell or develop open source products or offer complementary services and products. Companies that use open source components as part of their products or services are also of interest. It is not necessary that the companies use open source development tools to do business.

I have one main research question in this thesis:

What are the motivation and business models behind firms investing resources in open source software development projects?

Because of the nature of the open source development method firms have multitudes of mechanisms geared towards attaining business value from said development as described by West and Gallagher (2006) and Lerner and Tirole (2002). I'm interested in finding out how that value can be obtained and what motivations firms have for doing so. Also as previously mentioned, Gary et al. (2009) have discussed that this development field has not yet fully matured and new models for attaining business value are still developed. Therefore, I considered it interest to do an up to date study into this matter.

It is also my purpose to build on the findings of the research question. I aimed to find out whether companies understand what open source related business models involve. Of special interest was finding out how the open source community aspect is taken into open source business models that companies have.

## 1.3 Research methods

The background research required for this thesis was carried out by going through relevant literature and publications available through the Oulu University library. A firm sponsored open source software development project, that I have been part of, is discussed. I interviewed the representatives of that firm as I did with several other firms situated in Finland in order to gather research material. I also selected one successful open source software development firm from outside of Finland. The findings from these cases are combined and discussed in order to answer my research question dealing with open source software development motifs and business models among firms to form an encompassing study. The research method used is qualitative research.

In this thesis I use a framework developed by Rajala, Rossi and Tuunainen (2003) and further refined by Munga, Fogwill and Williams (2009) to appraise how firms and organizations are handling the special circumstances around open source software development for commercial purposes.

In the interviews that I have done to gather the research material for this thesis I have used the thematic interview technique described by Hirsjärvi and Hurme (1985). They state that the thematic technique suits well for interviews where the interviewees might

have to think or describe a lot to give answers to the questions or if the interviewer does not want to lead on the interviewees too much. This technique is suitable for gathering research material from a relatively small number of interviewees. In this interviewing technique the interviewer prepares by developing guiding questions to the subject matter. These questions must be loose enough to leave space for the interviewee to manoeuvre and describe matters that relate. However, the subject field for the interview must still be predefined and known to all the participants. Hirsjärvi and Hurme (1985) also indicate that for the thematic interview to succeed the interviewer must decide beforehand what he or she intends to find out and does he or she want to find out opinions or facts.

## 1.4 Thesis structure and expected results

This thesis is structured so that after this introduction chapter I discuss the topic of firms and open source business environments. Firm motivation and business logic for open source business models are also discussed. Open source developers themselves are also discussed and how they are connected to firms and organizations doing business with open source products. In chapter three I discuss actual open source business models and the framework I have used to assess those models. In chapter four I discuss and analyze my research material and how it has been collected. Chapter five is dedicated to discussing my results. The sixth chapter is titled Conclusions and in it I summarize my findings, analyze how the study went and suggest topics for further studies. I have reserved several pages for the references and appendixes at the end of the thesis. The appendixes contain in-depth descriptions of my research themes, the case companies and the conducted interviews.

The expected outcome is that the findings will indicate that while many companies use open source components in their products or do business by offering services related to open source products they do not give back much to the open source communities. I also expect that firms and organization see open source software development activities as a good way to scout for and hire new developers. I also think that companies can gain extra value from working with open source developers and communities and anticipate that the case studies will indicate that.

## 2. Background

### 2.1 Open source software and licences

Hecker defines open source software in his article in 1999 as software that has no licence fees for the usage or distribution of its source code, is free to be modified and redistributed by its users and is free to be used for derivative works. Fuggetta (2003) states that open source software can also be used to mean weaker forms of distribution of the source code, such as semi-free software and in some cases even proprietary software. For these reasons Fuggetta (2003) indicates that the loose definition of open source software can be considered to be any software that is freely accessible to the worldwide community of users and developers.

Open source software development is full of new and innovative ideas, state Watson, Boudreau, York, Greiner and Wynn (2008). It also appears to be creative destruction in action. It has opened previously untapped markets and the distribution of open source products via internet is almost cost free. Similarly in open source software development one can utilize the public as a cheap production mechanism. Karels (2003) dwells on open source software development, and he considers it is often quite different from that of traditional commercial software development. According to him it is typical of open source software development that a loosely controlled distributed group develops a program, which in turn attracts additional developers and this makes the project and the program grow over time.

According to Krogh, Haefliger, Spaeth and Wallin (2012) open source software development is characterized by three dimensions: incentives, control and coordination mechanisms. The motivations of the participants are varying and there is a multitude of them. Controlling open source software development projects happens through decision making based on common values and beliefs. Coordination hinges on the participants ability to exchange information and ideas over the internet at low cost. (Krogh et al. 2012).

There exists a host of open source licences. Some of them are very strict while others are loose in how the covered code can be used and modified. Karels (2003) provides a brief summary of open source licences in his article. A small amount of software is in the public domain and in this case there are no restrictions on its use. The usage of GNU General Public License (GPL) in open source products and code is a lot more common. It allows the distribution and modification of the covered code, but requires that the source code is made available without any restrictions. Integrating code under GPL to commercial applications can be tricky as the licence requires that all the linked code is made available under GPL as well. This is called “copyleft”. Firms need to utilize tailored business models when dealing with code licensed under GPL in order for the business models to work. Another practice is represented by the Berkeley Software Distribution (BSD) styled licences. In them the author of the code has a copyright but the licence allows the modification and redistribution of the source code with a few minor limitations. (Karels 2003).



## 2.2 Firms and open source software development projects

Open source software development will not be a good choice for all software development projects. The reason for this is that with open source companies will pretty much need to share the end results for free public use. Using open source as a development method in software development projects requires that the firm, organization or team utilizing it really has good analyzing and evaluation skills and tools. As pointed out by Behlendorf (1999), open source software development can be a good idea when the purpose of the firm is to make a product that can have supporting or complementing products which the firm can offer and make value of. Even better is if the customers that choose the open source product will have to spend a lot of resources and money if they want to change it to another product that does the same thing later on. Good examples of this are database products. A firm can use open source software development to produce the database and then distribute it for free while selling administering tools, support, training and so on. However, Behlendorf (1999) notes that producing and supporting something free will not be easy nor free of costs. For example, companies must dedicate resources for keeping contact and direction setting for open source communities. Product development outside the firm and its reach of control will also incur its own extra costs.

Companies that want to lead an open source software development project need to keep in mind that the tasks in it must be divided into small enough pieces so that single developers can work on them. According to Lerner and Tirole (2002) this can often be a great challenge for companies that want to lead an open source software development project. Companies also need to first develop a critical mass of code to show their dedication and intent so that the open source software development project would attract developers to form an open source development community around it.

An important thing when planning on an open source software development project as a firm is what open source license to use. Different licences facilitate different business models and are suitable for some strategies while being less useful for others. There are also psychological contracts in open source software development, according to Ågerfalk and Fitzgerald (2008). These contracts are mutuality, reciprocity in the form of copyleft, no legal contract for the developers, both written and unwritten norms of development that every involved is expected to be familiar with and finally, and the acceptance that open source developers are individuals with no connections to any formal organizations.

There is never a guarantee that an open source software development project will succeed. However, firms will naturally do their utmost to try and make open source software development projects succeed should they evaluate that it is in their interests. Ågerfalk and Fitzgerald (2008) consider it important for a firm leading an open source software development project to insert into the project explicit and comprehensive requirements and specifications, prompt delivery of any made promises, administering and holding of project meetings and discussions, strong leadership, support and above all commitment to the project. It is also an immense boon for any open source software development project if there is a participating firm or organization that is willing to see the boring bits, such as documentation and thorough integration testing, performed as often those are the things that will be left undone.

Ågerfalk and Fitzgerald state in their article in 2008 that firms can also help to make a project successful by being the trigger that forms a clear authority for the project, lists the current problems and does the human capital management. According to Shah (2006) companies can raise the chances of success by attracting open source developers that will hop in because they need something out of the project. Companies can also attract hobby based open source developers if there are interesting and difficult problems to be solved. Those that are involved in the project because of a need ask questions and develop problems while those involved as their hobby try to solve them.

If an open source software development project is led by a firm it is imperative that the outside developers feel appreciated and perceive their work rational and relevant for the project. An important aspect is also that the work the developers have done will be visible. Shah (2006) describes that work which will never be seen or appreciated by anyone is not worth doing by an open source developer. A firm, an organization or the project itself must have the trust of the outside actors on the fact that, if the project is completed and it has commercial value, the firm will not close the code and start demanding licensing fees. Along the same lines Deodhar et al. (2010) note that firms must make the relevant open source communities satisfied on how the projects will be led and executed. At the same time the firms need to keep their reputation, community service quality and participation intention clear and relatively unblemished.

Shah (2006) concludes that there are quite a few matters that constitute a successful open source software development project led by a firm. The first central issue is about the control of the code and who gets to decide these. The second central issue is about who gets to use the project forums and message boards and uses them the most. If firm representatives take part all the time to all the discussions it is harmful for the project. The third central issue is about source code. If in the end only the firm gets to keep and use the code it is highly unlikely that the project will succeed. The licensing conventions of the final product might also influence the open source developers interest towards the product so companies will want to define them right at the start. The final central issue is about leadership as both too little and too much of it can lead to a ruined project.

Ågerfalk and Fitzgerald (2008) discuss firm leadership in open source software development projects too. They state that firms cannot lead well, if at all, by setting up all the goals and tasks by themselves. They can set the overall goal, but there has to be a consensus among the participants on what functionalities will be added. It is as imperative for the companies to convince their own workers that working with the open source project results in something that pulls in money for the company and pays their salaries or their lacking motivation will hamper the project.

Firm success in open source software development requires that there is developer interest towards the projects the firm is involved in. A firm can garner this interest by dedicating resources towards marketing the projects as at any given time there exists a plethora of projects a developer might partake in, state Ågerfalk and Fitzgerald (2008). Open source software development projects need visibility desperately and it does not come by itself. Ågerfalk and Fitzgerald (2008) also advice firms leading open source software development projects that they should set up clear milestones for the projects. Being flexible and patient in seeing what kind of open source community and monitoring-regime forms around open source project is a must as is working with them. It is not advisable to try and force decisions but to rather try and give hints and nudge

carefully things towards desirable paths. Once a project is up and going well companies still need to keep a technically well versed person in it because the developers in the open source community will want to talk to someone like him and not just to marketing people.

Compared to the traditional method of developing the products completely inside firms, the firms need to accept that firms in open source software development have often little or none at all control over the actual direction the community developers want to proceed in, state Gary et al. (2009). They also describe that open source software development requires that companies partake as a member of equal value in the decision making process of the community, no matter how much the companies would like to dictate the direction by themselves. Credibility is earned through active participation and a major drawback is that often the release cycle is instable. The documentation can and probably will be insufficient for commercial needs and there will not be enough of deep and thorough testing. Support will also be poor if there exists any at all. However, companies can combat these drawbacks by utilizing software in their commercial offerings that is one release cycle behind the current open source component. (Gary et al. 2009).

A drawback of open source software development for a firm is that competitors can check out the product and the code so there are no real secrets. However, this means that there is a huge amount of free testers and faults are found more rapidly than any internal quality control team could find them. (Watson et al. 2008).

### 2.3 Firm motivation for open source software development projects

There are many good reasons for firms to turn existing projects into open source software development projects. The same goes for completely new development projects. Making the right products open source and choosing an appropriate business model benefits a company in the end in so many ways that they off-set any short term loss of profits, discusses Hecker (1999). Young writes in his article from 1999 that firms should be motivated to engage in open source software development because it bestows them a huge competitive advantage when utilized correctly. Dahlander and Magnusson (2005) state that the reasoning for firms to engage in open source software development projects is often one of the following: speeding up the development process, enhancing competitiveness, spreading products out widely, attracting talented developers, defining standards by having the first widely spread product in a category, gaining market share rapidly and gaining positive reputation. Some firms might also just have such an attitude that they want to develop software that is free for everyone and help open source developers gather into thriving open source communities.

It is also possible to save in development costs if free open source developers develop and generate ideas towards a product. Instead of spending the companies own resources to do those tasks, the companies can use the saved resources for developing value added features, describe Gary et al. (2009). Evans and Reddy (2002) are on the same lines. They state that firms must have a good reasoning for investing resources in open source software development projects and saving in development costs is one such a reason. They have also stated that supporting open source projects is a great way to obtain good reputation for the whole firm. Gary et al. (2009) state that open source software development can be used by firms to share risks so that otherwise risky projects can be

shared and completed among many participants. Also according to them, open source software development is a very good method to use when developing something that would not necessarily give any competitive advantage. Open source software development can also be used to make development processes for other products shorter by for example sharing components or design solutions between products.

Motivation for companies to take part in open source software development projects can also be to scout and recruit developers that take part in such development projects, if they are the kind of people the firms are after. If an open source project is a popular one there tends to be a very large pool from which to choose the most skilled or fitting developers. Also according to Lerner and Tirole (2002), open source software development projects are a good way to obtain free help in finding bugs and developing customization ideas and solutions. In a way one can think of open source communities as very large beta tester teams, state Rossi and Bonaccorsi (2005). Getting funding for big and expensive software development projects can be difficult for young fledgling companies so those companies can tactically first take part in a few open source software development projects and show their skills in them after which funding might be easier to acquire for other projects the companies wants to pursue.

Lerner and Tirole (2002) state that firms might find it a good idea to invest money and resources into open source software development projects in order to increase the popularity of the platform the project deals with and thus the firms' other related products will sell better. A reason to invest in an open source software development project might also be to hammer the businesses of a competing firm. If the open source project results in a product that eats up the sales of the competing firm, even if it is not on a field the firm doing the open source supporting deals on, it might make the competition less fierce by weakening the competing firm. Gary et al. (2009) discuss a related matter describing that firms can help in putting an open source software out to the field in order to create a market position. By placing an open source software in the market firms can leverage entire open source communities to provide enhancements and to creating marketing name. If the open source product is successful it will block out other commercial products from that market space.

A good motivation for companies to invest resources in open source software projects, according to Rossi and Bonaccorsi (2005), is to gain independence from the price and licence policies of large software development companies. It can also be a very viable business model to offer software related services by exploiting commercially code bases developed by open source communities. Rossi and Bonaccorsi (2005) continue by stating that companies can also engage in open source activities to obtain indirect revenues by selling related products.

A firm might have a platform that has multiple products so making one of those products open source might result in increased amount of attention. This increased attention from open source developers can increase the amount of developed functionalities for that product. This on turn, as Lerner and Tirole (2002) discuss, leads to increased sales and popularity of the other products on the platform too. On the other hand, once this turning a product into an open source one has been made, companies have little alternative left than to keep on supporting the open source development and community around it. West and Gallagher (2006) state that the same goes if a firm has a product or a platform that requires an open source product just to work or sell. If the development for this product is stalled or ended, it is rational to dedicate some of the

firms resources towards the product. It will almost always be more costly to develop a replacing product from the scratch than to just assign a couple of company's own developers to the existing open source project.

Shah (2006) describes that a rationale for firms to assign a couple developers to open source software development projects to expand or update them makes also sense when this helps the firms to avoid updating to a new version or changing a product that they are using. Markets might also have changed so that a firm might now be too small to compete in the field that has been in the past its primary focus. Rather than trying to pour all resources desperately into making the primary focus profitable again, the firm can turn it into a secondary focus by giving it to an open source community and making the past secondary focuses into new primary focuses that connect and benefit from the old primary focus. Lerner and Tirole also discuss in their article from 2002 that if a firm is badly lagging behind of its competitors in product development open source software development can be the stimulant that helps to catch the competitors.

Behlendorf's (1999) findings are that a main component that has been developed for open source distribution will attract many more customers than a paid one would. Also, product development for this component can happen for free outside the firm if motivated customers will further develop and expand the component and search and squish bugs in it. Furthermore, if competitors offer the same main component for a price they have a really hard time competing against something that is offered for free. Open source software development can also be used as a method to transfer knowledge and ideas from inside the firm to outside of it so that the transferred knowledge and ideas can be seen as separate elements. For this, as West and Gallagher (2006) discuss, it is good that what the firms transfer is not really central for the firms' interests. This can lead to an open source community forming around the idea or basic element. The community might generate new ideas for the firm that can be used to form totally new markets for products the firm offers. This whole aforementioned process can also be used if the firm has a clearly good idea with potential markets on its hands, but these markets are too small or will open up too far away in the future to be in any way profitable to target right now.

Above all firms need to understand that even if it is tempting to make a project into an open source one just so that it could be saved or to just obtain good reputation with a failing project, these are not really that good reasons, as Behlendorf (1999) states. Firms need to conduct deep research so that they could understand what the project needs to become and contain so that the open source software development model would work. The first step is to do a market analysis on the whole market, on the products there, on the competitors and on the other related organizations and players. Behlendorf (1999) continues that the firm needs to think what components there will be in the open source software development project so that they can map which ones will be sold, given away for free or can be gotten for free from other open source communities or projects.

A motivation for firms to take part in open source software development projects can very well be to learn from them and then move the experience and ideas from the project to the firms' own commercial products, state Munga, Fogwill and Williams (2009). Well versed and skilful users can give feedback and help which can speed up product development dramatically or give ideas for new products and business opportunities for firms and organizations. Lerner and Tirole (2002) have noted that users and developers in open source communities might think of a solution to a problem

that no one has even thought of before. If a firm is part of such a community it can lead to commercialization of the solution in a different product or project.

Firms do not need to sit idle and just wait for the open source developers to come up with ideas as they can systematically the developers for idea generation, describe Enkel et al. (2009). Holding competitions or challenges are examples of generating novel solutions to problems by utilizing the creative potential of open source developers. A firm can take the generated solutions and ideas and refine them further for the purpose of innovation development, state Enkel et al. (2009). Open source projects have a lot of actors in them and with this variety comes the chance to develop and obtain unexpected new ideas. Chesbrough et al. (2005) have discussed that often it is not profitable for businesses and organizations to try and own all these new ideas for example by patenting them. Patenting everything leads to a huge pile of unused patents not crucial for the businesses and organizations. Instead a good idea would be to think how they can be put to use so that the profit oriented actors don't need to dedicate many resources to them.

Enterprises that are not in any monetary distress can utilize open source software development projects to relax their developers. Lerner and Tirole (2002) discuss that being part of an interesting open source software development project is often a good chance to get a break from boring software development projects. After even a small such a break the developers will be more productive and efficient in their regular development projects. It is possible that the developers will also adopt new skills and ideas from this experience and thus add value to the enterprise. West and Gallagher (2006) state that it is hard to motivate the developers in companies to do this in specific projects outside their work hours for free, but it is not impossible. Companies just need to figure out how they can make it seem a really interesting opportunity.

## 2.4 Open innovation

The term "open innovation" goes often neatly along with businesses dealing with open source projects. Chesbrough, Vanhaverbeke and West describe in their article from 2005 that businesses which have such a business model that they work with or support open source projects and gain value from this are open innovation actors. In open innovation companies and other actors pool ideas and conventions from both inside and outside of the themselves and then produce value from that. All this requires that the businesses have a clear understanding what is and will be profitable for them and have a plan towards that. It also requires, as West and Gallagher (2006) state, that the workers in the companies are open and prepared for this kind of work. There are three main problems in this and the first one is how to maximize profits for the firm. The second problem is how to absorb ideas and conventions from outside the firm and the third problem is motivating innovation and development outside the firm. Solving these problems is the secret for combining open innovation and open source software development. Also worth noting is that a part of the open innovation concept is that the ideas that the business hatches can also be delivered to markets via outside channels and this can lead to added value.

Open innovation attracts firms because it promises better payoffs for money spent on innovation. Firms utilizing open innovation recognize that interesting new ideas and knowledge is not only found from the development teams inside themselves, but from

competitors, customers, researchers, other organizations and talented individuals. Rossi and Bonaccorsi (2005) discuss that knowledge sharing related to a product with an open source community is a great chance for a company even if there are competing companies that have similar products in the community. Harnessing all these sources to produce value for the business would be the ideal situation. West and Gallagher (2006) describe that traditional model for innovation has been to spend money and resources on product development inside the firm and just wait for results. Sometimes the results have not been what the companies have hoped for so they have been left unused. In the worst cases these unused results have then leaked to other companies and those companies have then successfully used them.

Enkel, Gassman and Chesbrough (2009) state that open innovation offers the ability to tap into more research and development activities and this allows a firm to outpace its rivals. However, there are risks and barriers involved that can hinder the profitability of open innovation. These can be the loss of knowledge, higher coordination costs, loss of control, high complexity and difficulty in finding right partners. Negative aspects can also be the difficulty of finding the correct balance for open innovation activities and other activities or insufficient time and resources to do well enough in all of the required related tasks. Realistically businesswise it would be wise to invest in both open innovation and closed internal research and development because of the aforementioned risks. After finding a suitable balance for these activities a firm can push out products faster than the competitors. (Enkel et al. 2009).

Chesbrough et al. (2005) discuss that there are four great outside sources of useful knowledge, ideas and skills for companies. The best sources are the companies' customers and suppliers. Researchers, competitors and other cultures/countries are also very good sources of new knowledge and ideas. Companies can always play shamelessly and just copy ideas from others but then these companies are always the late comers unless they are extremely skilled at putting those copied ideas to use. So for companies it is usually better to consult for example the leading customers, because they have used the companies' products extensively and thus they can have some ideas related to the products that the people working inside the companies have not thought about. This all goes with the idea in open innovation that businesses and organizations search for all kinds of idea rich people from both inside and outside of themselves.

A great thing about open innovation and open source software development is that they allow firms to interact with companies from other industries, state Enkel et al. (2009). This can lead to reconfigurations of existing knowledge and concepts that benefit and invigorate all the parties partaking in the development projects.

## 2.5 Open source developers

For firms one interesting point in open source projects is that part of the knowledge and skills required to complete them might not reside in the firm itself but outside of it. If firms want to get people from the outside of the firm to work on the open source projects they have invested in the firms need to consider what kind of people these people are and how to motivate them. Dalle, David, Rishab, Ghosh and Steinmuller describe in their article from 2004 that these people are usually in paid labour and want to improve the open source alternative to commercial products. Many of them end up working with open source software development projects because of the low

opportunity costs, discuss Rossi and Bonaccorsi (2005). The tools required to be part of open source software development projects are relatively common, cheap and easily accessible. A developer can partake just by having access to a computer that has an internet connection.

Studies have shown that mainly there are two kinds of people working in open source software development projects. Shah (2006) describes that they are those that work because need something from the project and those that work just purely out of hobby. Those that work out of hobby often get creative pleasure from completing open source projects. Buying the software from a vendor might be easier and faster, but it does not provide as much satisfaction. For those that work because they need something from the project it is important that even if there was a firm partaking in the project they get to use the result of the project and maybe even sell it in a form or another. They usually work on weekends or after their workdays and they put more work into the open source project the longer they have been in touch with it. According to Lerner and Tirole (2002) it is good for firms to note that open source development and developers are quite an elitist group. Even in very large open source software development projects the fifteen or so most skilful open source developers will produce up to 91% of all changes. It is worth noting, according to Shah (2006), that open source developers are interested in any one project usually for about three or four months and up to a year at the most.

Open source developers sometimes treat partaking in open source software development projects as a learning opportunity, state Rossi and Bonaccorsi (2005). They can study others and implement the work they have done anew later in a different context. Often open source developers only want to tackle the interesting and difficult tasks and problems. Behlendorf (1999) describes that this leads often to open source software development projects that result in products that are very hard to use for end users. The developers have only wanted to solve the problem at hand and an intuitive graphical user interface has not been seen as a very important function. To keep the project going, firms need to dedicate resources for doing the boring and mundane bits. Shah (2006) discusses that especially those developers that partake in the project out of hobby skip the boring parts. Their paid work might be disciplined and defined software development and they want to work on informal software development projects on their free time.

Open source developers that take part in open source software development projects out of hobby want to spend their free time with interesting but at the same time challenging tasks. If an open source project has only routine tasks these developers will move on to other projects, discusses Shah (2006). Also, one should note that these kind of developers are quite unreliable because they set up their own timetables and if they run into boring tasks, such as doing documentation, they might leave them undone without informing about it. A company cannot just do the ground work on an open source software development project and then expect that an open source community filled with brilliant developers will develop around it. Companies must continually support and direct it the projects so that the open source developers feel there is activity to be part of and that they are not the only ones doing it. (Shah 2006).

A necessary item for companies dealing with open source projects and developers is to show that the companies agree with the values of the open source movement by using open source licenses. According to Rossi and Bonaccorsi (2005) this greatly increases the likelihood of open source developers joining companies' projects. Having an active



open source community around a product greatly eases the development and distribution of the product. Most preferably firms will want these developers to work around their products without them demanding any tangible payment for their work, states Hecker (1999). Instead their payment can be something intangible such as fair treatment, freedom of work and respect. As the vast majority of open source developers does not expect the firm or organization leading a project to reward them in any direct tangible way, instead they reap their tangible rewards indirectly from revenues from other sources such as related products or services, describe Hars and Ou (2001).

## 2.6 Motivating open source developers

Motivating open source developers to work with the projects the companies want can be done in a variety of ways. Dalle et al. (2004) discuss that businesses can motivate open source developers by making the project such that the people involved in it can utilize the result to produce value for themselves. The result of the project does not need to be able to produce money as such because value can be derived from the project if it supports other tasks the developers do. Ke and Zhang (2010) are on the same lines as they describe that developers often motivate themselves with external motifs. Examples of external motivation are making profit by selling the software one day or other financial benefits, improving future job prospects and signalling capability.

Hars and Ou (2001) state that the motivation for an open source developer to join an open source project can be the need for a specific piece of software or an upgrade to an existing one. Dahlander and Magnusson (2005) describe that the motivation of these open source developers is often to elevate their social standing by acquiring experience and skills or just to help others out of good will. According to them, earning a monetary reward by some other related way is a possibility also. Rossi and Bonaccorsi (2005) discuss that an open source developer does not belong to a community necessarily just to try and climb the social ladder there, but sometimes only to have the feeling of belonging to a community.

Developers can also be motivated by being there when a project is completed so that they can add it to their resume and thus be more alluring in the job market. However Shah (2006) states that these kind of developers are sparse. Intrinsic motivation is a common theme also according to Ke and Zhang (2010). They describe that there are open source developers that just enjoy the intellectual stimulation, enjoy working in open source projects as a hobby or like to help others to improve software. Hars and Ou have stated in 2001 that this was a common trend back then too. They state that firms should not count solely on hobbyists in open source software development projects because their work cannot be counted on to satisfy the needs a software should.

Hecker (1999) describes that the knowledge of working in a project that aims to solve an important problem can be a motivation for an open source developer. Reciprocity is also a very important theme, states Shah (2006). Those open source developers motivated by it feel that if a firm is giving so much to an open source project they should also contribute something however little it is. This interest towards a project generated by reciprocity usually does not last very long but on the other hand it is very productive when measured in deeds. Some developers take part in open source software development projects because they are really interested in designing and executing an improvement to something. Often these people stay in the open source development

project to see that the improvement they have made stays relevant and in working order, describes Shah (2006).

Lerner and Tirole (2002) discuss in their article that businesses should remember to show honour to commendable open source developers that took part in a project led by the businesses. Even those that took part only a little should be mentioned in the list of contributors. Showing this kind of respect spreads good word and the current and new developers will be more eager to work with the firm in the future. Shah (2006) is on the same lines stating that a big part of the open source developers are in the projects because they feel that somebody needs and uses the resulting products. Those developers will work harder if they feel that the updates and additions they do to the code get noted and commented. Even better is if this done by their fellows. According to Hecker (1999), motivating open source developers to enlist as bug fixers can be done by doing frequent releases that incorporate their fixes with appropriate credits.

A firm can attract open source developers to join their projects just by searching and identifying unfilled markets. It is as tempting for open source developers to join such a project as it is for any firm, describe Rossi and Bonaccorsi (2005). Ke and Zhang (2010) have concluded that the best motivation for open source developers is identified motivation. The open source developers identify themselves by sharing with the development team, develop needed code and build an identity in the open source communities. The second best motivation is introjected motivation where the developer is motivated by gaining peer recognition, showing off his or hers great work and enhancing personal reputation. Hars and Ou (2001) describe that peer recognition is at its best when it is rapid and constructive. The feedback that additions receive shows that people are checking out what the developer has contributed. Ke and Zhang (2010) suggest that whether or not a participant in an open source software development project experiences satisfaction of needs for competence, autonomy and relatedness influences how energized a worker one is. Open source projects are very good at offering those as a developer can choose what to do and the group as a whole constantly improves a software. Also, forums and other used communication methods allow rewarding peer feedback and connection making.

Companies should work hard to motivate the open source developers in as many ways as they can. It is not that hard to provide external motivation, but triggering internal motivation from the developers is a much harder task, state Ke and Zhang (2010). Internal motivation is far more productive and leads to a longer connection with the project than external motivation. External motivation might only last as long as the tangible reward has not been received or the developer assesses that the cost-benefit for the project is still worth it.

## 2.7 Connections and control

When working with open source software projects it is extremely important for firms to keep up good connections to related open source communities. This in turn requires that the developers in the firms or teams have a positive attitude towards working with the open source community, discuss Deodhar, Saxena and Ruohonen (2010). Maintaining a good and fair reputation is a great boon also. Open source developers will migrate to other project quite fast if a firm breaks the unwritten norms of open source development, state Rossi and Bonaccorsi (2005).

Deodhar et al. (2010) discuss that as a software development project is made into an open source one the previous owners need to relinquish some of the project ownership to the community in order to have high accessibility and transparency. This leads to the open source community accepting the firm more easily. Dahlander and Magnusson (2005) state that project related communities do product development and innovation related to the project as long as that work seems relevant. They have also come to a conclusion that relationships between firms and open source communities can often be categorized as symbiotic, co-existing and parasitic. No firm surely wants to have knowingly a parasitic relationship to open source communities but it is very easy to lean towards it. Symbiotic relationship is the relationship where the firm can have the greatest influence over the open source community and the direction of said community. However, it is very difficult, work intense and problematic to administer this kind of relationship since for example it is hard to separate and manage the knowledge that reside in the firm and in the community, describe Dahlander and Magnusson (2005).

Firms need to think carefully about the level of control they want to have in an open source software project they initialize or lead. Dahlander and Magnusson (2005) discuss that too much control can extinguish the interest and creativity in a project and too little of control can lead to wasted potential. They also state that there are seven main problems for firms when it comes to dealing with open source software development projects and control. These problems are: respecting the values and norms of the community, utilizing licenses in a commercially viable way, attracting skilful developers and users, maintaining resources that enable the community to converse, aligning different interests about the nature of work, resolving ambiguous issues about control and ownership and obtaining acceptance for utilizing a community developed product in a commercial function. According to Shah (2006), too much control in the hands of companies in open source software development projects drives away open source developers. Those developers often only want to be in projects where they can choose the tasks by themselves and also at the same time be able to affect the direction of the project.

Dahlander and Magnusson (2005) have found out in their studies that because there is no direct control between firms and open source communities the firms would do good to utilize subtle methods of controlling the open source communities. Five mechanisms for this purpose were identified: dedicating workers to work within or with the communities, acquiring and maintaining good reputation, clearly defining and displaying the benefits of working together, maintaining and utilizing tools for working together and developing and pitching interesting development tasks for the open source community to solve.

## 2.8 A summary of the background literature

This chapter describes what researchers have discussed and described in prior literature about open source software and open source software development. Conducting this research gave my thesis a basis on peer reviewed material. It also gave me a view of the subject field and related concepts. I discussed in this chapter based on the prior literature such things as when it comes to open source software the most important matter companies should realize is that open source software has opened previously untapped markets. These markets can often only be reached with new open source

business models. Open source software development also enables companies to utilize the public as a cheap production mechanism.

I also discussed in this chapter how open source software development will not be suitable for all software development projects. Companies have to carefully assess which development projects will be better off as traditional development projects. In the following list I have summarized software development project types where open source software development works well.

- Projects that will help to sell supporting or complimenting products.
- Projects that result in products with high swap values.
- Projects that can be split into very small tasks.
- Projects which can utilize open source licenses without harming sales.
- Projects with an existing critical mass of code that can be built upon.
- Projects with high visibility.
- Projects that have flexible deadlines.

I described in this chapter how there are a lot of good motivations for companies to partake in open source software development projects. Financial motivations are the easiest to identify but there are also many other motivations. For example, open source software development allows companies to pool ideas and conventions from both inside and outside of them which is offers tremendous benefits if executed successfully. I have summarized in the following list reasons that motivate companies to engage in open source software development projects.

- Gaining competitive advantage.
- Decreasing software development time.
- Spreading software products widely.
- Attracting talented software developers.
- Defining standards.
- Gaining market share rapidly.
- Gaining positive reputation.
- Carrying out the open source values of the company.
- Generating new and novel ideas.
- Sharing risks between participants.
- Developing a product that does not give a competitive advantage.
- Obtaining free help for finding out bugs.
- Demonstrating skills for software development.
- Hampering competitors by making products that compete with theirs.
- Creating a market position.
- Gaining independence from other companies and their products.
- Enhancing the skills of the developers working in the firm.
- Stimulating software development in the firm or for the firms' products.

It is also important to understand open source developers and what motivates them to behave as they do, as I discussed in this chapter. Understanding these people better allows companies to lead open source software development projects and prospering open source communities around those projects. With vast amounts of developers there will be problems and conflicts so companies should carefully plan the control and communication methods in open source software development projects they plan to

lead. I have summarized motivations for open source developers to partake in open source software development projects in the following list. Companies can harness open source developers to work in the companies' projects by generating these motivations in open source developers.

- Obtaining creative pleasure.
- Partaking in a learning opportunity.
- Tackling difficult but interesting problems and tasks.
- Obtaining something to sell forward or to use in making profits.
- A need for a piece of software.
- Obtaining something to put into their resume.
- A feeling of reciprocity towards the project or the company behind it.
- Obtaining peer recognition and fame.
- Working to fill an unfilled market for a software product.

As stated above, open source developers can be harnessed for the benefit of companies but the co-operation of those developers hinges on the companies' commitment to the unwritten contracts of open source software development.

### 3. Business Models

This chapter describes how profit oriented companies can generate revenues from open source software development projects. A framework for analyzing this is also presented.

Chesbrough and Rosenbloom (2002) state that the term “business model” is often used, but seldom defined explicitly. They have gathered and gone through quite a few sources in order to be able to define the term in their article. They discuss that in essence business models describe how firms plan to make money in the long-term. Chesbrough and Rosenbloom (2002) describe that the functions of a business model are to articulate the value proposition, identify a market segment, define the value chain and the complementary assets needed, estimate the cost structure and profit potential of the product or service in question, describe the position of the firm in the value network linking suppliers and customers and formulate a competitive strategy for gaining and holding advantage over rivals.

Munga et al. (2009) also state that many definitions exist on what the term “business model” means. According to them some authors distinguish between a business model and strategy, while others do not. Munga et al. (2009) state that essentially a business model should define the structure of the value chain within a company that is required to create and distribute the offering the company has. It should also describe the assets needed to support this value chain. If a business model articulates these matters it also identifies the company’s position in the markets. (Munga et al. 2009).

Rajala et al. (2003) describe that a single business model deals with a single product. As such, a firm can have multiple business models that they utilize at the same time. These business models can complement or be dependent on each other. Business models can also evolve over time.

#### 3.1 Firms and open source business models

Some people might still think that utilizing open source software development does not offer any chances of revenues from products and thus there cannot be any successful open source business models. However, open source successes have proven this assumption wrong and shown that successful business models can be built around open source projects and products, states Hecker (1999). Because there are profits to be had, there is business logic for firms to invest resources in open source software development projects. Firms need to develop suitable business models for capturing those profits as traditional business models do not work as such with open source. Goth (2005) discusses that firms must carefully assess which products fit to their open source business models. He also states that open source business models should take into account that open source products can be spread widely and fast due to customers ability to try them for free. Therefore there should not be any hindrances for easy testing of open source products.

A central idea for most of the successful open source business models is that much of the value provided to customers will not only come from a firm but rather from open

source developers working in a community related to a product. These developers are attracted to that product and they will help to improve it instead of competing products. Much of the success depends on how good the involved firm is in motivating and facilitating the work of these free open source developers. Young (1999) describes that when a firm considers open source as a business model one factor usually goes unnoticed. This factor is the customers' love for choices. Open source software offers a choice unlike the established and traditional ones and this can often be what makes the customers choose open source products over other options.

Since the open source project by definition is open and the resulting product if there is any is distributed for free the businesses need to think very carefully what they want to do and offer besides the open source project so that they can produce money for the company. According to Dalle et al. (2004), it can be consulting, supporting products or training related to the result of the open source project. These offerings should be designed so that they can be offered to customers on the other side of the world without the distance costs rising to any significant levels. Evans and Reddy (2002) have noted three main ways for acquiring profits from an open source software development project. The first way is to sell supporting or complementing products, the second is to sell hardware that is designed so that open source software can run on it and the third way is to offer support services for open source products that customers feel good paying for.

Gary et al. (2009) state that it would seem that the most often used business model firms use regarding open source software development is service and support. That means the training, contracting and consulting related to open source products. For example, developers in the firm can work half their work hours with the customers and half their hours developing the open source product so that they constantly have an intimate knowledge of it which is required in contracting and consulting. A company can also be the provider of support, software tools and training for products on an open source platform. The selection and viability of business models is drastically affected by how the open source product was evolved. If the firm was in the lead when the open source product was developed, there is a much greater likelihood that the product behaves and does what the firm wants. This means that it will be quite likely that the business model that the firm wanted to use once the product is complete will be viable. (Gary et al. 2009).

It is quite risky for a firm to produce a product that fits between two open source products, as Behlendorf (1999) discusses. At some point an open source developer might get to his head that he will set up a project that aims to bridge this gap currently covered by a commercial solution. One rationale for open source software development projects for firms is to use it to develop the secondary products or add-ons to those that the firm has, as West and Gallagher (2006) describe. Supporting add-ons made for free sound really tempting but research has shown that these kind of add-ons and expansions are a lot more uncommon than open source software development projects that are tackling a central issue. However, West and Gallagher (2006) continue, if a firm makes this kind of expanding of a product possible it is far more likely that the product will stay profitable and visible way longer than otherwise it would have stayed. To make this kind of outside development happen companies need to make away with technical obstacles that prevent it, create an infrastructure that makes partaking and co-operation in this possible and to recognize and reward those that take part in it.

Weiss states in his article in 2010 that the most direct source of profiting from open source software is to use pieces of or even complete existing open source products and build on top of them. This leaves more time for firms or teams to focus on other features. Weiss (2010) continues, stating that if this is the business model of a firm, the firm should also consider contributing back to the original open source products so that their longevity is increased as this leads to a longer lifetime for the firm's product too. Simply donating code to open source projects for the purpose of increasing the lifetime of commercial products is also a totally viable business model. Weiss describes in his article in 2012 that this donated code need not be new. It can be old code that does no longer provide direct value for the firm. However, to gain wide spread adoption for the donated code the firm needs to completely give up the ownership of the code. This creates interest towards the code because of built trust that the firm will not rescind its decision to donate. It is also recommended that a foundation is set up for the purpose of managing this code, states Weiss (2012). A foundation allows different companies, organizations and individual developers to collaborate and trust each other.

Since firms and other organizations cannot use traditional profit making models as such with open source products they need to think of other models of getting revenues from providing value to customers, discusses Hecker (1999). Some tried successful models are the support seller, loss-leader, widget frostier and accessorise business models. In support seller a firm generates revenue from media distribution, branding, training, custom development and post-sales support of open source products. In loss-leader a firm uses an open source product as a no-charge loss leader for traditional commercial software. In widget frosting a firm is actually in the hardware business but for this reason wants to support open source drivers and interface code so that the hardware is easier to sell due to good drivers. In the accessorise business model a firm distributes and sells physical items that are associated and supportive of open source software products. (Hecker 1999).

Quite similar to the widget frostier business model is the hardware integrator model. In that model a firm sells hardware with open source software already installed and configured, describes Karels (2003). Karels also discusses the support seller business model stating that it has the problem that even though a firm provides support for a piece of open source software it does not own it and thus might not be able to make it do something a customer might want the software to do. Contract developer business model works with open source projects as well as it does with the traditional software project, states Karels (2003). In this model a firm is hired by another firm to do development of open source software for the benefit of the one doing the hiring. Specializing in this kind of software development can serve as stepping board for different kinds of business models later on. (Karels 2003).

Successful open source business models also include the service enabler, brand licensing, sell it - free it and software franchising business models, states Hecker (1999). In the service enabler model a firm creates and distributes open source software to support access to revenue generating online services. In brand licensing a firm charges other companies and organizations for the right to use its brand names and trademarks in creating derivative products. In the sell it – free it business model firms start their software product life cycles as traditional products but are they are converted to open source products at an appropriate time. In the software franchising business model an enterprise authorizes the use of its brand names and trademarks to be used in



custom software development and also sells training and related services for this task. (Hecker 1999).

Watson et al. (2008) discuss hybrid open source business models. Those models offer the benefits of accountability, talent base and ecosystem for the customers. For example, a hybrid model can work so that if customers want to augment a product they have acquired and that product comes with an open source licence the customers must also release the modifications they do under the same open source licence. If they do not want to do that they can purchase a commercial licence. The firm using the hybrid model manages the licences and easily competes against proprietary vendors but there is a drawback. This business model requires that the firm controls very tightly the source code of the product and usually this leads to a situation where open source developers and communities do not contribute much at all. (Watson et al. 2008).

The three benefits of hybrid open source business models by Watson et al. (2008) can be summarized as follows. Accountability means that all paying customers are protected against patent or copyright infringements and it is extremely rare that an infringement would go unchallenged for so long that it would eventually disrupt the process of product adoption. The talent base means that the developers behind the product are of very high quality because they have been hired from a huge pool of candidates from which only the best have been picked. The ecosystem springs forth quite likely by itself as soon as the hybrid model product gains a bit wind under its sails. The open source community and all the other entities working with the product contribute to a cheap marketing engine. A firm benefits from using a hybrid open source business model also because the resulting products are very tempting for customers. The products are tempting because they are free for customers to acquire and test out for a period of time of their own choosing. (Watson et al. 2008).

Subscription based open source business models are also viable according to Goth (2005). For example, a firm can develop or help in development of an open source product or just provide complementing tools for it. Then the firm can sell support and access to the tools for a yearly subscription. This ensures a steady income for the firm and the customers are safe in knowledge that they get the latest updates and tools. Munga et al. (2009) discuss that there can even be different levels of support in the subscription based open source business model. Some levels might just allow the customer to download updated tools while more expensive levels allow the customer for on-site support and such. In the commercial value-added business model a firm bundles free open source software with commercial software and sells this bundle. Karels describes this in his article from 2003.

According to Weiss (2010), different firms and organizations can also pool resources so that they all benefit in their business models from jointly created open source software. No single competitor can keep up with that and those that pooled can each differentiate to different markets around the pooled product and create value for themselves. Another possible business plan is to provide installation wizards and service for open source software, state Gary et al. 2009.

An open source business plan must also deal with the specialities of open source licenses like copyleft. Karels (2003) describes that if the open source licence allows, the business model for a firm can be the commercial enhancement of open source software.

In that business model companies modify both the open source software and the companies' own software to support each other and then sell them as a bundle.

A successful open source business model requires that companies address most if not all the challenges of working with open source. These challenges eventually always manifest because open source software development projects have so many actors in them. The business models must be flexible enough when it comes to release dates and requirements implemented per release, state Gary et al. (2009). According to them, release dates and the fulfilled requirements per release often do not go according to the planned timetable due to the amount of different actors. The business models must also take into account the testing and documenting of the open source software as it might in the end be solely the firms' task. (Gary et al. 2009).

The integration of open source business models into firms requires that the actors in the firms understand what does interacting with open source developers and communities require of the firms, describe Munga et al. (2009). The business models must also take into account the needs of the firms themselves. Firms cannot use traditional business models as such with open source, but firms can adapt those models by making open source a fundamental part of them. Firms also need to ask themselves what are the value offerings, market, revenue logic and the future implications of those open source business models. (Munga et al. 2009).

### 3.2 Frameworks for analyzing software business models

In order to assess software business models a framework is needed. This framework needs to take into account the special characteristics of software business and offer ways to disseminate the elements of software business models. The framework developed for this purpose by Rajala et al. (2003) has been used quite widely. Their framework consists of four elements which are the product strategy, revenue logic, distribution model and service & implementation models. The product strategy element describes the core product or service and the way the development is organized. The revenue logic element describes the sources of revenue and the way a software vendor generates its revenue from those sources. The distribution model element deals with the marketing and sales of a product. It defines who the sellers and marketers are and describes the sales process. The service and implementation model element contains information about how the product offering will be delivered to customers as a working solution. (Rajala et al. 2003).

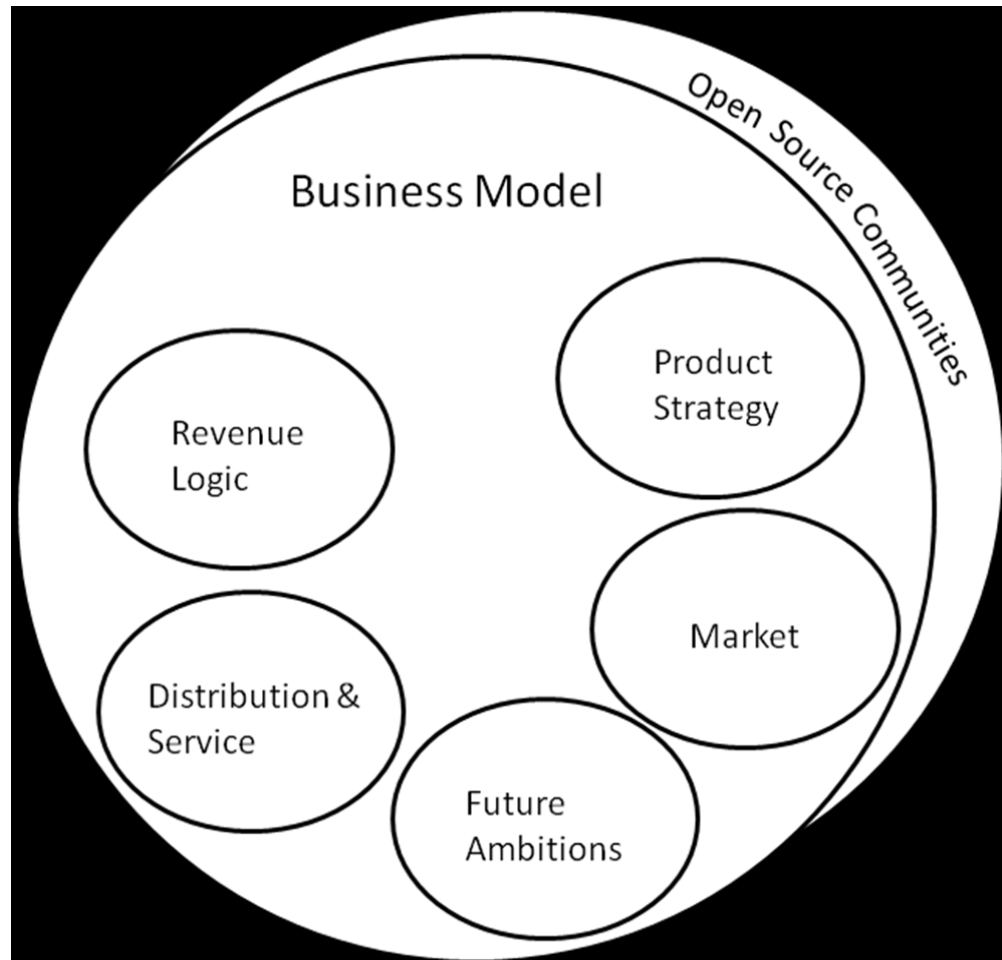
Other researchers and teams have taken this model as a basis and developed it further towards specific fields or cases. One such example is the framework by Munga et al. (2009). Their framework is focused on evaluating the impact and influence of open source business strategies on a firm's business model. Their framework consists of four components and these are the value offering, the market, the revenue logic and the future ambitions of the firm. The value offering deals with the product or service offering and how it creates value for the firm. It also includes the firm's core competence and the advantages it has. The market component describes the firm's position in the value chain and who it creates value to and how the firm can maintain an advantage over competitors. The revenue logic tells how the firm makes money and whom from this revenue is generated. It also covers sales and marketing. The future

ambitions component describes the time, scope and size ambitions of the firm and how these affect the firm's business models. (Munga et al. 2009).

Neither of these aforementioned frameworks tackle the open source community as a major theme or component. However, as discussed in chapter two of this thesis, open source communities and the developers in them can be vital parts of business models for firms and other organizations. An element of a business model, for example, can be that the community does marketing on the firm's behalf. Chapter 3.1 of this thesis describes different open source business models and many of them include the open source community in an aspect or another. Based on those models I decided that the aforementioned frameworks need to be combined into a new framework which has the community element clearly included. This framework consists of the following components:

1. The product strategy. This component describes the core product or service proposition and how they are developed. It also lists the firm's core competences and advantages. How value is created for the firm is also described.
2. The market. How the firm is positioned in the value chain and who it creates value to. How the firm will maintain advantage over competitors is also defined.
3. The revenue logic. The economic model and whom from the revenue is generated. Sales and marketing is also defined in this component.
4. Distribution and service. How customers in reality get their product or service and how it is deployed to or for them. All related maintenance, add-on and updating matters are also described.
5. The open source community. This component defines how the open source community is involved, controlled, motivated and led.
6. The future ambitions. What are the aims of the firm in the future and how they will affect the product or service in question. Also describes how the product evolves over time.

These components are illustrated in Figure 1. The figure does not include constraints such as the competing environment, customers, resource environment, financing environment and stakeholders' utilities. Those surround and affect every business model, but they are not the focus of this thesis and thus they have been left out. All the components of a business model also affect each other, but that has been left out for the same reason. For example, revenue logic can often be dictated by the state and players in the market field. The figure is an adaptation from the Rajala et al. (2003) model.



**Figure 1.** Components of an open source business model.

I've discussed firm motivation for partaking in open source software development projects in chapter two of this thesis. In principle everything firms with commercial purposes do can be categorized as something they do to increase their revenues. As such one could answer "money" to every question that contains the words "firm" and "motivation". However, in this thesis I wanted to give deeper and more varying answers than that. Firm motivation in partaking in open source software development projects contains other factors than just the increased revenues. Motivation is linked to and affects each of the components displayed in the framework in Figure 1, but it cannot be accurately put under any of them because there are so many different motivations, as described in chapter two. Motivation is more of a driving force that forces companies to act in a way or another and affects what options are feasible in each of the components of business models.

When it comes to motivation and the presented framework there is also the matter of the open source communities and the open source developers in them. Some of their motivations for partaking in open source software development projects are drastically different from those of what profit oriented firms have. As discussed in chapter two, firms can and should try to identify what motivations open source developers might have in relation to the open source software product the firm is involved with. Identifying these motivations makes it possible for the firm to focus their marketing and public relations resources better and attract more open source developers to work with the product.

This chapter presented and discussed open source business models for profit oriented firms. There are a lot of such business models and the playfield is still relatively new so that new open source business models are still invented and tried out. This chapter also presented a framework for analyzing open source business models. This framework categorizes the components of business models which are in the scope of this thesis. The topic of the next chapter is to discuss the research material I've gathered based on the framework.

## 4. Analysis

### 4.1 Research data

The literature mainly based on referenced articles in chapter two of this thesis discusses what firms would do well to consider when they do business in an open source environment. Chapter three presents open source related business models that have been found viable. Together the findings in these chapters make the tools for answering my research question which is aimed to find out what business models and motivation firms have for investing resources in open source software development projects. Finding relevant articles was not difficult but required a lot of work in order to sort out multiple, if slightly differently worded, representations of the same open source related business models.

The open source business models and related findings in chapters two and three come from studies that other researchers have conducted and then described. These researches have conducted case studies to identify open source business models and motivations but some of the business models and motivations were only theories. In order to see how open source business models work in real life, what motivates firms to partake in open source software development projects and what firms think of open source business models I needed research material from actual firms. I looked into the open source business model analysis framework I presented in chapter three. From that framework I derived questions in order to find out how firms tackle the components present in the framework. I grouped the questions under seven themes and these were the product strategy, market, revenue logic, distribution and service, community, future ambitions and miscellaneous. Interviewing firms with those themes would give me answers on how firms have approached making business with open source software. I have explained the themes and questions for my interviews in appendix A.

The research material I gathered in a sense consists of four components. I have an open source software development case in which I worked for half a year. This development project was led and managed by a large Finnish software development company. I will call this it “company A” in this thesis. The second component consists of face to face interviews with representatives of two software development firms situated in Oulu. The first of them, “company B”, works with a truly wide range of customers and projects. The other one, “company C”, instead focuses mainly on the development of their own core products in a market field dominated by the company. The third component consists of interviews with representatives of software development firms situated elsewhere in Finland. These interviews were conducted by e-mail. The first of them, “company D”, has in the past worked in various software development projects ordered by their customers while at the same time they developed their own large open source product. They dropped doing customer projects when their own product became widespread enough and built their business model around it. The second firm I interviewed by email, “company E”, has developed an open source software around which their business has been built.

As the last component I wanted a successful multinational open source software development enterprise. Arranging interviews with these kind of enterprises proved too

difficult so I had to rely on other types of research material available. I will refer to the selected enterprise as “company F”. The selection of this enterprise was affected largely by the fact that there was no lack of research material related the company. I have described all the case companies with more detail in appendix B.

For the purpose of finding suitable firms that could have open source related software business models I took a look on 75 companies which stated that they do business or research in the software development field. Of these I concluded that 26 might use or develop open source components as part of their business models. Two companies were found that had a business model in which they offered courses or instruction for aspiring open source developers. I contacted each of the selected 28 firms either by e-mail, phone or contact forms depending on their preferred contacting method. Some of them answered that either they did not actually have any significant business model connections to open source software or just didn’t want to divulge answers for this kind of study. In the end I managed to arrange two face to face interviews and two e-mail interviews besides the software development case that I had been part of. I also gathered several academic papers and other articles related to company F so that I could analyze their business models and how open source communities relate to those. I have compiled information about the interviews I conducted in appendix C.

## 4.2 The case companies

In order to answer my first research question I looked into prior literature about open source projects with firm involvement. Open source business models for firms and the matters that affect those business models were also of interest for me. I found out that the main motivation for firms to invest resources in open source software development projects in a word is money. The incurred losses from investing into open source projects are recuperated by firms in a way or another. For this reason firms have developed many business models that work well with open source software.

Other motivations for firms to invest resources in open source software development projects were also identified. For example in chapter two I wrote how Watson et al. (2008) have described that open source software development has opened up previously untapped markets.

### **Company A**

As I stated earlier in this chapter, I worked in an open source software development project for six months. I didn’t see the beginning of the project because I came to work with it a few months after it was initialized. I completed a part assigned to me and a couple of other students in the project and then we parted ways with the project before it was fully completed. This project was initialized and led by a large Finnish software development firm, company A. The company had assessed that there is demand in the market for open source software with enterprise level support and consulting. The company handles a lot of software development projects every year and they regularly use open source components and products in their development work in order to produce commercial software products. This open source software development project that I was part of was their try on a business model where they spread the developed software for free and then capitalize with support services.

In this software development case the product strategy was to develop an open source software product by mainly utilizing the in-house developers. The source code was made publicly available in neat packages so that open source developers from around the world could look at it and contribute. An open development hub was set up for ease of communication and task sharing. The firm representatives posted small and defined tasks for the purpose of garnering open source developer curiosity. About the firm's core competence and advantage can be said that as they are a rather large firm they can spare resources to careful project viability assessment. It is also well in their ability to take calculated risks and try out new business models.

This software development project was dependent on a few key frameworks on the code level. Other than that the end product would stand quite well on its own. The value created by utilizing the end product would come to the firm's customers in the form of reduced software development time, enhanced resource allocation and combined development and publication operations. The advantage the developer firm would have over its competitors with the developed product is that they would have a neat product which could cater to a wide variety of customers with standardized components. The open source community around the product would also add to the value and credibility of the product.

As the product would be free for anyone to download, try out and use the revenue logic would have to take this into account. A business model for this purpose was developed. The firm would offer customized development services, enterprise level support, consultancy, methodology instruction, training and hosting in required customer cases. Depending on the services rendered the firm would either receive lump payments or monthly support fees. A majority of the customers were expected to be other firms. However, the product was designed so that it scales well along the project size so that individual open source developers could also become paying customers. Marketing was to be extensive and include trade fairs, newsletters, web pages and discussion forums. The forums would be especially important in getting contacts with open source developers who in turn would spread the word about the product.

The product would be distributed digitally and updates and bug fixes would be offered in the central development and discussion hub. The customers could either run the software themselves or pay for developer firm to host it for them. Resources were dedicated into forming an open source community around the product right from the release of the first open version. It was the hope of the developers that open source developers would offer new ideas and improvements as well as actual code donations if acting so was made easy for them. The product was also designed so that add-ons by open source developers could be made to run with it. The open source developers would also help to market the product by making it known far and wide. A lot of activity in the development hub would also make thoughtful customers choose the product instead of competing products.

The developer firm wanted to keep the quality of the product good so all code donations to the main modules from open source developers were reviewed. They also wanted to maintain control over the direction of the project as they were the ones doing most of the development work and they had their business interests in it. No special open source developer motivation schemes were used, though there was talk of them. It was appraised that they would be of more use when the product was more mature. The



coordination and communication between the firm and open source developers was done by utilizing discussion forums. Each task or subject had its own thread so that they would not mix up and that they would be easy to track.

The only open source licence involved in the product was the Apache licence. It is not very restrictive so it suited the project well. Since the product is designed to be applicable to many different types of customers it could not include restrictive licences which could potentially lead customers to deal with copyleft issues. The main motivation for this open source software development project was to tap into a previously unfilled market with commercial gains for the firm in mind. As such development work would concentrate on the product and not in improving any related open source modules or products. A motivation was also to develop a product that could be used internally and thus speed up software development cycles inside the company in the future. With this new product in the firm's product library the firm would also have a larger offering selection for potential customers.

The drawback in working with open source software was seen in the uncertainty of how many open source developers could be attracted into the project. There would neither be any guarantee on what they would do or contribute back as they are outside of the firm's control. There is also the problem that, once an open source release has been made, companies have to keep the project open source however it fares unless they want to incur negative publicity.

## **Company B**

The first software development firm I interviewed in Oulu, company B, does software design, production and upkeep. Some of their software development projects involve a product they build from the ground up and in other cases a customer asks them to come and save a project that is trouble. They use agile software development methods and open source components and products in their software development cases which bring value to their customers. However, their view is that contributing back to open source projects is mostly a waste of time and resources. The software developers employed by the firm may contribute back to open source projects if they have used an open source component or product, but the management views that they should not spend too much valuable work time into this kind of behaviour. Any contribution back to open source projects should somehow benefit the firm and its ability to make business. The firm's representative stated that the firm's core competence is in network and server technologies and in utilizing open source code where ever they can. This leads to an advantage where they can compete with low price.

The firm is in a sense outside of the value chain as they are not dependant on any providers or technologies unless a software development project ordered by a customer is. The firm has acquired direct discussion connections to certain open source foundations and they use these to deepen their understanding of the open source components and products the firm uses. They also use these connections to suggest what they would like to see improved in the open source products. The foundations get publicity through the firm as the firm uses their code.

The revenue logic for the firm is to compete for software development projects and win contracts by being the lowest bidder. Their marketing is very aggressive and they search

constantly for customers that are looking for developers. The firm distributes their products digitally and delivers upkeep and hosting services if customers desire that. Negotiation processes are often very long so they have to have many of them going on at the same time in case of one of them falling through.

The software developers working in the firm sometimes use forums of open source projects to solve problems they have but other than that the firm does not really communicate actively with open source developers in open source communities. The only communication channel they kept open was to an open source software development project that greatly benefitted the firm. The firm representative stated that if the firm would lead an open source software development project in the future the decision making process would have to be very dictatorial so that they could be sure of the result of the project. However, they view open source software development projects to be such long processes that they do not want to partake in them as they are content in using working components and products that others have created. They plan to use this method in the future too as they cannot afford anything else but to channel all their resources into efficient software development.

When it comes to open source licences they avoid restrictive licences and will not even consider code, components or products that might impose copyleft. This firm might have a bit mixed feelings toward open source software development but in their opinion it is the current trend and market situation. They have to compete by offering low price and open source components and products are a good option for monopolized commercial products. Customers have also gotten more informed and actually are able to demand products that include open source elements for cost saving purposes. In those cases they have to assess if utilizing open source components would bring the total development costs down.

The hardest part in working with open source components and products is the assessment of quality and future. Documentation in open source software development projects is often quite sketchy and the developers might lose interest and leave the project drifting. The firm sees this as a quite realistic danger. However, they do see the benefits in using open source software. Benefits such as increased reputation, knowledge, experience and vision on such fields like software technologies and programming languages.

### **Company C**

The second firm I interviewed in Oulu stated that their main product strategy is to sell their own software product. They also produce implementations and special modules for clients upon request, but that is secondary for them. Their product development method is such that they give quite free hands to their software developers. These developers are free to visit and act in open source communities if they perceive this to help the current software development task. Besides open source components the developers have been acquiring work motivation and pride by utilizing open source components as part of the commercial products that the firm sells. The management has noticed this connection. The software development in this firms therefore happens so that at first the developers will check if there is a relevant open source product or component available and then decide if it can be used as such or if it can be improved by the developers in the firm so that it will become usable. The firm perceives that their core advantage is in offering

innovative products to their customers. The customers' benefit is in reduced development time for their own software products.

The firm has positioned itself so that they are not dependent on any other companies or organizations. In fact, they have reached such a market dominance with their products that many customers are dependant of them. They maintain their advantage by pouring significant resources into research activities and further development of their products. Their revenue logic is to have their customers pay for the products either on a yearly basis for support services or as a lump sum when the products are delivered. Marketing is done by visiting potential customers, attending conferences and trade shows. Open source developers and communities also play a part in marketing the firm's products. The firm offers their products for usage in open source software development projects for a limited time after which they ask if the project wants to keep using the products with support provided for a fee. In this way the open source developers talk to each other and to developers in other projects about these products and free marketing is generated.

The firm handles the distribution of their products digitally. Informing customers about updates to the products is done via the internet. Open source developers very rarely contribute to updating some the products because of the specific field of the products. Other products the firm offers have been designed open source developer friendly right from the beginning and the firm offers interfaces for open source add-ons. The firm stated that if they assess that offering improvement ideas or actual executed improvements for open source products or components will create value for the firm they will do those thing gladly. The firm employs a dedicated person for centralized communication and public relations toward open source communities. When the firm leads an open source project or partakes in leadership duties the firm demands a coordinated repository for related code. Utilizing dedicated forums for open source projects has been found a good communication method.

The firm does not have experience of turning commercial software products into open source products, but if such a time in the future would come that they assess that this would produce more value for them they would do it gladly. The representative of the firm stated that they tend to avoid open source licences that could infect their whole product, such as GPL and copyleft. This is unless a customer has asked for a product where GPL and copyleft do not hamper using it. The representative stated that the hardest part in working with open source projects outside of the firm's control is the assessment of said projects. Sometimes it is hard to see if an open source project will be developed by its current developers in the future or if it will cease to be developed and supported in a year or so. In a worst case scenario the firm could end up maintaining an open source product alone because all the other developers have moved on to other projects but the firm requires that the product is up to date and working. The representative also stated the major drawback in working with open source software products is that you get so many things that you did not want and these often pose an information security risk.

The main motivations for partaking in open source software development project for this firm are efficiency and resource saving in software development. Also, some open source libraries or products just are so widespread that the firm has to support them. The firm also perceives that benefits of partaking in open source software development projects include increased knowledge and know-how for their software developers as

open source code tends to be very portable. It is also extremely good exercise for their developers.

### **Company D**

The first e-mail interview I conducted was with the company D. The firm is based in Finland but they have a global business model. They have developed a completely open source software product which they keep updated. Their business model is to try to make this product as globally widespread as they can and offer other products and services to firms and organizations utilizing that open source product. Open source developers have made some minor additions to the code of that main product. Most of the benefits the firm enjoys from working with open source developers come from the hundreds of add-on components those developers have made. Those add-ons have been a major boon for the sales of the main product.

The advantage the firm has against its competitors is the quality of their main product, its ease of use, openness and free usage. They have also differentiated themselves from their competitors in a positive way as the competitors' similar products all require licences in order to be used. Their main product creates value for the firm in the form of marketing and positive reputation. The firm is very dependent on certain enabling technologies and the firm's customers are dependent on the firm's main product. However, since the main product is open source, technically the customers are free to swap out of using the product at any time. Due to the openness and licence free usage of the product customers rarely if ever switch to commercial products once they have started using the open source solution. The vibrant open source community also helps to keep customers as users of the product. The customers obtain value from the firm's main product in the form of more efficient software development cycles and less resources spent on software developers having to learn the workings of new development methods.

Company D earns its revenue from monthly support fees and from offering consultation services and training. The customers consist mostly from other companies and organizations. The firm does not just stand still and wait for the customers to come knocking. They do a lot of marketing by partaking in trade fairs, organizing meet-ups, posting newsletters, communicating via social media and their web pages. They also dedicate a lot of resources to product development and competitor analysis so that they are able to develop their product and service offerings to directions in which their competitors are weak.

The firm's products are made available to their customers as digital downloads. According to company D this is all but the written rule for open source products. A designated location also allows for ease of access to new updates for customers and open source developers alike. The firm sees that open source software developers are a critical part of the success of their main product. These developers offer information about which functionalities are needed and are important. They also represent potential customers. Part of the open source business model of the firm lies in the realisation that many of the open source developers work in paid labour. They work in enterprises and organizations and once they have tried the open source product in their free time they are very prone to endorse their workplace to adopt it.

This firm, as the others I've interviewed, wants to keep the open source products they develop in tight control. It would be too big a risk to relinquish the leadership responsibility of the projects to outside of the firm. On the other hand, if the firm partakes in developing open source products which do not originate from the firm, they are more than happy to work with steering groups in project coordination. When it comes to open source and licences the firm requires that the components and products they use and integrate must be under truly open licences, such as Apache or Intel Open Source licences.

The main motivation for Company D to have an open source business model is to increase their reputation which in turn will create more revenues for the firm. The firm is focused in developing their own products but the firm and the employed software developers also work in and contribute to other open source software development projects. Preferably such projects that connect in a way or another to the firm's own products. The firm representative whom with I did the interview stated that the drawback in open source software development projects is that the road from a product under development to revenues from support and consultation services is relatively long and challenging. They noticed and anticipated this already when they chose the open source business model for their main product. However, the firm employs energetic software developers who view open source as a fun opportunity to change the world and are not afraid of facing difficulties.

### **Company E**

The second e-mail interview I conducted was with a representative from the company E. The company is based in Helsinki. Their product strategy is based around an enterprise resource planning system. Their main business is in offering services. They get paid for setting up this system for their customers, developing specific modules and offering support services. The resource planning system is open source software. Most of the development work has been and is done by the firm itself but some work also comes from open source developers.

The firm's core competence is in the knowledge of how the enterprise resource planning system they have developed works and how they utilize open source code. There is a big advantage for their customers in getting the product for free and independent of any vendor. The customers can enhance their business and have increased revenues with less resources by utilizing the software product. Company E maintains advantage over competitors by listening to the needs of their customers. The firm representative stated that they are not completely dependent on any other providers or developers, but it is very important for them that certain open source software products stay as such when it comes to convenience. If these products were turned into commercial ones the firm would need to do a lot of work in order to replace them.

Company E earns money by charging their customers per hour of work spent working in commissioned projects and in the case of support services by a monthly fee. The firm does marketing work in the internet and by approaching potential customers. Active participation in discussion forums has also been found a good method. Satisfied customers also often refer the products of the company to other firms. Most of the development work that the firm does is done at their offices but some customer cases require visits. Support and installation services are also handled from the offices. The

enterprise keeps enterprise resource planning system up to date but open source developers may add functionalities by partaking in the work done at the development hub online.

Company E tries their best to garner open source developer interest so that reciprocal benefits could be attained. Communication happens via a development hub and newsletters. The enterprise keeps the reins of the project solely in their control as they want to uphold the high quality of the product. They audit all code donations and additions and decide if they are up to the standards. As their core product is open source in nature they also try to move to open source software in other areas of their business and development tools. Besides the monetary compensation that they get from their customers the firm's developers feel that they obtain knowledge, skills and reputation from working in open source software development projects.

The main motivation for company E in being part of open source software development in such a way as they are is to attain added value for the firm. The enterprise wants to develop universal solutions so that other developers can concentrate in different functionalities and services. The developers in company E also try their best in sharing bug fixes and additions they do to open source components and products developed by other actors. The firm feels that there is one drawback in working with open source software development projects and that is that some customers find it very hard to understand why they have to pay for services if the software is free.

### **Company F**

Company F is a very large and successful software development firm with offices in many countries. They have many products which are open source and those are sold mainly to enterprises. Essentially the products are free as they are open source, but many enterprises are willing to pay for tested and certified open source based products. Company F also offers services related to those products. Their product development method is such that they invest greatly in it themselves but also all but require input from open source communities that have been nurtured around the products. The firm works as a sponsor for several open source software development projects which are in a way or another connected to the firm's products. New ideas and solutions are in a sense tested in those open source projects and then when they are refined enough the ideas and solutions are also adapted in the firm's core open source products.

The core competences for company F are in their knowledge and skills related to open source software development, open source products and open source ecosystems. The advantages they have over their competitors are their strong open source community connections, encompassing selection of products and technologies, lively ecosystems around their products and strategic partnerships and alliances with other companies. Their products and services create value for the firm in the form of revenues, increases in technical skills and knowledge and due to the nature of their products, ideas for new products.

Company F is dependent of the value chain. The company has acquired technologies and other companies so that it could be independent of the value chain if it wished to be and transform its business models. The company has, however, positioned itself currently so that it is dependent on several hardware manufacturers and software

developing companies. This dependency rewards the company with vastly larger markets. Being dependent of open source developer participation in testing and code development also makes product development cycles shorter and increases the potential customer base and marketing force.

The firm creates value for its customers by enabling them to conduct their own businesses and functions with competitive advantage. The firm maintains advantage over its own competitors by utilizing open source communities the firm has dealings with. Co-operating in open source software development maintains a good reputation for the firm. The firm also utilizes open source software development as a way to tap into a large and innovative developer pool for new ideas. Company F also maintains an ability to adapt to a customers' needs and develop extensions to their software according to those needs. Open source communities also provide the firm a huge free marketing engine. The firm utilizes open source software development as a way to test and fine tune their products and this leads to top notch reliability and to a price advantage. If these advantages weren't enough the firm has also formed strategic alliances and partnerships, which were already mentioned. The firm has also had time to learn the value of quality in support and consulting services so they have paid attention to those and turned them into competitive advantages. Their subscription based services are also very competitive and as an added bonus they have made sure that their products are compatible with third party products.

The revenue logic for company F is to sell support services for the open source products they provide for their customers. The services are subscription based and there are different levels of support which entitle to different levels of service in terms of updates, help desk, functionality enhancements and upgrades. The company also offers training and training material, certification and migration services. A central part in their business logic is how they are connected to open source communities. Without these connections the firm's revenue models would cease to work as they are now.

The company markets and sells its products and services by itself and by indirect channels. Indirect channels include hardware vendors who bundle up company F's software products with their own products. Company F conducts marketing by sending out newsletters, contacting potential customers, attending trade fairs and events. The firm also organizes a lot of events and conferences by itself as it is big enough to do so. Some of those events are held purely for marketing purposes while others are held with more noble goals in mind. Open source developers get their fair share of attention too and the developers treat the company with free marketing. The customers get their products as digital downloads or bundled with hardware that is purchased from an affiliated vendor.

Open source communities are strongly involved with company F. The company sponsors open source software development projects and actively seeks to partake in them. The firm balances between the needs of the open source communities and paying customers. The firm might do the majority of software development when it comes to their central open source products but open source developers contribute a lot of additions and extensions. Communication channels are kept open and tended with care to relevant open source development projects. The firm takes part in them when needed or if the firm sees financial gains in taking part. The firm garners open source developers to its projects by arranging seminars, conferences and other events. Newsletters, social media and white papers also play a role. The company has a hub-like website where one

can easily find what relevant is going on and where. The company is very active at finding new ways to connect to open source developers, especially to fledgling ones.

The firm controls open source projects by partaking in steering committees or working at the lead of the project. The firm motivates open source developers to partake in relevant open source software development projects by doing specialized marketing with this purpose in mind. They've made contacting the firm's representatives easy and they invite open source developers to sponsored events. They also sponsor interesting projects monetarily if such a thing is required. The firm tries to be visible at universities and it organizes workshops, symposiums and presentations. Partaking developers get back a lot from the projects such as the ability to use the product they've helped to develop. Company F also tries to hold a position where open source software developers want to partake in the firm's projects because it is good for the firms overall reputation.

Company F requires that open source software development projects they are a part of utilize open licences as the firm's customers need to be able to modify the open source product they acquire. The firm is very motivated in partaking in open source software development projects because it helps the firm to grow and stay competitive. Open source projects also help to broaden potential markets, attract new skilful developers for recruiting purposes, generate novel ideas, enhance the research and development activities and generate positive reputation. Being part of open source projects helps the firm to see where their competitors are and connected open source software development projects help in selling the firm's products. The firm also believes that there is special value in offering an open source option for potential customers.

### 4.3 Analyzing the case companies

I have divided this sub-chapter according to the components in the framework for analyzing software business models that I presented in chapter three of this thesis.

#### **The product strategy**

The case companies had relatively similar product development methods. They all wanted to utilize open source code and products in their software development endeavours because they perceive it to result in decreased development time and costs. However, due to their business fields, attitudes and products some firms were able to enhance their software development better than the others. Companies A, B, D and F worked with software development projects from many different fields and were also willing to take on service and support projects besides plain software development. These companies have potential for connecting to a lot of open source software developers. Companies C and E focused on narrow markets with their products and services and thus had a smaller open source developer pool to tap into.

The software development method and open source developer involvement differed between the case companies. Company A had open source software developers in mind right from the get go because their assessment identified them as a resource they could benefit from. As such, company A tried to make their involvement with the project as easy as possible and encourage them to contribute. Company A also purposefully



wanted to seem as a good open source contributor with this project. Company B utilized open source code and products in their software development projects as well, but they did not let open source software developers partake in the company's development projects. Company C utilized a somewhat chaotic method with open source code and products. Each developer working for the company was free to utilize and act in open source software development projects as they saw fit as long as it could be said to benefit the company's software development projects. Companies D and F worked hard to maintain good connections to open source developers and to motivate them to partake in open source software development projects relevant for the firms' interests. The firms' own developers did a majority of the development tasks regardless. Same went for company E, though they had not tried as hard in attracting open source developer interest towards their product.

Companies A and F were large actors so they had a lot of resources to spare for open source software development projects, including a large pool of skilled in-house software developers. Their advantages also included refined and thorough assessment processes and personnel. They can also afford to take calculated risks without the danger of bankruptcy. Of the two, company F was better oriented and connected for working with open source communities. Company B's competence in network and server technologies utilizing open source code resulted in a pricing advantage. Companies D and E likewise trusted in pricing advantage resulting from open source software. Company C stated that their advantage is not necessarily in open source code but in innovative solutions which could be developed with the help of open source communities. Companies A, F and E also saw the advantage over competitors in constant product development for the good of the customers.

All the companies stated that their products created value for them in the form of revenues. Companies D and F also stated that their products create value in the form of positive reputation and marketing. The products of all the interviewed companies relied on widely utilized open source solutions. They could produce their own solutions to replace the open source ones, but that would be waste of resources unless the open source ones would somehow cease to be free to utilize.

### **The market**

Even though the interviewees stated that they were dependent on widely utilized open source solutions, all of them, except company F, considered themselves outside the value chain. Company F was dependent on certain other companies because the company had made strategic alliances and partnerships which helped to vastly increase the potential markets. It felt a bit contradictory listening the firms stating that they are outside of the value chain because they can stop using open source software at any time.

All the companies interviewed created value with their products and services to their customers by enhancing or enabling their activities. Like company B, whose representative explicitly stated that open source software allows them to offer cheaper development services, the interviewed companies created value for their customers by being efficient software developers. The representatives of companies A and D stated that active open source communities around the firms' software products created extra value for the firms' customers. Company F went as far as to state that their ability to

create value for their customers is largely dependent on open source communities and the developers in them.

There seemed to be a trend among the interviewed companies. The companies stated that they maintained advantage over their competitors by dedicating significant amount of resources into product development. Another trend was the trust in that carefully tended open source communities around the firms' selected products would help to maintain the firms' advantage over their competitors. Company F had truly embraced this and made it a reality. Company D was working really hard to do the same and while they weren't yet at same level of open source community activity, they were really close. The rest of the interviewed companies were doing fine without massive investments in engaging open source developers, though there was a certain awe to be felt in the interview situations towards the more successful companies.

### **The revenue logic**

The interviewed companies had such business models in relation to open source software products that they offered customized development, support, training and other services. Analysis showed that a large majority of the customers would be other companies. Companies A and F were able to provide a wider variety of different services due to their sheer size compared to the smaller companies. Open source communities played a large role in the revenue logic for companies D and F. The communities also played a role in the revenue logic for companies A, C and E, but not such a major one. The interviews showed that the companies did not need to be extremely large in order to be able to base their business models on open source community interaction. However, long term commitment and vision were requirements and those companies that were only able to plan and commit into open source software development projects for the next half an year or less were not able to leverage the communities very well in their business models.

Company C had an interesting business model in the sense that they targeted open source developers. However, open source developers did not easily invest money into software products so the company had to lure in the developers with free trials of the company's products and services. Company D saw open source developers as different kinds of potential customers. They wanted open source developers to try out their products because they would in all likelihood end up recommending the companies they work in to switch to company D's products. The other interviewed companies did not explicitly state that they would perceive open source developers to be a group they could market something to.

The interviewed companies had thought about open source communities and how they could be harnessed to spread the word about the companies' products and thus provide free marketing. Especially companies A, C, D and F worked to make that happen. It seemed that the companies knew there was no single best strategy in utilizing open source communities for free marketing and opted to activate the communities utilizing as many ways as possible. Especially company F shined on this field. A reason for why many different strategies was needed was that the developers in open source communities vary so much in motivation and character.

## **Distribution and service**

All the interviewed companies preferred to provide their products as a digital download. Some customers needed physical installation discs, but those were in a clear minority. An online depository for software project related files also allowed the companies to coordinate the contributions of open source developers more easily so that unintentional branching of code wouldn't happen.

## **The community**

Company A designed its product so that open source developers could develop add-ons to the product developed by the firm. The enterprise didn't want open source developers to control the actual software product because the development project could veer off its desired course. After all, the enterprise wanted to do business with the software product and a software product veered to a strange direction and with hastily scribbled documentation would not attract customers. Of the interviewed companies company B let open source communities affect its software development projects the least. The company preferred to utilize open source code and products while committing as few resources as possible to giving back. The company credits this to the tough state of markets they act on. Company C was doing so well that could easily afford to let their software developers slack a little and use their work hours working in open source software development projects. Company F also could afford to let the software developers working for it to invest work hours in other open source projects than the central ones for the firm's interests. However, from what I read about the firm I understood that the firm wants these development projects also to serve the firm's business models in a way or another.

Of the interviewed companies A, D, E and F had grasped well that firms can use open source communities and the developers in them as a huge research and development pool. The firms did the initial product development and then they would let in the open source developers who would point out new ideas and give improvement propositions. Company F had gone much further in harnessing the power of open source software developers than the other interviewed companies. Company F had cleverly made separate branches of certain open source products in which new solutions could be tested out and then introduced to the more stable and business model central open source products. Company F also sponsored a lot of different kinds of open source developer meets, conferences and other events. This does make the firm seem very noble and is good for public relations but it is motivated by eventual financial gains for the firm.

Company F was the only one of the case companies who targeted software developers which were not yet familiar with open source or at least not had not partaken in open source software development projects before. The enterprise has a long term goal where they intend to increase the amount of active developers in open source communities because this will be beneficial for them. Due to the firm's introductive efforts the new developers will most likely enter communities relevant for the firm's interests. I see this as a relatively long term goal and a method which requires an enterprise with a steady financial base. I also think that it is a very cunning method.

Companies C, D and F stated that they have felt communicating with open source developers so important that they had designated persons whose job were to specialize in that kind of communication tasks. The companies C, D and F had found that utilizing newsletters, white papers and social media were efficient ways of keeping developers interested in open source software development projects relevant for the companies' interests. Company A had carefully analyzed what methods it would use but the open source project was not yet far along for these methods to bear fruit. Companies E, F had come to a conclusion that it would serve the companies' interests well if they could host and control discussion forums where most of the open source community activity related to the firms' products happened. Company A also specifically wanted to have active discussion forums related to its product because they perceived them to make customers interested in the product. Company B was interesting in the sense that while they did not want to spend resources on open source development they wanted to keep a private communication channel open to a certain open source development project. This channel helped the company to affect the direction of the development project so that the company would benefit.

It was clearly important for the interviewed companies that once they started or partook in an open source software project the project would not sidetrack to some wild directions. For this reason companies A, B stated that they wanted to maintain control of the open source software products in the firm. The interviewee from company B stated that if their company would be doing the majority of the work in an open source software development project they would also demand a total control over it. This attitude was clearly more apparent in the smaller companies such as companies B and D. They have to invest so much to each and every software development project that they cannot really afford to lose them or let them be sidetracked. Company F had navigated itself to such a good position that they could afford to not be the entity in control in open source software development projects. It would have been interesting to see if other large software development companies with open source business models were in a similar situation.

In open source software development projects outside of the firms' control the firms clearly preferred projects that had distinct decision making structures, like the interviewee from company D stated. In most cases it would be either be too troublesome or too risky for the firms to work with projects that had no clear decision making structures. The firms always take a risk when they participate in development projects so there is no need to increase the risk by partaking in projects that seem to be on shaky foundations. Company C wanted open source software development projects to have a coordinated repository for code so that the projects could be kept track of. The other interviewed companies felt like that too and companies like C, D and F were open to helping projects in establishing coordination practices and platforms.

Motivating open source developers to partake in the companies' open source software development projects was seen as a difficult task in many of the interviewed companies. Company A had assessed and researched them but had not put them into action very well. The same mentality was rampant in the other companies, except in company F. Keeping open source developers informed with newsletters and such and organizing meets now and then was seen as enough. Company F however tried to bring the projects to the developers and potential developers with specialized and targeted marketing which was much more comprehensive and intense than what the other companies described doing. Company F saw motivating open source developers to partake in the

firm's projects as a good way to increase reputation. The other interviewed companies saw this as a possibility which they could try to capitalize in the future while company F was already doing it. One could argue here that company F is much older and with more solidified business models than the other interviewed companies and thus has had more time to develop and embrace good policies not absolutely critical for making money. Company A could be in a same kind of situation as company F if they had been working with open source business models longer.

None of the interviewed companies stated that they had studied open source developers as to what kind of people they are and what makes them tick. They relied on generally available information and impressions in this matter. The firms saw that the impressions they had now were enough to understand how they could motivate open source software developers to partake in the firms' projects. Company F might have done this kind of studies, but it didn't turn up in the research material I collected towards this thesis.

### **The future ambitions**

Most of the case companies had not planned very far into the future and they stated that this is due to the difficulty of foreseeing how the markets are in a year or two. Making plans that would stretch even further in time than that would not be reasonable, stated the interviewee from company B. Company C saw that it could be a very viable option to turn some of their currently commercial products into open source products once the sales drop to a certain level, but that is not happening in the near future in all likelihood. Company F had already done this kind conversions in the past and they have benefited from those greatly. Worth noting is that they only did this after they had other products which brought in money through service provider open source business models.

### **Motivation**

According to the research material I can say that the main motivation for the firms to engage in open source software development activities is money. The companies however are taking different routes towards that goal. Company D for example saw open source software products as a way to increase their reputation and this will lead into more sales. For companies A, B, C, E and F open source products represented a new market with relatively few competitors for the companies. The companies had also gathered feedback from their customers which indicated that open source is a trend they have to hop along or be left to lick their fingers when customers come calling. Interviewees from companies A and B both stated this. The interviewee from company C stated that some open source products are so widely spread and used that commercial companies and organizations have to have employees that can work with them. Company F also saw that being part of open source development projects sponsored or led by their competitors is a great way to see where they are at.

### **Miscellaneous**

The case companies tended to avoid open source licences that could be troublesome. The interviewee from company A stated that working with copyleft issues would have required too much resources so products and components with easy to deal with

licences like the Apache licence were selected. The companies A and B also noted that was not just them who would have to worry about copyleft issues in their products but their customers also.

The drawbacks of open source software development projects were apparent for the case companies. The unpredictability of open source developers was a major issue for companies A and B. The companies also saw the finality of turning projects into open source ones a drawback. Once a project is turned open source the decision cannot be reversed. Companies B and C stated that open source projects always come with the possibility that the projects will be abandoned by everyone else but the firm involved and this is a major drawback. Company E had ran into issues in their service provider open source business models when their customers had not understood why they have to pay for the firm's services if the product is free.

The companies identified each several benefits in open source software development projects. The companies stated that they would benefit from open source projects the most if they were leading them, but taking part in projects that had already been going on also provided benefits. Identified benefits included such things as programming knowledge, increased reputation, greatly varying development experience and other skills. The interviewee from company D went as far to as calling open source software development an unique chance to change the world for the better and thereby a superior method for generating work motivation. In the interviews there was a noticeable connection in how many benefits the company representatives listed and did they view open source software as a chance for something new or a trend that had disrupted their regular business models. One thing for all the interviewees was certain and that was that there is no going back to a world where open source software does not play a role in the software markets.

In this chapter I presented how I collected my research material for this thesis and what that research material is composed of. I presented the material and then proceeded to analyze it. I analyzed the material by combining the material from the different sources and discussing the differences and similarities between the sources.

## 5. Discussion

### 5.1 The analysis framework

The framework for analyzing open source business models presented in chapter three of this thesis worked very well in thematic interview situations. I had prepared some guiding questions for the interviewees under each of the components in the framework. I had also some miscellaneous questions that were helpful for forming big pictures of the interviewed companies. The questions under the components are described in appendix A.

The framework consists of six components and these served well in separating themes so that the interviewees proceeded smoothly from one theme to another while fully discussing the matters under each of the themes. The interrelationship of the components in the framework was apparent even before I conducted the interviews for gathering the research material for this thesis but I decided not to include it in the interviews specifically. That did not hinder the interviews as I expected since the components can easily stand on their own without requiring links to the other components in interview situations.

I have divided the discussion in this sub-chapter according to the components of the analysis framework in order to make the discussion easier to follow, as I've done in previous chapters.

#### **The product strategy**

The value creation for the firm in question under the products strategy component in the framework was easy enough to understand but who the firm creates value to was not so. It was hard for the interviewees to understand the theme in this question and it was also the hardest question to explain and define. This led to haphazard answers compared to the other questions that the interviewees could easily explain in relation to their companies. I was quite unhappy with that question but after consideration decided to proceed through the interviews without modifying it and instead opted to explain it with detail to the interviewees. One goal for future work could be to develop this theme and related questions further so that it would be more understandable.

#### **The revenue logic and the market**

The revenue logic theme in the interviews gave interesting results. Especially the marketing process and utilized methods grew with company size. The larger case companies were able to organize conferences and seminars and even some original events besides the newsfeeds that the small case companies were happy using. Utilizing open source developers as a free marketing force was possible for case companies of all sizes but required belief in that it works. The older case companies were less likely to see open source developers as a viable marketing force, though there were exceptions. The study I conducted indicated that open source developers can work as an efficient

marketing force but setting it up requires more time than assigning workers in companies to marketing duties. It might be slow in getting going but it offers a tremendous payoff due to the sheer possible size of marketers. It is worth noting also that it doesn't keep going by itself but requires tending as like any other marketing method.

### **Distribution and service**

The distribution and service component I had in my interview questions about the dispatching of products and their deployment. The answers to those were almost identical from each of the case companies. In future works this component could be further defined. The question about product maintenance and how open source communities are connected to that however resulted in varying and interesting answers. The case companies saw developing add-ons, extensions and connections to other products as a very good way to utilize open source communities. This was very useful and increased the visibility and lifecycles of products and didn't complicate the case companies' plans for their products as letting open source communities affect the core products would have.

### **The community**

The case companies saw open source communities around their products and the products they utilize as a thing to aspire towards. Depending on the business models of firms open source communities are of different importance levels. Some open source business models desperately require the forming of open source communities around the products in order for the models work while other models just benefit from them but do not require them. Thriving open source communities around firms' products also conjure a certain pride in firms' developers and increase noticeably their productivity. Some of the case companies saw helping open source communities profitable just for this reason.

### **Community motivation, coordination and control**

In the interviews I had included questions about how firms motivate, coordinate and control open source communities under the open source community theme. The firm representatives were not able to give very detailed answers to these questions mostly because the firms didn't have a very good grasp on how to handle them beyond the basics. I discussed how firms can control and lead open source software development projects in chapter two of this thesis without seeming too dictatorial for the participating open source developers. Dahlander and Magnusson (2005) have identified seven main problems that firms have to work out with control and open source software development projects, as I discussed in chapter two of this thesis. The small and medium sized case companies had solved these problems but the method they had been using was to solve each problem as they encountered them. The larger case companies were able to assess and plan for these problems and that indicates that experience will help the smaller and younger companies to plan in a same way in future projects.



The study I conducted indicates that firms could profit from understanding how to better motivate open source developers to partake in the firms' software development projects. Many of the case companies saw that it was enough to make it possible for open source developers to participate and that they would come and partake if they saw it worthwhile. A few of the case companies actively sought to motivate open source developers to partake and treated this as a marketing method with specialized personnel and resources. The companies that conducted this kind of activity had comparatively very large and active open source communities around their products. The goal of this thesis was not to find a connection between the size of communities, the profitability of firms' open source business models and firms' motivation activities, but it would be only natural to suspect that there is a connection. A further study is required to confirm this.

### **The future ambitions**

I estimated that the future ambitions component would be the weakest theme in the interviews before I started conducting them and that came true. The case companies had trouble planning very far into the future but what really surprised me was how much trouble they seemed to have. I should have included questions inquiring why the case companies have so much trouble planning into the future but that is in hindsight. Some of the case companies revealed that they were in such situations that they felt secure in how the future would unfold and didn't currently need to develop contingency plans for their products and services. It would also be well to question under this theme how companies plan to evolve or direct open source communities around their products and services in the future, but this thought came to me only in the research material assessment phase.

### **Miscellaneous and motivation**

I had also prepared questions and themes for the interviews that did not fit under the analysis framework's components and could not be presented as components themselves. These questions concerned open source licences, motivation for firms in partaking in open source software development projects, firms and contributing back to open source projects, open source software development projects and their special needs, the drawbacks of open source software development projects and what firms felt that they got from said projects besides revenues in the form of offered products and services. I have discussed these matters in the chapter two of this thesis and identified them as important factors to consider for firms partaking in open source software development projects.

In chapter three I discussed how motivation affects and in a way is part of each of the components of the open source business model analysis framework. Now that my study has been conducted and the results analyzed I would take motivation as a major component and develop specific questions under it. These questions would aim to identify what motivates firms to do as they do with open source developers in relation to the other components of the framework. It would have to be weaved so that it would fit into the analysis framework and that is no simple task.

## 5.2 Business logic and open innovation

The research interviews conducted on the case companies gave a positive image of the status of commercial companies and on the business logic they have in investing resources into open source software development. The business logic in investing resources into open source software development projects is in increased revenues, decreased development time, new markets and other benefits discussed in chapter three which open source business models offer. As discussed in chapter two, Behlendorf (1999) has stated that open source software works well with business models in which the firm gains revenues from offering support services or complementing products. Most of the case companies had these kinds of business models and those models were working quite well. When the case companies made software product development projects open source they did as Lerner and Tirole (2002) have advised and first developed a critical mass of code to show their dedication for the projects.

In chapter two of this thesis I discussed Ågerfalk and Fitzgerald (2008) and how they have described that in order to attract open source developers to open source software development projects profit oriented firms have to show reciprocity in the form of using open source licences with copyleft. However, the results obtained from the case companies show that this is not necessary. The companies' open source related business models worked very well even as the companies tried their best in avoiding licences that could have led to copyleft issues in the companies' products. The firms just needed to develop a good and interesting enough open source project to attract participating open source developers. The open source developers might hope for a project which is not controlled by a profit oriented firm but on the other hand they understand how much such firms contribute to the project and accept the trade-off.

The case companies had also understood that they needed to do the boring bits in the open source software development projects like documentation and the laying of the ground work. Ågerfalk and Fitzgerald (2008) have also discussed that. In chapter two I discussed how Shah (2006) has stated that open source developers want to see their work used and beneficiary for someone. The case companies carried out this quite nicely as they utilized development hubs in their projects where each contributed file and bit of code could be credited and shared. Both of the parties benefited here. The developers could concentrate on solving interesting or required software problems and get their work credited while the companies would get open source developer communities around their products.

The term "open innovation" was as strange to the case interviewees from the companies as it was for me before I researched material in the form of previous subject literature for this thesis. The case companies however did as Chesbrough et al. (2005) have defined open innovation and worked as open innovation actors by working with and supporting open source software development projects. The case companies saw that there is value to be captured by pooling ideas from both outside and inside the companies and open source projects are a wonderful tool for that. It requires careful assessment and planning as the benefits will be left few with rash acting. I listed this as one of the expected results for this thesis and I'm glad to see that the case studies showed it.

It was interesting to hear from the case companies that they worked together with other companies in some open source software development projects and then as competitors

with the same companies with other projects. Competing companies can treat each other with trust and respect in open source projects, as Rossi and Bonaccorsi (2005) have described. Open source software development projects give the chance for firms to work and innovate together with a multitude of actors and if they can balance the open innovation and internal research activities they can come out as winners. The case companies had tackled this dilemma by always dedicating more resources to internal research activities than to open innovation activities. I got the understanding from the interviews that most of the case companies relied here more on hunches and feelings than on solid analysis.

The case companies wanted to have thriving open source communities around their open source products. I didn't realize to ask in the interviews what kind of people the firms see open source developers to be and that is a bit pity. It was clear however that the case companies preferred to pay open source developers for their participation with fair treatment, freedom of work and respect, as Hecker (1999) has discussed. Only a few of the case companies had thought that open source developers need other motivation to participate than the completed product which they get to utilize. The answer to the question, that do the firms do anything special to motivate open source developers to participate in the development projects, was in most cases they don't do anything special. I think firms could motivate more open source developers to participate if the firms would understand how to motivate them better. I discussed motivating open source developers in chapter two of this thesis and how researchers have already researched this quite a lot.

The interviews with the case companies that had utilized special methods to attract open source developers gave me the impression that the companies saw making it easy to participate as the best way to motivate open source developers to partake in the projects that the companies wanted. Discussion and code sharing forums were found to be a good tool for making the open source developers able to identify themselves. Ke and Zhang (2010) have come to the conclusion that being able to identify oneself is the best motivation for open source developers. These two researchers have also concluded that triggering internal motivation in open source developers is hard but it is the best kind of motivation from the viewpoint of companies. Some of the case companies tried to cultivate internal motivation by reaching out to aspiring young open source developers when they had not yet chosen any sides in the form of software preferences. I view this as a very long term project, but it doesn't need to necessarily be very expensive for the company doing it.

### 5.3 The case companies and open source business models

I discussed Goth (2005) in chapter three of this thesis and how firms should build their open source business models on the assumption that the open source products they provide can be spread widely due to the customers' ability to try them for free. The case companies had realized this and saw it as one of the best sides of creating open source software products. The case companies had also chosen open source as their business model in some cases because they perceived that this would be an interesting and new option for their customers and for this reason the customers would choose the open source products over competing options. Young (1999) has stated that firms usually fail to see this point but my case companies acted on the contrary.

According to Dalle et al. (2004), open source business models work best when companies can offer the products and services to anywhere in the world right from the start without any significant distance costs. The case companies adhered to this by offering their products as digital downloads in almost all the product cases. Support services were also offered as online services. This makes the traditional models of sending the software or personnel physically to the customer's location lose when costs are compared. Nearly all the open source software products the case companies offered were such that open source developers could develop add-ons and extensions to those products. The case companies saw this as an excellent method for harnessing the innovation power of open source developers. West and Gallagher (2006) have discussed how this also makes the product lifecycles longer by increasing the visibility of products.

The case companies did as Weiss (2010) has advised and used ready open source code or products to speed up the time their own business models needed to reach fruitfulness. However, the case companies were not as eager in contributing back to open source products they utilized as Weiss (2010) wrote would be beneficial. The case companies were doing well even without this kind of action, though the general opinion was that they should do more of this if they have any spare resources in the future. Open source gives the chance for each and every participating actor to decide if they are going to contribute back or just utilize the results of the development projects. No real harm will come from not contributing back unless the actor not contributing back is also the leader of the project.

I wrote in chapter three how support seller is a tried and successful open source business model. Hecker (1999) has discussed this. All the case companies have used this open source business model at one point or another as it is relatively simple to understand and put into action. It is also not so far from traditional software business models where companies would also charge for the products they offered support for. Karels (2003) has discussed about the support seller business model stating it is troublesome for the firms utilizing it if they do not own the software they sell support for. The small and medium sized case companies had all come to the conclusion that this issue is best taken care by owning the software. Only the very large case companies could afford take the risk of selling support for open source products they did not own.

Karels (2003) has also discussed the contract developer open source business model. The case companies did not utilize this business model per se but the customers ordering software development nowadays are able to demand utilization of open source components in order to drive the development costs down. In those situations the firms had to analyze if they could indeed use open source code or products in their software development projects. According to the interviewed firms this was how they approached any software development projects even if their customers did not specifically ask for it. The reason for this was that once one firm is doing so and driving their development costs down all the other firms have to do it also or risk so much higher software development costs that they will lose in all bidding competitions.

Sell it – free it business model somewhat resounded with the case companies but the general response from the small and medium sized case companies was that they just cannot plan that far in the future accurately enough. For this reason it can be concluded that this business model is a model that can be kept in the repertoire of a company but used only after a business model the company has been using has ran dry. It is a

business model that can easily be transferred to from other business models and it can be utilized to bring positive public relations attention.

Subscription based open source business models were also in popularity with the case companies. These models are easy to explain to the customers, which seemed to be a problem with some of the other open source business models. One of the case companies had found a market niche in offering open source development projects an open source tool to assists in development tasks. If the development project decided to use this tool the firm then proceeded to offer support services for a monthly fee. As Munga et al. (2009) have concluded, the enterprises had also realized that offering different levels of support services could be used to lure in a more broader range of customers. Subscription based open source business models were the most versatile in the study I conducted for gathering research material. They could be utilized by a firm of any size in any market field.

Gary et al. (2009) have described the installation specialist open source business model where a company obtains revenues from providing installation wizards and service for open source software. The case companies had this kind of business model history but it was viewed as a very hazardous and risky model in such cases that the company itself did not control the open source software. All in all, the case companies were quite negative towards situations where they did not control the open source software their business model rotated around. This was most true for the small and medium sized software development companies. It is too risky to make a company totally dependent on other companies or organizations when in instead the company could have invested in something they can control themselves. Being dependent on other companies and organizations also throttles the chances to grow.

Licence issues were a big deal for the case companies, as already discussed in the chapter three of this thesis. Karels (2003) has stated that commercial companies can tailor their open source business models so that more limiting open source licences can be utilized. The small and medium sized case companies opted to not even try working with limiting licences. It was considered always the better option to develop a comparable solution from the scratch than to deal with open source products which had troublesome licences like GPL and the copyleft it imposes. The larger and more experienced software development companies however were able to build business models around more limiting licences. Even the large and experienced companies cannot utilize open source software with a limiting licence in all their projects as there are many important customer segments that require totally open licences.

The case companies also overcame the challenges of working with open source by doing most of the development work by themselves. With this method they could easily control the direction of the development projects and keep track of all the code integrated in the project as well as the documentation. The case companies were very willing to work with control groups in open source software development projects that they did not lead themselves but for business critical projects control was a must. This however once again did not apply for the large and experienced software development companies. They had the financial reserves and other resources to take risks and let the control of development projects out of the company even in projects that they started themselves. The case companies had also found it a good way to avoid the unreliability of release dates and requirements per release, as described by Gary et al. (2009), in open source software development projects by lagging a release behind in builds that they

delivered to their customers. Limiting the reliance on the work of open source communities worked best for the case companies. The open source communities can be a great boon and source of new ideas but too much reliance on them can too easily lead to catastrophes.

## 5.4 Motivations

I discussed in chapter two of this thesis how Young (1999) has stated that open source software development can bestow profit oriented firms a huge competitive advantage. This is now even more the case as the case companies saw that they have to use open source as much as they can to be able to drive the development costs down and compete. Open source is no longer an option but rather a must, except for niche markets where a firm has no serious competition.

The motivations for the case companies to engage in open source software development projects were, as Dahlander and Magnusson (2005) have stated, to speed up the development process, increase competitiveness, increase potential markets and gain positive reputation. This means that the companies had varying motivations to choose the open source software development method in order to obtain some advantage in the components of the analysis framework presented in Figure 1. Attracting talented developers didn't surface up as a major reason for firms to engage in open source development activities, contrary to what I expected. There was that kind of activity for sure, but the case companies saw it just as a side product of their other activities.

A motivation for the case companies to develop products as open source software was to utilize the open source communities around those products as a free marketing engine. In some business models the case companies had navigated their products and services so that open source communities actually worked both as marketers and customers. These kind of possibilities for product and service offerings didn't come without careful assessment, planning and a certain level of ruthlessness.

Gary et al. (2009) have stated that firms can save in development costs by using open source communities around their products to generate ideas. The case companies were motivated by this but they felt that it is better that these ideas are not inserted into the main products governed by the companies too rapidly. It is better that the ideas come in the form of add-ons so that they will not mess up the focus of the product and cause instability that could upset paying customers. One of the case companies even had an example of this as they had had to separate the more experimental open source community to work with a different branch of a product in order to keep the main product stable.

Utilizing open source communities as idea generators for already completed or nearly completed products was one motivation. A more prominent motivation for the case companies to partake in open source software development projects was to gain access to ideas already realized in the form of open source products or components. The case companies could save in development costs by partaking in open source development projects led by somebody else and contributing a relatively low amount of resources. In this way they could utilize the ideas or components from these open source projects in their own software development projects without generating negative reputation. Because open source software is by definition free companies can do this even without

contributing back anything but that would break against the norms of open source software development listed by Ågerfalk and Fitzgerald (2008). It was good seeing in the case companies that the developers in them felt that they should contribute back if they could when they used open source products and components developed by somebody else in the case companies own software development projects.

A motivation for the case companies to develop open source software was in some cases to be on a new and exciting frontier. The companies saw there was something noble in developing beneficiary software and then giving it out for free. They were motivated by this but still wanted to earn their living so they saw it only fair to obtain their livelihood from offering complimentary services for a fee. Companies with this kind attitude do well as they can garner trust from open source developers quite fast and grow with their help, describe Evans and Reddy (2002).

Evans and Reddy (2002) have also discussed that supporting open source projects is a great way to obtain positive reputation for a company, as I've discussed in chapter two. The case companies had realized this but it was somewhat hard for them do to this due to their resource constraints. Only the very large case companies could do this on a large scale. Firms have to be careful with these kind of activities as this can in worst case scenarios result in situations where firm support is demanded for a multitude of projects that the firms wouldn't really want to support and declining the requests without good reasons could result in negative reputation.

Gary et al. (2009) have stated that a motivation for firms to develop projects as open source software can be to create trust and visibility for projects that would not give any competitive advantage even if they were not open source. The case companies had this kind of motivations in several products. They had assessed that there were already products on the markets that do the same as their planned product would do but with the difference that the products were commercial. They attacked these products by developing an equivalent product which was open source and free and thus much more appealing and made money with support services. I view this as a very devious strategy. However, I also theorize that this could eventually lead to a situation where all companies have to develop open source products because the customers expect that and will not accept products that they have to pay for.

Lerner and Tirole (2002) have described that firms can increase the popularity of platforms they wish by supporting open source software development projects dealing with the desired platforms. The case companies had this motivation for partaking in or leading some open source projects. Platforms were seen as great sources of revenues because the case companies could group many products under them. This would serve a larger customer segment and lead to more profits. It takes however a lot longer to develop a complete platform compared to developing a single product and the case companies had had trouble pulling it off.

I was also very interested in how the case companies had made some main components open source and allowed open source developers freely to build add-ons and dependent components to them. The companies then capitalized in by offering high quality and supported dependent components to those main components, as Behlendorf (1999) has stated. These dependent components could easily be partly based on the work of the open source community and offered as substitutes with the difference of support services for a fee. According to Munga et al. (2009), this kind of experience and idea

transfer from open source software to commercial products is often a motivation for firms to partake in open source software development projects.

The interviews with the case companies showed me that enterprises have noticed that working with open source software development projects make developers in the enterprises relaxed and in the end more productive, like Lerner and Tirole (2002) have discussed. However, they did not identify it as a motivation by themselves directly in the interview situations. This indicates to me that it should be discussed and researched more widely so that enterprises would notice that they are already doing it and could aim to increase that kind of conduct.

The firm motivations for open source software development projects that didn't come up with the case companies, but were presented in chapter two of this thesis, included the motivations for defining standards, sharing risks, finding funding for more expensive projects and hammering the businesses of a rival firm. Included in this group were also the motivations for carving a market position, gaining independence from the price and licence policies of larger software companies, avoiding updating firms' software products and market manoeuvring by changing focus. The case companies also tried their best to be in the lead in product development and development skills so they were not motivated to turn commercial products into open source products to stimulate development with the help of open source developers. None of the case companies neither had so many good spare ideas that they would have been motivated to transfer them outside the companies in the form of open source software development, as West and Gallagher (2006) have discussed.



## 6. Conclusions

In this thesis I researched open source software development and what business logic is there for firms to invest resources in it. The answer is that there are many kind of business models firms can use to gather revenues from open source software development projects. I also found out that utilizing open source business models is viable for firms of all sizes. I collected research material from actual companies and found out that some open source business models are easier to pull off than others. Subscription based open source business models worked especially well on all market fields and with all types of customers. I found out that in order reap the revenues and benefits from open source software development projects firms have to understand what actors there are on the open source field and how they affect development projects. It is also recommendable that firms understand what motivations these actors have and how to benefit from these motivations.

I was also interested in identifying and analyzing different kinds of motivations for firms to partake in open source software development projects. Gaining revenues is a central motivation for many a firm to partake in open source software development projects. There are many motivations for firms besides gaining direct revenues and I identified these by going through prior research and by interviewing firms. The current market situation, the attitudes of the developers working for firms and the market field firms deal on affect what motivations for partaking in open source software development projects firms have.

I developed a framework for assessing open source business models in order to reliably assess firms. The framework worked well and by utilizing it I could assess business models from firms I selected and interviewed. This provided me real and current information about firms and their open source business models.

I selected firms for my study from Oulu and other cities in Finland. Oulu was the primary location for my study and I audited every single software development firm from the city in order to sort out firms that have open source software development business going on. So that the study would also include global trends I included a company headquartered elsewhere in the world. The number of case companies could have been higher but resource and time constraints prevented that. Nevertheless I consider the validity of the study to be good as there were companies of different sizes, histories, types and software development fields.

The results I obtained can be generalized quite well. The interviews with case companies showed that prior research has accurately predicted what open source business models will work well and around the world. The case companies included small, mid and large sized enterprises. The case companies worked on a wide range of markets and with many kinds of customers. This showed that open source business models are viable methods for software development companies for generating revenue regardless of the size, type and market field.

This thesis contributes by providing researchers an up to date study on open source business logic and business models and by identifying motivations that firms have for partaking in open source software development projects. The thesis also contributes

with a case study of actual firms that engage in open source software development projects and do business with them.

The study I conducted wasn't without its flaws. The value creation component of the framework and the question of who firms create value to was not as clear as it could have been and interviewees had trouble grasping it. I also should have asked the interviewees what kind of people they think open source developers to be and what firms should do with that knowledge. I also couldn't position motivation as a component into the framework. Developing the framework so that motivation is contained in it is something that I could do in a further study.

Another subject for further studies includes is developing the distribution and service component of the framework so that it results in more variance between firms in terms of what the component tells. A further study could also be conducted to see if there are connections between the company size, profitability of open source business models and the open source developer motivation activities firms do. Connection between productive motivation activities and project types could also be researched. I tried to assign the case companies into tiers of open source software development ability according to the answers, but that just didn't work out so I gave up on it. A further study could research into this possibility.

## References

- Behlendorff, B. (1999). Open Source as a Business Strategy. *Open Sources: Voices from the Open Source Revolution*, O'Reilly & Associates Online Catalog. Retrieved 7.3.2013, available: <http://oreilly.com/openbook/opensources/book/>
- Chesbrough, H. & Rosenbloom, R. S. (2002). The role of the business model in capturing value from innovation: evidence from Xerox Corporation's technology spin-off companies. *Industrial and Corporate Change*, 11(3), 529-555.
- Chesbrough, H., Vanhaverbeke, W., & West, J. (2005). Open Innovation: A New Paradigm for Understanding Industrial Innovation. To appear in *Open Innovation: Researching a New Paradigm*, Oxford University Press (2006).
- Dahlander, L., & Magnusson, M. G. (2005). Relationships between open source software companies and communities: Observations from Nordic firms. *Research Policy*, 34, 481-493.
- Dalle, J., David, P. A., Ghosh, R. A., & Steinmueller, W. E. (2004). *Advancing Economic Research on the Free and Open Source Software Mode of Production*. A Discussion Paper by the Stanford Institute for Economic Policy Research. Stanford, CA Stanford University.
- Deodhar, J. S., Saxena, K. B. C., & Ruohonen, M. (2010). Firm-oriented success factors of an open source software (OSS) product. In *Proceedings of the 3<sup>rd</sup> International Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development*.
- Enkel, E., Gassmann, O., & Chesbrough, H. (2009). Open R&D and open innovation: exploring the phenomenon. *R&D Management*, 39(4).
- Evans, D. S., & Reddy, B. (2002). *Government preferences for promoting open-source software: A solution in search of a problem*. Cambridge, Massachusetts, National Economic Research Associates.
- Fugetta, A. (2003). Open source software – an evaluation. *The Journal of Systems and Software*, 66, 77-90.
- Gary, K., Koehnemann, H., Blakley, J., Goar, C., Mann, H. & Kagan, A. (2009) A Case Study: Open Source Community and the Commercial Enterprise. In *Sixth International Conference on Information Technology: New Generations*, 940-945.
- Goth, G. (2005). Open source business models: ready for prime time. *Institute of Electrical and Electronics Engineers*, 22(6), 98-100.
- Hars, A. & Ou, S. (2001). Working for free? Motivations of participating in open source projects. *Journal of Electronic Commerce*, 6(3), 25-39.
- Hecker, F. (1999). Setting up shop: The business of open-source software. *Institute of Electrical and Electronics Engineers*, 16(1), 45-51.
- Hirsjärvi, S., & Hurme, H. (1985). *Teemahaastattelu*. Helsinki: Gaudeamus.

- Karels, M. J. (2003). Commercializing Open Source Software. *Association for Computing Machinery Queue*, 1(5), 40 pages.
- Ke, W., & Zhang, P. (2010). The Effects of Extrinsic Motivations and Satisfaction in Open Source Software Development. *Journal of the Association for Information Systems*, 11(special), 784-808.
- Krogh, G. von, Haefliger, S., Spaeth, S., & Wallin, M. W. (2012). Carrots and rainbows: motivation and social practice in open source software development. *Management Information Systems Quarterly*, 36(2), 649-676.
- Lerner, J., & Tirole, J. (2002). Some Simple Economics of Open Source. *The Journal of Industrial Economics*, 50(2), 197-234.
- Mehra, A., Dewan, R., & Freimer, M. (2011). Firms as Incubators of Open-Source Software. *Information Systems Research*, 22(1), 22-38.
- Munga, N., Fogwill, T., & Williams, Q. (2009). The adoption of open source software in business models: a Red Hat and IBM case study. In *Proceedings of the 2009 Annual Research Conference on the South African Institute of Computer Scientists and Information Technologist*, 112-121. Association for Computing Machinery.
- Rajala, R., Rossi, M., & Tuunainen, V. K. (2003). A Framework for Analyzing Software Business Models, In *Proceedings of the 2003 European Conference on Information Systems*, 1614-1627.
- Rossi, C., & Bonaccorsi, A. (2005). Intrinsic motivations and profit-oriented firms in Open Source software. Do firms practice what they preach? *First Monday*, 10(5). Referenced 24.2.2013, available: <http://pear.accc.uic.edu/ojs/index.php/fm/article/viewArticle/1242>
- Shah, S. K. (2006). Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Management Science*, 52(7), 1000-1014.
- Watson, R. T., Boudreau, M., York, P. T., Greiner, M. E., & Wynn, D, Jr. (2008). The Business of Open Source. *Communications of the ACM*, 51(4), 41-46.
- Weiss, M. (2010). Profiting from open source. In *Proceedings of the 15<sup>th</sup> European Conference on Pattern Languages of Programs*. Referenced 24.2.2013, available: <http://dl.acm.org/citation.cfm?id=2328916>
- Weiss, M.. (2012). Profiting even more from open source. In *Proceedings of the 16<sup>th</sup> European Conference on Pattern Languages of Programs*. Referenced 24.2.2013, available: <http://dl.acm.org/citation.cfm?id=2396717>
- West, J., & Gallagher, S. (2006). Challenges of open innovation: the paradox of firm investment in open-source software. *R&D Management*, 36(3), 319-331.
- Young, R. (1999). Giving it Away: How Red Hat Software Stumbled Across a New Economic Model and Helped Improve an Industry. *Open Sources: Voices from the Open Source Revolution*, 4(3).
- Ågerfalk, P. J., & Fitzgerald, B. (2008). Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy. *Management Information Systems Quarterly*, 32(2), 385-409.

## Appendix A. Interview Themes

For the interview themes I looked into the open source business model analysis framework presented in chapter three and took heed of what Hirsjärvi and Hurme (1985) have discussed. I wrote questions in the form of loose themes that would answer all the main points in each of the analysis framework components but in the actual interview situation I would let the interviewee have space to manoeuvre.

The first question was if the firm in question would like to stay anonymous. From there the questions proceeded grouped by the components of the analysis framework.

### **The product strategy**

The questions related to the firm's product strategy involved questions about the firm's core product and services and if open source communities play a role when it comes to the firm's product development method. These questions also tackled the firm's core competence and advantage and how value is created for the firm from their products or services.

### **The market**

In the market component the questions touched the firm and how it is connected to the value chain, who the firm creates value to and how the firm maintains advantage over its competitors.

### **The revenue logic**

The questions in the revenue logic component were all about business. How the firm earns its revenues and if open source code, products or communities are connected to that. I wanted also to know how the firm does marketing and if open source communities were part of that.

### **Distribution and service**

The questions under the distribution and service component dealt with the dispatching and deployment of the firm's products and if open source communities partake in the maintenance tasks in the form of bug fixes, add-ons and extensions.

### **The community**

In view of this thesis the questions under the community component were the most important ones. These questions were formulated so that the interviewees could give answers regarding the firm and how open source communities are involved with it. The questions also prodded how the firm keeps connections to open source communities and

how the firm controls the developers in those communities that are important for the products or services that the firm offers. For me personally, the most intriguing questions were about how firms try to motivate open source developers to partake in open source software development projects relevant for the companies and how the firms coordinate the developers.

### **The future ambitions**

In relation to the future ambitions of the firms the questions were about what changes the firms plan to do to their products or services over time and if they have considered turning their commercial products into open source products.

### **Motivation**

A question touched the firms' motivation in partaking in open source software development projects.

### **Miscellaneous**

For the sake of gathering relevant information which would not directly fit under any of the components in the analysis framework I had also prepared miscellaneous questions on the firms and the open source licences they use or try to avoid. I also inquired if the firms contributed back anything to open source projects from which they used code or products.

I had also questions which asked if the firms have identified anything special that needs to be consider when working with open source projects and products and if there are any identifiable drawbacks in working with them. My last question gave freedom for the interviewee to answer as he or she liked on what the open source software developers working in the firm feel that they get from working with open source software projects, products and communities besides the revenues for the firm in the form of offered products or services.

## Appendix B. The Case Companies

### **Company A**

Company A is a very large Finnish software development firm. They have a lot of workforce in Finland besides of development teams all around the world. The firm employs nearly twenty thousand people. The firm focuses in software development projects that result in banking, insurance, social welfare, healthcare, telecommunication, wholesale, retail, manufacturing industry, car manufacturing, energy production, government, logistics, media and lumber industry information systems. With such a broad spectrum of fields the firm works on it is safe from fluctuations in sales of any one product or product group. The enterprise tries to utilize open source code where ever it can because it drives down software development costs.

### **Company B**

Company B is a mid size Finnish software development enterprise. They have offices in three cities in Finland and in one city outside of Finland. The firm has not tied itself to any one type of projects or customers but instead they are willing to take on any software development project. The firm regularly uses open source code and products in its software development projects. The firm takes pride in how well they are connected to and can use certain open source software. This knowledge of open source software translates into faster software development projects which in turn serves as a way to compete with the firm's competitors.

### **Company C**

While significantly larger of the two software development companies I interviewed in Oulu, Company C is also a mid size Finnish software development firm. In Finland they have offices in two cities. They also have offices in several other countries, but a significant portion of their research and development takes place in Finland. They are focused in catering to a very narrow type of customers but they've grown to almost completely dominate that field. They also do software development projects for other fields upon customer request but the firm does not do a lot of those. Because of the field they work on, they have not utilized open source code much in the past and have only relatively recently decided to branch out to open source markets after noticing the untapped markets there.

### **Company D**

Company D employs about sixty people in Finland. They also have an office in the United States. In the past the firm competed with other software development firms for customer projects. Now the firm has a single software product which they market and develop further. This product can be used by customers on any business sector and thus their customer profiles vary greatly including in size. The software product is open source and for this reason they are extremely connected to different kinds of open source communities and individual developers. The company and the developers in it

hold open source software development on a high standard and see it as a force of the future.

### **Company E**

Company E is a firm called Devlab and they are based in Helsinki. They employ ten people and the firm has been established in 2006. Their business is based around an enterprise resource planning system called Pupesoft. The system is open source software and handles such tasks as accounting, purchasing, accounts payable and sales. Open source communities play a minor role in relation to the firm, even though their main software offering is an open source product. The firm is working on harnessing the open source communities more efficiently in the future. At the moment the firm sees working with open source communities mainly as a way to garner reputation, free marketing and software development knowhow.

### **Company F**

As company F and the example of a successful multinational open source software development firm I chose Red Hat Software. The firm is a large software development company with nearly five thousand employees. They have offices all around the world. The company was founded in 1993 and has grown steadily ever since. The firm has its headquarters in the United States. The business model of the firm is to provide mission-critical enterprise level open source solutions to its customers. The firm offers several open source software products and solutions which focus on cloud services, middleware, their signature Linux operating system, storage and virtualization. The firm is highly involved with open source communities and open source software development projects and spends a great deal of resources on these activities.



## Appendix C. Interviews

Since I worked with company A for such a long time I've had many discussions with several of its representatives in relation to the open source software development project. The main e-mail interview with company A on which the research material I collected can be said to be based on was conducted on the 30<sup>th</sup> of May in 2012. I also acquired marketing and technical brochures concerning the open source software development project at that date. The interviewee worked as a senior consultant. Because this interview was conducted well before I started my work on this thesis on full and I did not have a truly accurate idea yet what questions to ask I played safe and conducted a much larger interview in terms of the amount of questions.

The interview with company B was conducted on the eleventh of March in 2013. The interview took 50 minutes. The interviewee worked as the chief finance officer of the company. The interviewee was also responsible for running the Oulu branch of the company.

The interview with company C was held on the 25<sup>th</sup> of March in 2013. The interview took 45 minutes. The interviewee worked as the head of product development.

The e-mail exchange with company D occurred on the 27<sup>th</sup> of March in 2013. The interviewee worked as the chief finance officer.

The e-mail exchange with company E happened on the third of April in 2013. The interviewee worked as the chief executive officer.