Turun yliopisto
University of Turku

SUCCESS FACTORS OF COMMERCIAL OPEN SOURCE SOFTWARE PROJECTS

Master´s Thesis
in International Business

Author:
Erika Loimukallio

Supervisors:
Ph.D. Peter Zettinig
M.Sc. Mari Ketolainen

31.8.2012
Turku

Turun kauppakorkeakoulu • Turku School of Economics

Table of contents

List of figures

List of tables

List of abbreviations

BSD	Berkeley Software Distribution
CSF	Critical Success Factor
GPL	Gnu General Public License
FLOSS	Free/Libre/Open Source Software
IPR	Intellectual Property Rights
KPI	Key Performance Indicator
KSF	Key Success Factor
LGPL	Gnu Lesser General Public License
OS	Open Source
OSS	Open Source Software
R&D	Research and Development
VC	Venture Capital

1 INTRODUCTION

1.1 Background

> *The gap between what can be imagined and what can be accomplished has never been smaller. (Hamel 2000, 10)*

In the past decades the business world has experienced rapid changes in all areas affecting how business is conducted. The major phenomena and topic of discussion has been globalization and its impacts. The process of globalization has led to many positive effects to the world economy as the economies and market players can reach across their national boarders more easily and benefit from larger resource base as well as from new opportunities. (cf. Albaum, Strandskov & Duerr 2002)

Technological development is not only a factor facilitating globalization but also an outcome, as global integration at all levels (economic, social, technological, political and cultural) enhances knowledge transfer and diffusion of technologies. (cf. Albaum et al. 2002) The business cycle has become faster in most aspects, and a demand for tools in managing business operations has become undeniable. The managers are facing times of constant changes in the market structures, fierce competition and customer intelligence, which all demand close monitoring of global markets. Sustainable development of own products and business models are needed in order to be ahead of the competitors, and therefore the winners must follow the infamous guideline: Innovate or die! (Hamel 2000, 1-29)

Innovation management has become a notable factor in today’s business world. Not only has innovation become a key determinant of success, but also the way innovations are managed is shifting towards a new approach: open innovation. (cf. Chesbrough 2003a, 2003b). Open innovation is

> *..a paradigm that assumes that firms can and should use external ideas, and internal and external paths to market, as the firms look to advance their technology. (Chesbrough 2003b, xxiv)*

This broad definition by Henry W. Chesbrough, the prominent author of open innovation concept, can be extended to include all kinds of initiatives from joint ventures to a reach outside of the company for help in solving a particular problem (Hagel & Brown 2006). The software industry is one example of such initiative and has already taken the leader position in implementing and introducing the open innovation approach, through the development of open source software (OSS).

Open source software industry is based on freedom to access, modify and redistribute source code of any software under open source software license (see Open Source Initiative). In comparison to the traditional software industry, the business models in the open source software industry are very different, and thus the logic of income generation as well (see e.g. Koenig 2005; Lytras 2006; Daffara 2007). A completely different approach to business and managing models lets assume that the strategies for successfully making business in open source software industry also differs from the strategies in the traditional software industry (Helander 2007).

Software with freely available source code, which is one implementation of open innovation, has previously been mainly the interest of software developers. Introduction of the concept of open innovation and its cost- and resource-effectiveness, along with the success of open source software (OSS), raised the interest and recognition of business people to seek the potential benefits of such software. Traditionally there has been a wide gap between the roles of the business people and the IT people, and therefore the adoption of truly turning OSS into a core business of an organization has been slow due to the lack of a common perspective between the two viewpoints. During the last decades the whole software industry has been gradually forced to recognize the inevitability of OSS as the future software development model, and start changing their attitudes towards code sharing. Basically the giants in the industry are not able to make a full turn overnight, and trying to keep their existing share of the market and dominant design, they are strongly resisting the change. Instead, there are a number of small and start-up companies that have realized the huge potential of OSS, and have innovated a way to make business out of seemingly non-profit product, causing the shift from the brain-in-house development model to a service model. (Chesbrough 2003b; Helander 2007) For the reasons that the shift has happened very recently and abruptively, and that the innovators are the small players in the industry, the research of OSS from the business perspective is quite scarce, especially privately financed, although there is an increasing interest.

1.2 Research gap and research objectives

First of all open source is such a new way of making projects that the area has not been researched much. It has only been in the recent ten years that software business people have been taking any real interest in what is happening in the OS world, and only a couple of years that any research has been conducted specifically considering the managerial issues or success factors of commercial OS projects. OSS companies are fairly young – most of them founded after year 2000 (Seppä 2006a) – which means that the income generated by OSS companies has not been interesting until the very recent years. Secondly, most business related studies are concentrating on the different strategies the different OSS users take, mainly the public sector, because the buyers are more new to the subject and obviously in more need of guidance and managerial strategies.

OS is a multi-dimensional phenomenon that touches all aspects of the society: cultural, economical, social, political and technological. Eventually OS as any other field will be research in every aspect, but in the meanwhile the most demanded research areas go first. The stage has finally come where the commercial side of OS development has become the topic of many research and gained the focus of academics. For many OSS business model is an enigma because of the obvious dilemma of making money with something that is free, but taking a closer look reveals that there is a serious commercial touch in OSS, and the next step is to build strategies from managerial aspect. The search for success factors could not only benefit the managers of OSS firms, but possibly also the proprietary software firms.

The aim of the research is to find the key factors for successfully making business out of commercial open source software development. The sub-objectives of the study are:

- to find the factors influencing open source projects
- to find the relation between those factors
- to find out why certain factors explain more of the success than other factors

1.3 Structure of the study

The study aims to find out what are the critical factors affecting success in commercial open source business projects. The main parts of the study are: defining the key concepts, literature review of previous research, and the empirical research results.

First part of chapter two explains the concept of open innovation and its development, benefits and challenges, and moreover the concept of open source as the prominent example of the phenomenon. First the study examines the change in innovation management from closed innovation paradigm to open innovation paradigm. The shift explains the different deficiencies the closed innovation model possesses by the new course the whole society has taken, and therefore also clarifies the effectiveness of the open innovation model. This study mainly concentrates on those industries that are according to literature forced to shift to the open innovation approach, therefore assuming the shift inevitable. Next the study examines open innovation more closely, and explains its features, benefits and challenges. Because of the assumed inevitability of the shift from the closed innovation paradigm to open innovation paradigm caused by the evolvement on the economic, social, political, cultural and technological course of development on global scale, the study discusses the determinants and proposed benefits that drive companies towards open innovation model and the challenges of managing the open innovation, rather than advantages or disadvantages of the model.

The second part of chapter two focuses on the concept of open source (OS), and specifically on open source software (OSS) and the open source software industry. OSS is probably the most known and most deeply implemented open innovation project since the concept was developed, and therefore understanding the open innovation model is crucial. In this chapter the concept of OSS is first explained by definitions and then by comparison to proprietary software industry. The study presents a quick view on history of OSS and explains the role and nature of licenses, which are the main differentiator between OS and proprietary software.

Chapter three is the main chapter of the literature review. Business models are very closely related to success, and they are also different in OSS field from proprietary software field, and so the different business models are looked at from separate theoretical viewpoints. First the study looks at business model theories at a general level. Next the different software business models are reviewed, and last the business models specific to OSS field. The OSS business models are presented as a table summarizing the different viewpoints, and finally the typical OSS company is presented.

The second part of chapter three comprises of the key factors in the research: critical success factors. It is important to understand what CSF are, so the first part defines the concept, explains how to define CSF and finally what are the benefits of finding the CSF for an organization. The theoretical framework used in this study derives from the last section in chapter three; Critical success factors in open source software industry. There are two axes taken into account in this research: industry and innovation. On the second axis, it is clear that open innovation is the parent level of OS. In regards of the industry, software industry is both the parent industry and the opposite for OSS industry. The CSF are found in all three levels based on literature, and summarized as a list of CSF in three separate tables.

Synthesis summarizes the theory of chapter two and three. In the chapter is presented a synthesis of the CSF found in literature. First the theoretical concept definitions are summarized. The second part of the literature review is summarized as a framework in a form of a table that has been put together combining of all three sets of CSF in the previous chapter.

Methodology chapter describes the research methodology. It explains which are the different methods for research, and what is the background for choosing the particular method for this study. Next is examined how the data collection was made, how and why the interviewees were chosen and how the empirical data was collected. Last the trustworthiness of the study is being evaluated.

Results chapter presents the results of the study; synthesis of the findings in the theoretical background combined with the empirical research. The results chapter is constructed based on the framework presented in the synthesis, and presented in a form of the different sections of the synthesis, where each of the factors is evaluated in the light of the interviews.

Next the conclusions are made based on the theoretical framework and the results of the study. The conclusion chapter also discusses the managerial implications of the study, which are important so that the reader can grasp the essential idea of how the study can benefit the managers in their work. Suggestions for future research are discussed at the end of the chapter.

Finally a summary has been made of the whole research paper. The summary follows the structure of the paper and also presents the findings in brief.

2 OPEN SOURCE SOFTWARE

2.1 The shift from closed innovation to open innovation

Traditionally innovations have been produced by large companies who can invest vast amounts of resources in R&D. R&D has been seen as a strategic asset, which gives the company a competitive edge and even serves as an entry barrier to new entrants in the markets. Basically this has meant that only large companies with significant resources could be successful in R&D intensive markets. (Chesbrough 2004)

In the closed innovation model the focus is on the control. The company sees itself as a lone player against all other competitors in a hostile environment where all hard work has to be done and retained by the company itself. The innovation process from generating the idea to bringing it to the markets, financing it and holding the intellectual property rights is all conducted within the company. The implicit rules by Chesbrough (2003b, xx) explain the logic of the closed innovation paradigm:

- Hiring the best people will assure the smartest people in the industry for the company.
- In order to bring new products and services to the market they must be discovered and developed by the company itself.
- The first one to discover a new product or service will be the first one in the market.
- The company who can take the innovation to the market first, usually wins.
- If a company leads the industry in R&D investments, it will discover the best and most ideas and will be the leader in the market.
- Intellectual property should be controlled so that the competitors cannot profit from them.

The logic above can create a ‘virtuous cycle’ in closed innovation model, illustrated in Figure 1.

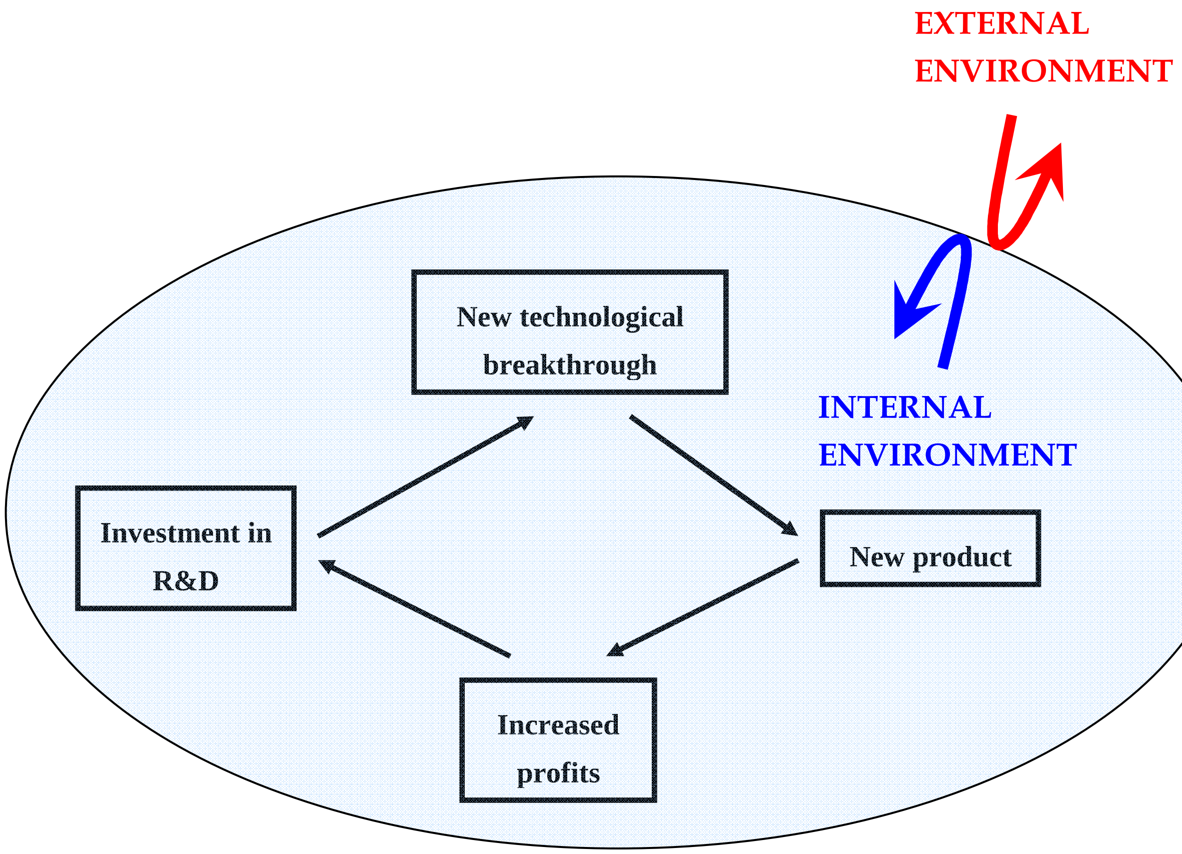

Figure 1 The ’virtuous cycle’ in closed innovation model (Chesbrough 2003b)

In the ‘virtuous cycle’ expressed in the figure the whole innovation process revolves around the company and neglects the external environment completely. It basically refers to a concept where a company is investing first in internal R&D function, then using the knowledge gained in its own operations to offer a new product or service in the market, consequently making more profit and investing that profit into more R&D. So the cycle goes on and on, without any inside-out or outside-in flows of assets. (Chesbrough 2003b)

Recently there has been growing awareness that the traditional model of innovation management does not work very effectively any longer for maximizing the return of R&D investment (Aarts 2005, 67-68). It appears that many large companies that had engaged huge amounts in long-term research programs found out that there is a remarkable competition coming from smaller new companies, with just a fraction of R&D investment compared to theirs. Also, many of the projects invested by those large companies turned out be of no use to them. Later on they discovered the same project being implemented profitably by another company. Chesbrough (2004) calls this change the paradigm shift, where the old paradigm is the closed innovation paradigm and the new one the open innovation paradigm. Whereas the closed innovation is focusing on maintaining the control and self-reliance, the open innovation encourages exploiting external resources, and opportunities in diffusion of own technologies.

Open innovation is a completely new playground. The old rules do not apply, and the companies must find the strategy that fits in the particular situation inside the turbulence of the paradigm shift. (Hamel 2000)

The former IBM research director James McGroddy describes closed innovation as playing chess, and open innovation as playing poker;

> *..you know the pieces, you know what they can and cannot do. You know what your competition is going to do, and you know what your customer needs from you in order to win the game. You can think out many moves in advance, and in fact, you have to, if you're going to win.*
>
> *In a new market you have to plan your technology entirely differently. You're not playing chess anymore, now you're playing poker. You don't know all the information in advance. Instead, you have to decide whether to spend additional money to stay in the game to see the next card. (Chesbrough 2003b, 13-14)*

According to Chesbrough (2003b, 13-14), a company that is practically doing everything right in its R&D management, but still fail, might simply be playing the game the wrong way. Succeeding in own R&D functions is not enough anymore when the yields are not sought outside the internal environment – it just means that the company is good at playing chess but lousy in playing poker.

Chesbrough (2003b) identifies four fundamental factors that have affected the obsolescence of the closed innovation paradigm. Those factors comprise of the availability and mobility of skilled workers, the venture capital market (VC), external options for unused ideas and the increasing capability of external suppliers.

The growth in the number of highly educated people and the increase in the mobility of such workers have led to availability of educated and skilled workforce. Companies have found it beneficial hiring workers from companies with valuable knowledge by offering more tempting career opportunities and incentives. That way the companies can exploit the research carried out by competitors or other market players and also exploit the knowledge and experience of a worker "trained" by another company. Diffusion of knowledge is therefore eroding the control a company has over its own research results and over explicit as well as tacit knowledge. (Chesbrough 2003b)

The venture capital markets have expanded enormously since the 1980's. Earlier the companies were not too worried by talented workers leaving the company, as they did not feel a threat by anyone exploiting the knowledge through new start-ups as the finance was simply not available. Since the emergence of VC, the employees are more tempted to engage in a small start-up business, as there are greater possibilities for success. As a result of shortening product life cycles and customer intelligence in addition to VC and mobility of employees, the companies with R&D investments can not wait anymore to be ready to use the research result sitting in the shelf. If they choose to wait, either competitors get to the markets first, or an employee starts a venture of their own. (Chesbrough 2003b)

Earlier many companies with extended R&D programs suffered from the inability of external suppliers to cover part of the value chain. Now, as the knowledge flows more freely, it is not beneficial for those technologically advanced companies to produce the whole range of value-adding chain internally. Naturally they benefit from supplier competencies in efficiencies in time, scale and costs, but the disadvantage is that so do the competitors. If buffer inventories of ideas waiting for further development or better timing are not used at the exact moment, it is far more likely that the research results can leak out of the company. The capabilities of managing the value chain are not any more the sole property of the R&D intensive companies. (Chesbrough 2003b)

2.1.1 Open innovation

The global changes in environments affecting companies have caused the 'shift of paradigms' effect, which means moving from closed innovation approach towards the open innovation model. The key idea behind open innovation is the effective way of accessing valuable knowledge, not only by exploiting the external innovations, but also by turning the unused R&D projects beneficial to the company. As discussed in the above, closed innovation model has presented many shortcomings in today's global environment in terms of knowledge acquirement and control. Open innovation is a model towards which most companies are forced to shift if they choose to win. (Chesbrough 2004)

Chesbrough's assumption is that companies must shift the paradigms eventually, but moreover, try to do it sooner and more effectively than the competitors in order to win. First the companies must identify their own position in regards of their innovation management and strategies compared to the industry as a whole. Then they must implement the shift to achieve more effective utilization of internal and external resources in a manner most suitable to the company. (cf. Hamel 2000) This chapter discusses the open innovation model and the management requirements in more detail.

In the shift towards open innovation the ‘virtuous cycle’ (Figure 1) is being disrupted, mainly for the erosion factors explained in the previous chapter. The new cycle demonstrated in Figure 2 presents the situation where the virtuous cycle has been broken and the closed innovation model has seized working or being the most effective innovation process. Once the change starts to affect the companies should become aware of the new threat which cannot be overcome without a radical change in their innovation management strategies. Unfortunately not all companies become aware of the need for shift of paradigms until the threat has become the reality. (Chesbrough 2003b, xxiii; Hamel 2000, 1-58).

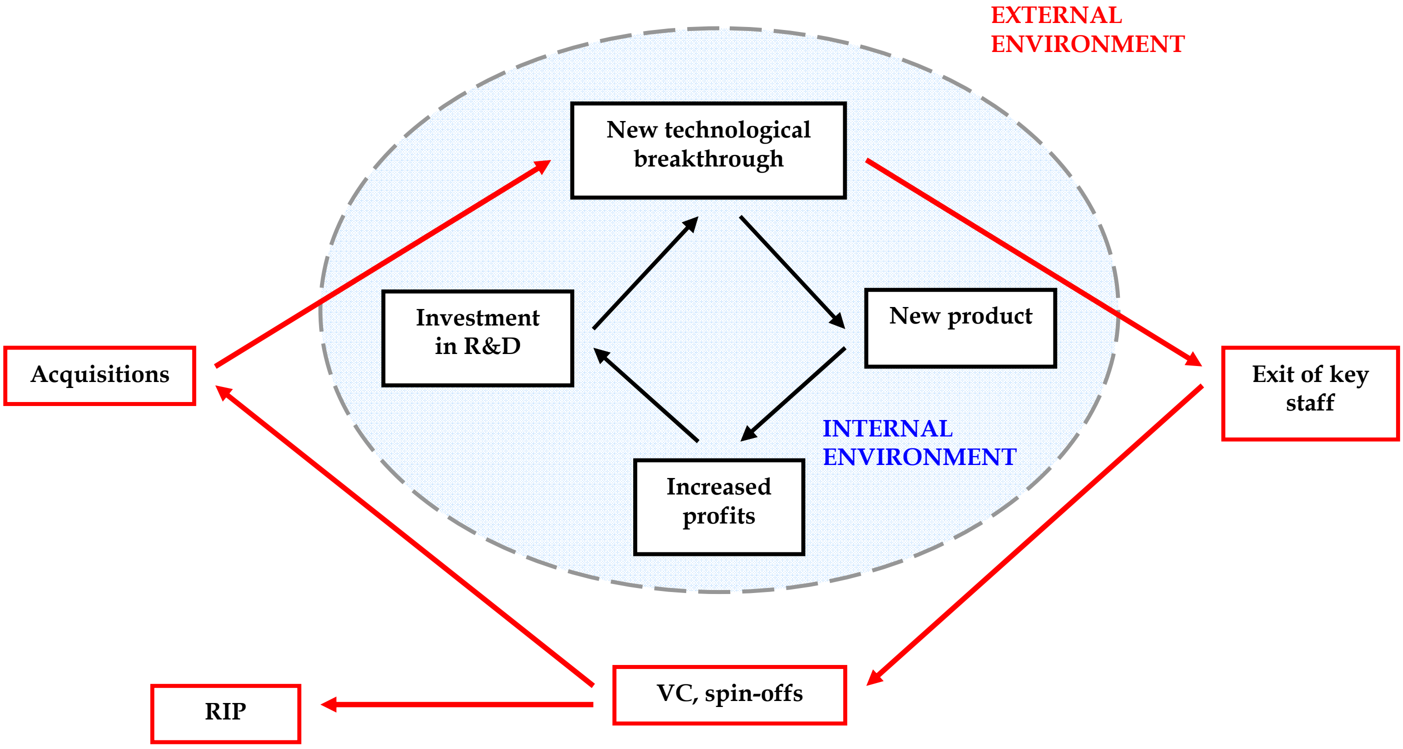

Figure 2 Disrupted ’virtuous cycle’ (Chesbrough 2003b)

Chesbrough (2003b) suggests that in such situation presented in Figure 2 the companies have only one direction to take in order to achieve the win-situation again. They must start looking outwards and explore external resources as well as ways to exploit their own R&D achievements on external markets. First of all, innovations are not all useful for the companies themselves, and secondly, sharing innovations and avoid overlapping R&D expenditure is not resource-effective only on global scale generally, but also inside the companies themselves.

There are three different processes in the open innovation model for resource efficiencies that separates the model from the closed innovation paradigm (Gassmann & Enkel 2004, 6-13).

- Inside-out process
  - gaining profits by transferring internal knowledge to external players
  - selling IP, licensing agreements
- Outside-in process
  - acquiring value-adding knowledge for enhancing the company's innovative processes and development through integration with external players
  - integrating vertical resources
- Coupled process
  - creating strategic alliances and partnership with external players for both inside-out and outside-in knowledge flows

The study found that some companies do not choose only one of these three archetypes, but in order to gain most efficiency might use one as the main strategy and integrate some elements of one or two of the others. The company has its locus of innovation within, but can combine it with a certain degree of knowledge and commercial exploitation, which leads to a combination of different processes of two-way innovation flow. (Gassmann et al. 2004, 6-13)

As discussed above and in previous chapter, there is certain inevitability due to global changes, although in some industries the environment demanding the change is not as strong as in others, e.g. the nuclear power industry where the innovation lies in so few companies that spill-overs or lack of resources are not an issue, or in industries where the innovations are highly confidential in nature (e.g. national intelligence, armed forced). (cf. Chesbrough 2003b) Moreover, not all companies in industries where the shift is already initiated acknowledge the situation to extent that lagging behind means lost innovation success. The phenomena affects most crucially the high-tech industries, where the diffusion of technologies has been widely experienced phenomena, and sticking to the closed innovation model especially when the competitors have shifted lead to failure in most cases. Thus it can be concluded that the pressure to shift comes from outside of the company, from its environment.

Some companies do find the shift more beneficial by themselves, in which case the triggering force comes within the company. The company might simply see the benefits of the open innovation model through a first-hand experience or by an opportunity that arises. More collaborative and open environment can be achieved by the realization that a largest possible amount of the brightest people translates into more effective idea creation, which can be turns into innovations (Aarts 2005, 68). Rigby and Zook (2002, 2) call it the garage effect when a company realizes a group of people can come up with an innovation as good – or even better – than their own R&D team can.

There are also some factors that might trigger – or force - the company to change its innovation strategies in shorter notice compared to forced change by industry evolvement, e.g. in an *unpredictable situation* (Rigby & Zook 2002, 2), as happened with Pitney Bowes, the world's largest manufacturer of mailing systems when the envelopes with anthrax spread infection through the U.S. postal system a few years ago. The company had to find the solution basically overnight, as the threat was immediate to the whole society, and therefore the company could not develop a solution with its own internal R&D resources. The only solution was to look for external possibilities generated by greater number of engineers and wider knowledge base.

2.1.2 Features, benefits and challenges of open innovation

The shift to the open innovation model cannot be regarded as an easy process without problems and challenges. It is important to take into account the nature of the new model with its challenges and possible problems and outcomes, and develop a strategy for managing the new model. The major challenge in open innovation approach is managing the change of the whole business model the new paradigm proposes (Hamel 2000). Constant evaluation and monitoring of the innovation processes internally and externally are essential as the R&D process is affected by uncertainties deriving from external influences. Internal interaction and coordination of different functions within the company have to be managed carefully in order to exploit the new innovation management model to gain maximized yield from the projects. (Chesbrough 2004)

West and Gallagher (2004, 3-5) found three main challenges a company is facing in the open innovation paradigm: maximization of the return on in-house innovations, incorporation of external and internal knowledge and motivating external innovation. Chesbrough (2003b) discusses four aspects – knowledge integration, venture capital exploitation, profiting from IPR and internal innovation – that call for a special attention of the managers of the companies that operate within the open innovation model. In the following the main challenges are discussed from a practical point of view.

The model of open innovation changes the role of research function. Earlier the researchers had the knowledge base mostly generated internally and they constructed new results on top of those. In open innovation model, the researchers have to be able to build the result from several sources, both external and internal. An example of the new role of R&D function is the leading pharmaceutical company in terms of internal R&D function Merck, who states in their annual report that as the research in the industry is too complex for one company to handle on their own, it reaches out to universities and other external R&D resources to create ‘virtual laboratories’ to gather maximized value adding knowledge base. (Chesbrough 2003b, 51-53) The challenge for companies is to reward the acquirement and fusion of those technologies the same way as if they were produced internally. The researchers’ new role requires a different working approach, where the achievements of one do not comprise only of the generation of breakthroughs, but combining them successfully. (West & Gallagher 2004) As the knowledge is diffused externally, and the internal researchers might not be willing to let their “babies” go outside the company, and might decide to flee with the innovation. The challenge is to maintain a positive environment among the R&D function, and foresee and manage the possible problem situations. (Chesbrough 2003b, 57)

Using the open innovation approach in managing innovations does not mean that internal R&D is not important, in fact, successful exploitation of resources deriving from the open innovation model requires well-functioning internal R&D ability to acquire and define the useful knowledge and incorporate it to internal functions. Relying only on external knowledge will most likely lead to failure, because different companies utilize the generated and incorporated knowledge in different manners, and use different combinations of external knowledge. Therefore, the use of external knowledge requires internal resources to adjust the knowledge to the business model of the company. (Chesbrough 2003b, 58-62; West et al. 2004, 4)

In closed innovation the venture capitalists (VC) are seen as threats, but in open innovation they can be regarded useful partners in regards of exploring market reactions, externalizing own knowledge and testing new innovations. The key challenge is to manage the relationship with VC markets to gain from it the best possible way. It is often difficult to foresee the different courses a new venture might take, and therefore the right tactics and strategic approach are required from the innovation management of the company. (Chesbrough 2003b, 54)

Licensing or selling the IPR are the new ways of profiting from R&D projects sitting in the shelf. Waiting is not the best way to force the full benefits from own research, due to the facts that the business cycle has accelerated its speed and the competition will not wait. Selling the innovation to an external party might also benefit the company through development of the industry sector generally, or through complementing the company’s own products, even though the innovator company has no direct use of its R&D result. (West et al. 2004; Viskari 2006; West, Vanhaverbeke & Chesbrough 2006)

As discussed earlier, the new role of the R&D function within the company is to fuse the acquired external knowledge in the most effective way in regard of the company’s technology and knowledge base. On the other hand the incorporation of such knowledge must also fit the organizational goals and strategies as well as the sales methods and external market factors affecting the company. Therefore the major challenge the company is facing is organizing its whole business approach in such way that all functions within the company interact (and even compete) to achieve the best possible outcome in creating a new innovation that fits the strategies of the company in all aspects. (Hamel 2000, 59-144)

2.2 Open source

2.2.1 Introduction to open source

Open source is one phenomena and embodiment of open innovation. Open source (OS) refers to permitting access to the "source" of the product, i.e. allows anyone the access to the design of the product and the possibility to modify the original design for any purposes. Open source projects can be anything, not only software related, that involves several instances contributing knowledge, information or other intangible assets to a community for sharing and modification for any use. (cf. Project Open Source / Open Access, Open Source Initiative). Figure 3 is collected from several sources for clarifying the relation between the different concepts (cf. Raymond 2000, Joutsen 2007, Stallman 2010, Open Source Initiative).

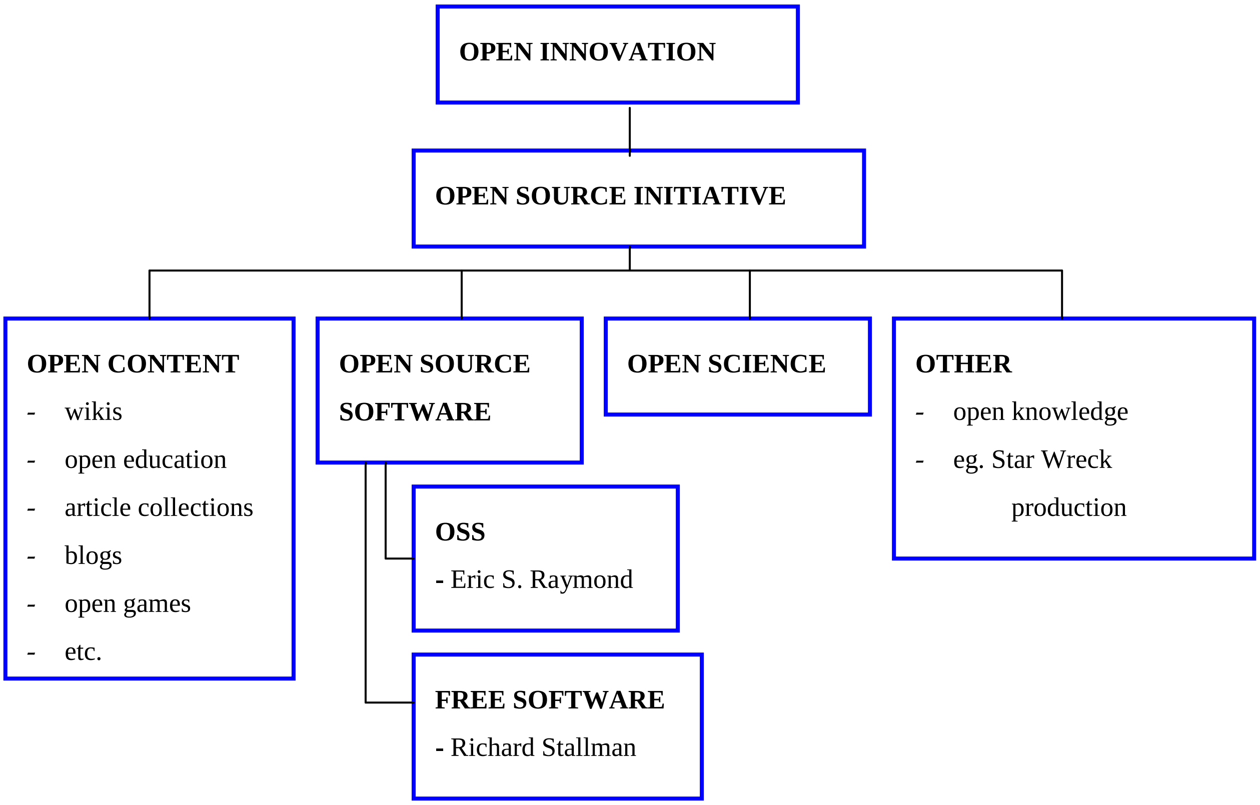

Figure 3 Concepts of open innovation (cf. Raymond 2000, Joutsen 2007, Stallman 2010, Open Source Initiative)

Another branch in addition to open source software is Open Content meaning content that is open for anyone to gain from and contribute to. Open Content includes wikis, open education, open article collections, blogs, etc. (Wolf 2007). Open source initiatives also include such projects where the members of the community contribute their knowledge or work to a common goal, e.g. the movies Star Wreck and Iron Sky were both created by several community members contributing different elements for the movie from special effects to graphics (Joutsen 2007, Västilä 2010). Open science means sharing scientific data, so that the methodology and collection of data is transparent, the data is publicly accessible and reusable, and scientific collaboration is wide with web-based tools (Open Science Resources).

2.2.2 Introduction to open source software

The term open source is most commonly associated with the software industry. Open source software (OSS), therefore, refers to any software where the source code is not confidential, but instead available for anyone to use or to modify freely (cf. Open Source Initiative, Free Software Foundation). The most known representations of OSS are probably Linux and MySQL. Another example of OSS known to most ordinary computer users in every day usage is OpenOffice.org.

Richard Stallman talks about free software, but points out that the term ‘free’ in OSS means the freedom in the sense of liberty, not in the sense of price (Stallman 2010). Later Eric S. Raymond acknowledged the misguiding form of the term ‘free software’ and came up with the term ‘open source’ which Stallman referred to as “impure” (Raymond 1998). However, the terms hold very little difference what comes to the actual meaning of OSS. Basically ‘free software’ is a bit stricter than ‘open source’ (Stallman 2010).

The official definition by Open Source Initiative of open source software determines software being open source when it complies with the following criteria (Open Source Initiative):

- Free redistribution of the software must be allowed
- The source code must be available
- Redistribution and evolution of modifications to the software must be allowed
- The integrity of the original author of the software can be protected
- There must be no discrimination against any persons or groups
- The use of the software must be allowed in any field of endeavor
- The original license applies to all redistributed software versions
- License must not specific to a product
- License must not restrict the use of other software
- License must not be restricted to the use of any specific technology

OSS is often confused with freeware or shareware, and therefore it is essential to explain the distinction between those three concepts. Freeware refers to a software that is available free of charge, but not necessarily available for free use for any purpose, with freedom to adapt or modify the software, freedom of redistribution or freedom to improve the source code. Freeware for commercial use can be distributed free of charge for certain use, but with cost for extra features or for certain groups, e.g. private people vs. companies. Shareware refers to software that is given free of charge for a limited time period for use only, but costs after the period ends. (Saastamoinen 2006) Therefore, the concept of OSS is not a freeware or shareware, but relatively strictly defined in order to maintain the original purpose and idea behind it and avoid consequent discrimination and misuse. (Open Source Initiative).

To explain further the idea of OSS, in the table 1 are shown the basic differences of an OSS and proprietary software projects (Lytras 2006, 4). Essential about OSS is that OSS development is carried out by a community, where each member contributes to the source code according to their motivation and capabilities. Usually the community members are private individuals, but also some companies encourage their employees to contribute to an OSS project if there is a direct utility to them. The developed software, regardless of who manages the project, is available for use, modification or redistribution not just to the community members but to anyone, which is the main idea behind the OSS concept. (Hars & Ou 2001; Lakhani & Hippel 2002) OSS development is usually managed by leaders with proven competence and the community members participate for reasons deriving from their own software needs and other motivation apart from financial gains, which means that the projects can produce as high quality software as in proprietary software projects. The OSS development is carried out in a less strict manner than in proprietary software development, but with standardized tools and open code which reduces the need to “reinventing the wheel” in every project. (Lytras 2006, 4) Peer review, i.e. in OSS world the different developers and users give feedback on each others’ work, implicates a higher degree of reliability in OSS products than in proprietary products. Total cost of ownership indicates directly that the OSS is cheaper for the provider because the original software is free of charge, which facilitates lower up-front costs. The main difference is the software risks, which are reduced in OSS compared to proprietary software, because with OSS the buyer is not locked into a specific vendor or its services. (Raymond 1999)

Table 1 Comparison of OSS and proprietary software projects (Lytras 2006)

	OSS projects	**Proprietary software projects**
Release planning process	Own ideals about quality and features	Time-to-delivery pressure
Quality assurance	Developers do not have write access to the repository	Developers do have write access to the repository
Leadership	Leaders are required to have proven competence via previous contributions	Project leadership is a hierarchical level where people are promoted by other than technical criteria
Tools and standards	Use of standardized tool chains (not modern modelling tools techniques)	Each project may opt for another and technology leading to different set of tools (code re-use between projects is limited)
Motivation of developers	Desire to learn and establish new skills (fun-to-code)	Typical task assignment by hierarchical superiors or salary incentives (not as efficient)
Roles of members	Members assume roles according to personal interests	Tasks are assigned

The major differences in OSS software development projects compared to proprietary ones lie in the more flexible and relaxed atmosphere in the whole process. As the table shows, most of the differences derive from the development organization structure. In OSS the development is done by the community, which requires or enables – not necessarily in a good sense – a less hierarchical and less formal environment what comes to project management and leadership. The projects are more difficult to manage, because the communities are in a state of constant evolvement and in many cases the developers are not even known by the management, and incentives based on mostly other than money create challenges of motivation of the developers. On the other hand the project needs a tighter and also a different management style than in traditional software development project, but on the other hand it allows more flexibility. The fact that developers do not have write-access to the repository, which means that only the project management has the right to decide what code is accepted in the actual production version of the software, is non-evitable, because otherwise the project might turn into a chaos due to non-parallel code and no control over the software features only dependent on each developer’s own interests that might not match at all to the requirements of the users. In proprietary software development the responsibilities of the developers are well-defined and project goals clearly drawn, which enables giving the developers access to the code itself. (Lytras 2006) Locke argues that there is basically a common understanding that OSS is a better way of making software because of benefits like technical excellence, reduced costs and wider selection of vendors. (Locke 2004). As mentioned there are also some deficiencies including the possible complexity of the software due to the multitude of developers and great number of versions to be maintained due to the freedom to modify (Helander 2007).

Typically an OS project is a repository in web, where only the key people have write-access (meaning the right to create or edit the code), and where the developers have read-access (meaning they can see the code and download it for reuse). The project is often broken into small pieces or modules that are managed by module manager, who can control the specific module. The users can freely participate in any module developing they desire, but normally they only contribute to one. (Goldman & Richard 2005).

2.2.3 History of open source software

The history of OSS dates back to the mid 20th century, contrary to what the mass media might suggest. The sudden explosion of internet and computer literacy has led to increased awareness of companies and individuals on OSS and its benefits in the last decade. Although OSS has existed already for several decades the mass media has not been interested in the topic until it became essential in everyday life. (Saastamoinen 2006) The history of OSS can be divided into three different eras: the early 1960’s and early 1980’s, the early 1980’s to early 1990’s, the early 1990’s to today. In the first era software was developed mostly by academic organizations and central corporate research facilities. In those organizations the researchers commonly shared their achievements with other researchers. The most prominent example is the development of Unix operating system was done in cooperation in order to achieve a system that would run on different computer platforms, and a community was established to join the forces of all the Unix programmers. In the end of the first era the original developer AT&T began to enforce its rights to Unix after a long period of informal practice of sharing and developing the code. (Lerner & Tirole 2000) The litigation processes started by organizations wishing to set ground rules in these cooperative software development processes led to the second era. The most significant party was the Free Software Foundation, known as GNU project, founded by Richard Stallman (Helander 2007). They developed the GPL – GNU General Public License – and LGPL – GNU Lesser General Public License – which granted freedom of use of the software and also restricted closing of the software for further use (Stallman 2010). At this point the developing community also started taking shape. As anyone could make changes to the code of the original software, the lead of a project had to be taken over by someone. Thus was developed the current model where the code can be taken freely by anyone, but the project of the original software is managed by a leader or a group of leaders who decide which bits are to be forming the new version. (Lerner et al. 2000) In the third era the internet was already taking off, which led to increasing number of new contributors and new OSS projects. One good example is Linux. Linux was started to be developed about five years after Microsoft started developing Windows NT, and yet, Microsoft had spent millions of dollars and unaccountable man-hours in the development whereas Linux was achieved by cooperation of numerous contributors. The media was attracted by the sudden arise of an open source operating system and widely covered the topic. Linux also started the era of corporate world entering the OS world. Number of companies started selling Linux, like Suse and Red Hat, and doing “the impossible”: making money out of something that is free. (Bretthauer 2001)

2.2.4 Licenses

Even though the idea of OSS business is that the source is available with no restrictions, there still need to be licenses. The purpose of OSS licenses is to ensure the freedom of the source code. (Helander, Aaltonen, Mikkonen, Oksanen, Puhakka, Seppänen, Vadén & Vainio 2007) The freedom means that a license holder can use and distribute copies without having to pay any fee to the original code owner or explain the usage. This does not mean that the license holder cannot charge for the copies – modified or not – it distributes forward. Just as important it is to define terms in proprietary software licenses in order to protect the copyright, as important it is to protect not only the freedom but also the recognition of the initial owner of the code with OS licenses. (Rosen 2004) Earlier in this chapter the definition of open source defines the degree of freedom of OS licenses, but still, many different licenses are needed, because just as in proprietary software licenses, there are variations. The degree of freedom that is set in the definition of OS is loose and permits these variations. (Saastamoinen 2006) The definition sets the boundaries and rules that cannot be bent, but the definition is also seen a bit controversial and gives room to interpretation (Rosen 2004). The following table shows how the licenses change when moving from strictly proprietary software licenses to the most open licenses.

Table 2 Software license permissiveness (Saastamoinen 2006)

License type	Free distribution	Free usage	Free modification	Free source code
Proprietary licenses	-	-	-	-
Shareware	x	-	-	-
Freeware	x	x	-	-
Public domain	x	x	x	-
Open source	x	x	x	x

In the table are shown the different license types, the less permissive on top and the OS licenses in the last three rows. Free distribution means the software can be redistributed without having to pay license fee to the original owner. Free usage means the right to use the software for any purpose without having to consult the original owner. Free modification means that the software can be modified. Free source code means the code itself is available for any use, modification or redistribution.

The most known and common licenses are GPL, LGPL and BSD (Berkeley Software Distribution). In addition to those there are many others, of which several the most used and numerous that are used less frequently. (Saastamoinen 2006) In the following table is a comparison of five common licenses with the purpose of giving an idea of the content and purpose of different OS licenses.

Table 3 OS license comparison (Saastamoinen 2006; p. 58,72)

	Persistence	Heritance	Badge	Proprietary use	Release of source code	Release of binary code
GPL	x	x	x	x	x	-
LGPL	x	-	x	-	x	-
BSD	-	-	-	-	-	-
Apache	-	-	x	-	-	-
MIT	-	-	-	-	-	-
MPL	x	-	x	-	x	-

Persistence means that if the OS software is redistributed, it must be so under the exact same license, i.e. the license persists in all redistributions. Heritance means a bit stricter version of persistence; even if just one piece of the code is used in another software, the whole software must be distributed under that license. If the license is not persistent or inherited it facilitates linking and releasing the code with software using other license. Badge means that the original owner of the source code must be mentioned in the redistributions, i.e. having the logo of the original developer visible in the new release. Proprietary use means prohibiting the right to use the source code in software that is closed for commercial purposes. Release of source code and binary code mean whether the source code or/and the binary code (which means the code produced by license holder) has to be published when the OS licensed code is linked. (Rosen 2004, Saastamoinen 2006, Open Source Initiative)

3 BUSINESS MODELS AND CRITICAL SUCCESS FACTORS

3.1 Business models

Basically business model is a representation of how a company buys and sells goods and services and earns money (Osterwalder 2004). More precisely:

> *A business model is a conceptual tool containing a set of objects, concepts and their relationships with the objective to express the business logic of a specific firm. Therefore we must consider which concepts and relationships allow a simplified description and representation of what value is provided to customers, how this is done and with which financial consequences. (Osterwalder, Pigneur & Tucci 2005, p.5)*

Going deeper, there are several different views explaining the formulation of a business model. This work presents four different views that Seppänen and Mäkinen (2005) consider the most essential in literature. The first suggests that there are six different elements that are causally related: customers, competitors, offering, activities and organizations, resources, and supply of factor and production inputs. In addition they have included in the model a longitudinal component that covers the dynamic business processes such as cultural and other constrains that the managers have to cope with. (Hedman & Kalling 2003) The second one proposes that business model is based on value creation rather than revenue generation. The value creation is composed of four different dimensions: efficiency, novelty, lock-in and complementaries. (Amit & Zott 2001) The third also concentrates on value creation and distinguishes several actors in the value chain that are interrelational: market segment, actor, value interface, value offering, value port, value object, value exchange and value transaction. (Gordjin & Akkermans 2003) The fourth model suggests there are four essential questions to be asked: What? How? Who? and How much? The importance when answering the questions are: customer relationships, product innovation, infrastructure operations and financial aspects. (Pigneur 2004) What is common for all these four views is that they all suggest that there are several different aspects, moreover the value creation component, to consider when creating a business model.

3.1.1 Software business models

As this study is concentrating on open source software, a brief look is taken on proprietary software industry’s business models and their formulation. The main division in software business is made to services and products (cf. Hoch, Roeding, Purkert & Lindner 1999; Hyvönen 2003). Another way to make the division is to project based and product based business. Project based means software that is tailored in cooperation with client, i.e. customized software. Product based is off-the-shelf software which is developed first and then delivered as it is. (Tähtinen 2001)

Hoch, Roeding and Purkert (1999) present one way of describing the division of IT sector, which is represented in Figure 4 (Hoch et al. 1999, p. 27).

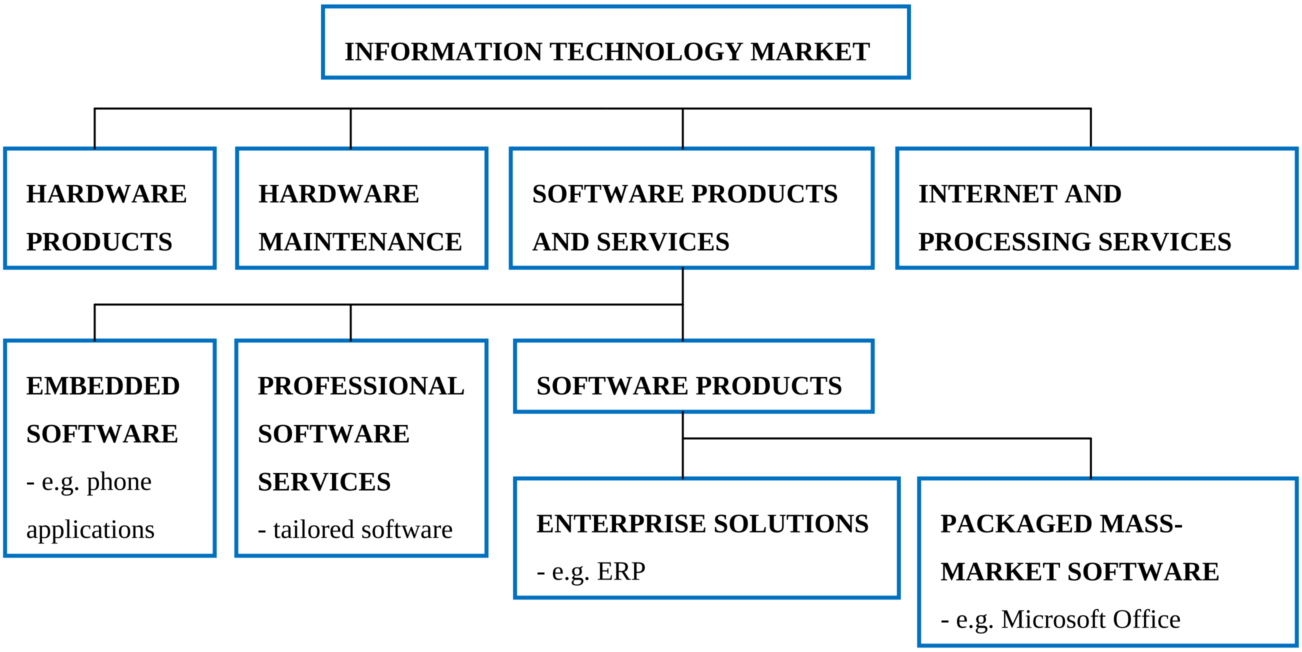
INFORMATION TECHNOLOGY MARKET
HARDWARE PRODUCTS
HARDWARE MAINTENANCE
SOFTWARE PRODUCTS AND SERVICES
INTERNET AND PROCESSING SERVICES
EMBEDDED SOFTWARE
- e.g. phone applications
PROFESSIONAL SOFTWARE SERVICES
- tailored software
SOFTWARE PRODUCTS
ENTERPRISE SOLUTIONS
- e.g. ERP
PACKAGED MASS-MARKET SOFTWARE
- e.g. Microsoft Office

Figure 4 Division of IT market business models (Hoch et al. 1999)

Considering the division of software products and services we find three main categories: software products, professional software services and embedded software. Software products are divided into mass-market software, which means products that are produced as a single product for all clients, e.g. Microsoft Office, and enterprise solutions, which means tailored software services based on mass-market software. Professional software services refer to the next level involvement of the customer, i.e. services where the product – software – is developed together with the customer. Embedded software is a software that is part of a device it is supplied with. Hyvönen suggests a similar model which is presented in Figure 5 (Hyvönen 2003, 3).

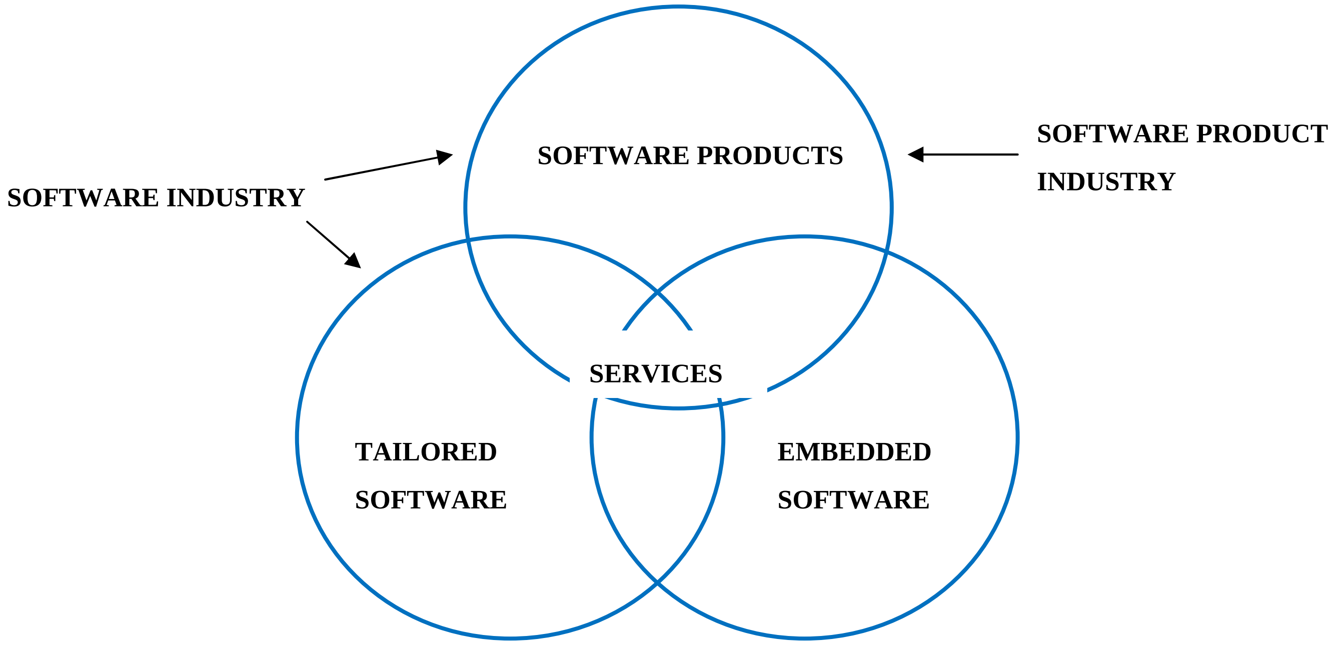

Figure 5 Division of IT business models (Hyvönen 2003, 3)

The model contains the same elements as the model of Hoch et al (1999), but places software services in the middle of the other components of software services and products, which indicates that services are nowadays an integral part of software business and is extended at all levels.

Seppänen (2010) divides software business clearly to products and services. He identifies different software business models through different contract models: project contracts, development contracts, hosting contracts, consulting contracts, licensing contracts and retailing contracts. Projects are services, where a complete customer project, e.g. integration project, is given to the company, who executes project for the customer. Development contracts refer to software development, i.e. a company wants a certain type of tailored software developed for them. Hosting means a business model, where existing software or a system is maintained by the company for the client on their server, possibly including support. Licensing means selling a software product that has been developed by the software company, e.g. Microsoft selling rights to use its Office software to a private consumer. Retailing contracts refer to another company selling products for the software developer. (Seppänen 2010)

3.1.2 Open source software business models

The emergence of OSS and the strengthening foothold it is gaining in all areas of the society is changing the structure of the software industry and forcing a shift from closed to open innovation model. The change is already happening and along with the initiated shift, the rest of the industry must face the reality and go along - or die. What makes OSS and its impact on the equivalent industry a very intriguing and unique example of the paradigm shift is the completeness and massiveness of the change it forces the industry to adopt. Although the paradigm shift, regardless of the sector undergoing it, can be characterized with inevitability, the shift usually involves changes and impacts on the innovation strategies. In software industry the paradigm shift affects the entire business model; the very basics of the income generation models and the strategies throughout all the operations of the organization. (Chesbrough 2003a; Benussi 2006).

Considering the fact that OSS has a free source code, which means that anyone in the industry can copy the same code and use it for commercial purposes, one may ask two questions:

- How is it possible to make money with OSS?
- Why does a company want to make business with OSS?

As explained in the first chapters on the shift from Closed to open innovation model, the paradigm shift changes the whole concept of the business. In OSS industry the shift does not only change the business strategies but also the business model from the income generating point of view (West et al. 2004).

Below is presented a table that shows different categorizations of open source software business models by four different authors. The rows in the table show all different business models named by the four authors, and the columns show which authors have included the business model in their categorization. The last column indicates which overlapping business models are combined.

Table 4 OSS business models (Daffara 2007, 5-6; Koenig 2005; Jing-Helles 2008; Rajala et al. 2007)

	Daffara	Koenig	Jing-Helles	Rajala	Overlapping
Subscription model		x	x		Subscription - hosting
Split OSS / proprietary	x		x		Split OSS / commercial products
Dual licensing	x	x	x		Dual licencing
Badgeware	x				Extension to other business models
Product specialists	x				Product specialists
Platform providers (integration)	x		x		Platform providers – general consultancy
General consulting	x	x		x	Platform providers – general consultancy
Embedded software		x	x	x	Embedded software
SAAS			x		Subscription - hosting
Patronage		x	x		Patronage
OSS event and publishing			x	x	Related services
Advertizing		x	x		Related services
Partnership			x	x	Combination of other models
Hosting		x			Subscription - hosting
Optimization		x			Patronage
Support				x	Subscription - hosting
Loss-leader				x	Patronage
Brand licensing				x	Combination of other models
Sell it, free it				x	Extension to other business models

Some of the models in the table above are overlapping to a certain degree, thus the models summary in this study will be a combination of the business models instead of a list of all of them. Constructing from Daffara's (2007, 5-6) research of more than 100 companies, Koenig's article (2005), Jing-Helles's presentation (2008) and Rajala's article (Rajala, Nissilä & Westerlund 2007) the main categories of adding value through OSS can be identified as follows:

- Subscription - hosting
    - In addition to the software itself, subscription includes services like hosting of the software, updates, support, new versions etc.
- Split OSS/commercial products
    - Software based on OSS but extended with copyrighted closed source code
    - Distributed under one license
    - Competes with the commercial software
    - In addition to revenues from commercial extension, benefits derive from large developer community
- Dual licensing
    - Software is distributed both under open source and commercial license
    - Part of the software is under free distribution and fee is charged on version with extra features or modifications
    - Benefits in improved awareness, free testing, free bug-fixing, etc.
- Badgeware
    - OSS with extended license that requires "visibility", e.g. a trademark in the user interface of the software
    - Benefits derive from larger developer community
- Product specialists;
    - Software developers that use pure OSS license, and instead of selling the license, they sell consultancy and other expertise services related
    - Service based on free technology
- Platform providers – general consultancy; Companies that provide selection, support, integration and other services related to different OS software to form a coherent platform for their clients, companies that offer consultation on selection process and integration projects
- Patronage; Giving the source of in-house developed software to an OS developer community for development and free use in order to sell well-integrated additional parts or tools commercially to users of that software
- Embedded software; embedded software means software linked to a certain hardware, e.g. mobile phone. The companies open the source code in order to

get the software for their hardware developed by different service providers, which services can then add value to their hardware.

- Hosted strategy; Building own commercially sold software on OSS platforms
- Related services;
  - Related services refer to e.g. advertizing, publications and open source event organizing
  - These are business models that are not directly related to the software, but indirectly

Even though, as discussed above, it is possible to make profitable business out of OSS, the possibly even more important question is why to do so especially because the prevailing 'closed source' model has proven to be profitable. First of all, by exploiting OSS a company can stay in control. The company can rely on itself when it comes to product modifications and software integration, which means that it can decide what it does and when it is done without having to consult a copyright holder. Often required and needed software is either not available, inadequate or unsatisfactory. Sometimes it is also more cost-effective building the software from a scratch rather than buying it. Open source software usually has a higher compliance with other software, which might be an important factor especially to a company operating in a monopolistic environment in terms of the providers. Independence from certain platforms and operating systems not only widen the selection of software and hardware providers for the company, but also facilitates establishing an industry standard of the open source software. (West et al. 2004; Koenig 2005; Wolf 2007)

Basically the idea behind using OSS for commercial purposes is to be effective, not just in costs but also in management and relations to external stakeholders. The effectiveness is also a goal of the non-OSS companies, but the difference lies in the exploitation of external resources according to the open innovation model which enables the OSS companies to achieve the effectiveness at a greater level. In addition to exploiting the external resources the OSS companies also contribute to the external society, in which cases the effectiveness derives from sales of related products (pooled R&D), generating demand for own products (patronage, see the list of business models above), the possibility to generate standards, generating goodwill, attracting improvements and complements that make own products attractive, sales of complementary products, demand due to external contribution for valuable components, etc. (West et al. 2004).

Seppä (2006a) in his research has found out that there are certainly differences in the characteristics between typical Finnish OS software companies and typical traditional software companies. The following table shows the differences (Seppä 2006a).

Table 5 Comparison of typical Finnish OS and traditional software firms (Seppä 2006a)

	OS software firms	**Traditional software firms**
Size – turnover EUR	300.000	700.000
Year of establishment	average 1998	average 1992
Personnel	average 25	average 190
Education – university degree	~85%	~85%
Customers type - comparison	no difference	no difference
Licenses – importance increases on scale 1-5	2,7	3,4
Patents – importance increases on scale 1-5	~1,6	~1,6

Seppä (2006a) found out that the size of OS firms on average is significantly smaller than that of proprietary software firms. This could be explained a little bit by the fact that OS firms are several years younger than proprietary software firms, most of them established in the 21st century (Seppä 2006a). Interesting fact is that the ratio between turnovers of OS and proprietary software firms (0,43) is substantially higher than the ratio between the average personnel of the same (0,13). Other differences lie in the importance of licenses (Seppä 2006a). In traditional software company the licenses play major importance, which is understandable, because the whole idea of the proprietary software industry is that there exists an owner of the code, and the licenses are the only way to secure the right to the product. In OS business the licenses are partly a moral issue, because their meaning is to secure the freedom and no infringement would normally affect the income of a company. (Seppä 2006b) Level of education is traditionally high in IT sector, and in both proprietary and OS software companies the level was very high; about 85% of the employees have a university degree. The customer types by different sectors of business did not show any significant differences between OS and proprietary software firms. (Seppä 2006a) Bonaccorsi, Rossi and Giannangeli (2004) found out in his research that the growth-rate measured in turnover was remarkably high in OS companies in after their launch; average of 121% (Bonaccorsi, Rossi & Giannangeli 2004, 12).

3.2 Critical success factors

The history of critical success factors (from here on CSF) begins already in the 1970's, mainly by John F. Rockart (Esteves 2004). There are many different definitions of CSF, but according to Esteves (2004) the most sited is John Rockart's definition from 1979:

> *Critical success factors thus are, for any business, the limited number of areas in which results, if they are satisfactory, will ensure successful competitive performance for the organization (Rockart 1979, p. 85)*

York P. Freund (1988) generalizes Rockart's ideas as the things that must be done in order for the company to be successful. Whereas Rockart focuses on the company itself, recently the research has taken a course towards including external factors to the CSF (cf. Jonker 2004). The CSF must also fulfill the following (Freund 1988):

- Important to the company's goals; being critical, the factor needs to be in line with the company's overall goals
- Measurability and controllability; it must be possible to be able to measure whether the factor in question has been achieved, and also to control how it can be achieved
- Being few in number; if there are too many, they can be called key success factors, and they are not detailed enough
- Obligatory in nature; if the factor is critical, it must be achieved at some level, otherwise the business is not running
- Industry-wise; the factors should apply all organizations with similar goals and strategies
- Hierarchical; some factors apply to more specific functional area, some to the whole organization

Very important evolution is the shift from company-based view to industry based view. The industry based view can be seen as a tool to define the relative competitive position of the company in the market. As Hofer and Schendel (1978) define the CSF:

> *Critical success factors are those variables that management can influence through it's decisions that can affect significantly the overall competitive positions of the various firms in the industry. These factors usually vary from industry to industry. (Hofer & Schendel 1978, p. 77)*

The main aspect of Hofer et al. (1978) is that the industry factors have a major impact on the success factors of the companies in that industry. Many authors have contributed in the defining of CSF and Jonker (2004, p. 61) suggests that among the many definitions – whether firm-specific or industry-specific – four common characteristics can be found:

- CSF are the sub-goals of the long-term goals
- CSF can be internally related or externally related firm-specific areas

- CSF are those limited factors that will ensure the competitive performance
- CSF are results that can be measured

The CSF should not be confused with competitive advantage, KPI (Key Performance Indicator), or KSF (Key Success Factor). The following table shows the difference (cf. Ellegaard & Grunert 1993; Freund 1998, Jonker 2004):

Table 6 Success indicator terminology comparison (cf. Ellegaard & Grunert 1993; Freund 1998, Jonker 2004)

	Critical/Important	**Company/Industry**
CSF	Critical	Company/Industry
KSF	Important	Company/Industry
KPI	Important	Company
Competitive advantage	Critical	Industry

The main differences come from importance to the company and whether the factor is company or industry related. As seen in the table CSF is quite close to KSF, and often the terms are used alternately (Ellegaard et al. 1993). The main difference is that CSF must be few in number. It is impossible to have many factors that are critical, so basically the CSF are the top KSF, which in turn are all the internal and external factors that make the company successful (Freund 1988; Ellegaard et al. 1993). Competitive advantage differs from CSF in the fact that the company may have all its CSF fulfilled, but might still fail to have the utmost competitive advantage in the market due to industry-wise factors including competitors positions, and typically only one or two companies in the niche can achieve competitive advantage. Competitive advantage refers to the position of the company in the market, not the internal strategies and tactics (Freund 1988; Jonker 2004). KPI is more of a set of tools to measure the outcome of implementation of the CSF, than the actual strategy. In other words, KPI is a tactical tool, not a strategy, but CSF is both a strategic and tactical tool (Ellegaard et al. 1993). Esteves (2004) defines strategy as which goals to pursue, and tactics as how to achieve these goals.

3.2.1 Identification of CSF

Esteves (2004) names main sources for CSF identification: case studies, interviews and research review. Basically there is not general measurement tool or set of procedures that would ensure CSF identification for each firm in each industry. Lynch (2000) has an approach where the identification is divided into three categories: environmental factors, industry factors and organizational factors. Environmental factors refer to factors at the political, socio-economical, legal and technological levels. Industry factors comprise of factors related to competition, market and geographic location. Organizational factors are those resources and competences specific to the firm. (Lynch 2000)

Practical internal CSF identification process involves a brainstorming session of the appropriate management personnel in the firm. First the mission and vision of the company should be reviewed and possibly clarified, since – as stated earlier – CSF are the sub-goals of mission and vision. Then the existing processes should be analyzed carefully in order to be able to formulate the problem areas. The next step is generating the CSF bearing in mind that the CSF should be important, necessary, obtainable, sufficient and critical. The CSF should periodically be reviewed, because market and the firm are in constant change, and thus the CSF are prone to change. (Liang 1999; Jonker 2004)

Esteves (2004) points out six different levels to which the CSF identification process should pay attention:

- Hierarchy vs. group of CSF
  - whether the CSF are formed hierarchical way, eg. industry, corporate or firm-level (Rockart 1979), or arranged as groups of roles of companies in industry (Esteves 2004)
- Temporary vs. ongoing CSF
  - Some CSF can be temporary in nature, i.e. prevail only a certain period of time in the project lifecycle, or ongoing which means the particular factor must exist indefinitely. All CSF can be defined as temporary, but not necessarily, and temporary CSF can be temporary for different time periods at different moments. (Khandewal & Ferguson 1999)
- Internal vs. external CSF
  - Internal and external mean the degree of which the management has control over the factor, i.e. the more external the factor is, the less the management can influence it (Arce & Flynn 1997). Rockart (1979) stresses that acknowledging the degree of externality is important when thinking of communication and information sources.
- Building vs. monitoring CSF

  - Building and monitoring CSF refers to the type of control the management uses over the CSF. Building can be seen as something that can be controlled and monitoring something uncontrollable that needs to be monitored. (Rockart & van Bullen 1986; Arce et al. 1997)
- Strategic vs. tactical CSF
  - There should always be a good mix of strategic and tactical CSF (Ward & Lincoln 1990). Strategic CSF mean critical goals that are to be achieved and tactical the means of achieving the goals (Esteves 2004)
- Perceived vs. actual CSF
  - It is impossible to determine the actual CSF (Dess & Robinson 1984), Trying to acknowledge the discrepancies between the perceived and actual ones, can help the management build more effective strategies (Ellegaard et al. 1993).

Tübke (2004) use the practical method in their work to identify CSF in certain industries respectively: they first identify the factors in the industry that can have an effect on business, the different characteristics of the firms in that industry, and finally compare the successfulness with the different characteristics and factors the firm possesses. Juga and Uotila (1993) separates between push-type and pull-type firms, which indicate the main strategy the firm can have in the industry. The discriminating components are then listed for both strategy types and evaluated.

3.2.2 CSF benefits

Rockart (1979, p. 87) discusses the benefits of defining the CSF for the company. The proper CSF helps the management to focus on right issues, acquire necessary data and resources, and develop good measures. This means that the CSF properly defined, the manager can focus on those issues that need the most attention by gathering the appropriate and necessary information and communication from the right resources, thus avoiding all misleading communication and resource-wasting. The manager can develop best-practices for the management work specific to the manager in question. (Rockart 1979)

It is important to remember that the CSF are in constant change, just as is the whole business environment – internal and external. The CSF can be industry-specific and even manager-specific, and therefore constant reviewing and the right focusing will enable the most effective management style. Having the CSF defined properly and regularly gives the management a tool to stay on top of the change what comes to resources and information. (Pinto & Prescott 1988; Rockart 1979; Esteves 2004) On the other hand, changing the management style according to evolving CSF always brings out resistance to change. (Liang 1999)

Managing with the focus on CSF will not ensure the success. Adjusting the management style and strategy definition to the CSF identification is a delicate process and requires the proper realization of the “big picture”. There are also other issues than those of CSF that need to be stressed, but with proper insight and initiative the management can use CSF as part of the puzzle of the way to success. (Liang 1999)

3.3 Critical success factors in open source software industry

As CSFs are specific to the industry, to the company and even to its different departments, there exists no indefinite theory or list of the CSF in an industry (Bruno & Leidecker 1984; Jonker 2004). However, there are many studies that have listed the possible factors in software industry and even those related to OSS industry.

This chapter is divided into three sections. The first part examines the previous research of the CSF in the whole software industry, which enables summarizing the main CSF in the software industry presented in the literature. The second part examines the previous literature in light of CSF in OSS industry, and summarizes those. The third part looks more closely at the OSS industry and draws conclusions from the aspect of risks and challenges that are critical for OSS industry specifically, and finally those factors are turned into possible CSF, presented as a summary.

3.3.1 Software industry CSF

Hoch et al. (1999) interviewed people from top 10 companies in both categories of software product business and professional software services in order to find out the success factors in those fields of business. Their findings are shown in the table below.

Table 7 Software industry CSF (Hoch et al. 1999)

	Factor definition	**Success factor**
Partnering	Amount of business partners to share the market with	- High number of business partners
Service strategy	Combine services with products vs. focus on one area	No unambiguous result
Marketing	Strong brand building vs. trust-based relationships	No unambiguous result
People management	Efficiency vs. long-term relationship	- Retain key people selectively - Hire according to cultural fit - Bringing new hires up to speed fast
Developing	Creativity and flexibility vs. disciplined execution	- Divide project into small sections easily monitored - Tighter project management - Generation of running versions of software more frequent - Discovery of more coding faults - Re-usage of source code

The study suggests that it is important to have many business partners. Partnering enables access to external competencies, which might be a requirement for the business. Also it helps bringing the product faster to the market. Third factor why partnering is encourage is that it enables bringing the products in the market faster and with greater volume. (Hoch et al. 1999) Here can be noticed, that these factors are very similar to the characteristics of open innovation explained in the first chapter.

Service strategy is an unambiguous issue, because it greatly depends on the sector of software business. Hoch et al. (1999) investigated two sectors - product and services – and found out that in product business speed is the key and high market penetration, while in services sector the key was continuity, long-term strategy and excellent reputation and customer relationships. Marketing approach also showed great differences between the sectors. Products are usually produced first and sold later, while services are first marketed and sold before fully created. Product marketing usually is mass marketing, but services marketing is based on relationships and selling is direct only. (Hoch et al. 1999)

People management has two sides: having great turnover of the staff or not. Having great mobility of the employees in the field translates to fresh ideas and more innovation. Also new leads might emerge and change in staff often facilitates change. However, it is important to retain the key people, especially those that hold the most knowledge and have good relationship with the customers. Hiring the right people considering the company culture is an important factor. The company atmosphere is important to the well-being of the staff and fitting to company culture also means that the employee can take forward and comply with the existing strategies. When the employee feels that the company is a fun place to work, he can be more effective in his/her work. It is also important to train the new employees fast for obvious reasons, that are often forgotten by the managers: the faster the employee adapts, the faster his/her work can become productive for the company. (Hoch et al. 1999)

What comes to software development projects, successful companies are able divide the project into smaller pieces since in all projects simplicity is the key to success because the project is then easier to manage and to bring to success. Project management has to be focused on the project and preferably not have any other duties. The tighter the management is the better the project can be carried out. Coding faults and bugs are discovered more frequently by successful companies, and those companies also use more often the same code they have already created, which increases the effectiveness of use of resources. Related to testing, those companies that were able to get new releases of the software in the market more often, were more successful. (Hoch et al. 1999)

A more project management type of approach is taken by Reel (1999), who distinguishes five different key success factors: starting on the right foot, maintaining momentum, tracking the process, making smart decisions and institutionalizing post-mortem analyses. He bases the software industry key success factors being different from those in other fields of business on the complexity of software development (Reel 1999). It is important to have the starting point correct. 70 % of the projects fail due to actions before the actual start of the project. The key issues in the prestart phase are understanding the customer needs, defining well and realistically the project scope and objectives, and hiring the right people. The objectives need to be realistic and well-defined in order to keep the team and the customers motivated. The outcome should solve the customer’s problem, and keeping track of that by communicating with the customer is very important; the project is of no use if the customer will not buy it. Building the right team is a task that should be devoted a special attention. Wrong and inappropriate team members can do more harm than good. Equally important is equipping the team with proper tools. (Field 1997) Keeping the momentum refers to having the state of perfect start carried out throughout the whole project (Reel 1999). There are always changes; project changes because the project is always evolving, technology changes, and business need changes (Field 1997). It is easy to build when the start has been done with the right foot, and fairly easy to keep up, but extremely difficult to rebuild. The critical factor here is to anticipate all changes to negative and turn them into positive. This can be done in three ways according to Reel (1999):

- Keep attrition low – not letting any knowledge walk out the door
- Quality – being sure the quality remains by setting the standard high form the beginning and demanding the same level in the course of the project
- Management of product, not people – managing more the product than the people, because the product is what will be sold

Finally, Reel (1999) says that many of the projects fail simply because of bad decisions and not learning from previous mistakes. The project manager is the key person, and if that person is not performing well, the project will not perform well. Also, there is definitely no need to repeat the mistakes made in other projects, there is always something to learn. (Reel 1999) Apparently lessons have been learned, since the project performance has improved significantly since 1994 (Nasir & Sahibuddin 2011).

Nasir at al. (2011) made a comprehensive literature review about CSF in software industry. They defined 26 different key factors affecting success in software development project:

- Clear requirements and specifications
- Clear objectives and goals
- Realistic schedule
- Effective project management skills/ methodologies (project manager)
- Support from top management
- User/client involvement
- Effective communication and feedback
- Realistic budget
- Skilled and sufficient staff
- Frozen requirement
- Familiarity with technology/ development methodology
- Proper planning
- Appropriate development processes/ methodologies (process)
- Up-to-date progress reporting
- Effective monitoring and control
- Adequate resources
- Good leadership
- Risk management
- Complexity, project size, duration, number of organizations involved
- Effective change and configuration management
- Supporting tools and good infrastructure

- Committed and motivated team
- Good quality management
- Clear assignment of roles and responsibilities
- Good performance by vendors/ contractors/ consultants
- End-user training provision

As can be noted, these 26 factors support the findings in the earlier literature review in this chapter. The different factors are broken in this study into more specific factors that are here called sub-factors, and this is done for clarification reasons. The following table summarizes the different aspects presented in this chapter:

Table 8 Critical success factors in software business

Factor	**Sub-factor**
Project plan	Clear requirements and specifications
	Clear objectives and goals
	Realistic time schedule
Project management	Tight management
	Tracking and reacting to changes in the environment
Product management	Dividing project to sections easier to manage
	Quality management
	Continuous and effective communication with customers
	Re-usage of code
	Proper and frequent testing
Resource management	Hiring the right people
	Keeping the right people
	Equipping the team with proper tools
Partnering	Multiple partners

The table shows that the above mentioned views of different authors can be summarized and categorized in five main categories: project plan, project management, product management, resource management and partnering. Project plan is further divided into three sub-factors that all strongly relate to the initial planning phase of the project; having clear and realistic goals and requirements. Project management is divided into two sub-factors that emphasize the role of the project manager as well as the agile nature of the development process. Product management is divided into five sub-factors that also relate to agile methods, and in addition emphasize quality and effectiveness. Resource management and its three sub-factors indicate the importance of the people involved and the effectiveness of the usage the resources. Partnering stresses the fact that multiple partners are key to successfully exploiting resources and methods.

3.3.2 Open source software industry CSF

Some, but not many research has been made on OSS development success factors (eg. Weber 2004; Seppänen 2006; Peng 2009). Most of them concentrate on knowledge creation through communities. It is important to notice that knowledge creation is associated with OS communities and community management and structure, which is only one area of the whole OSS development and OSS business processes. Therefore the success factors in the light of the community are not relevant if the community management itself is not a success factor in OSS business.

Communities and the relation between projects and the community members seem to have strong and powerful effect on project success – both technical and commercial. The embeddedness has a clearly positive effect on success, but those effects are so complex that the result is not unambiguous. (Grewal, Lilien & Mallapragada 2006) Singh, Tan and Mookerjee (2009) concentrate in their study in the community in OSS project and its success in the development, which they call network social capital. Their study found out that the external communities should be moderately coherent, internal teams very coherent, technological diversity moderate, and the number of direct and indirect contacts high. Cohesion with external groups is beneficial in the sense that when the members of the groups feel closer to each other, they tend to communicate more often and more freely. On the other hand there are also drawbacks if the cohesion is too strong: the information can be excessive and thus redundant, and being too tied to the common interest and norms might diminish innovation capabilities. Thus the results show that communities of a moderate level of external cohesion – not too much, nor too little – are most likely successful. Internal teams in turn are more successful the more coherent they are. The cohesion brings trust and facilitates more effective collaboration among the team members. It affects the members' motivation positively. In OSS projects the members tend to come from very different technological backgrounds. If the diversity is too high, the communication and achievement of common objectives in terms of the software become difficult. If the diversity is very low, the innovation capabilities are diminished. The direct contacts are costly to maintain and indirect contacts not likely to produce opportunities, but in spite the number of contacts – direct and indirect – increases the likelihood of success in the project. (Singh et al. 2009) Also Peng (2009) shows in his research that factors like communication and social capital are the key to success in OSS projects. They based their hypotheses on an assumption that innovation and knowledge creation is the basis for successful software development in

OS projects. They found out that the team should have strong knowledge base, and high social capital and diversity. (Peng 2009)

Restrictiveness of licenses has both negative and positive impact in OSS projects. First, the developers lose interest the more restrictive the license is, because their mindset is in the openness of the code, and the main reason for contribution. Secondly, the non-developer users and project administration value more restrictive licensing, because it gives them more secured business. (Subramaniam, Sen & Nelson 2009) Sen, Singh and Borle (2012) found out in their research that semi-restrictive licenses are the least favored among both users and developers. Restrictiveness of licenses thus makes it a factor influencing the success, because it is shown that the interest of both developers and non-developers is critical for the success of the OSS project (Subramaniam et al 2009; Sen et al. 2012).

Weber (2004) focuses on different characteristics of tasks and knowledge creation in OSS development projects. The community member should have also intrinsic motivation for the contribution. They should be able to see the common good and the value of enlargement their own knowledge base in the project. The developers should therefore have free access to all information regarding the project, that should also meet the developers' ethical values, and the product should have benefits for the users that would make it wide-spread. Midha and Palvia (2012) also stress the importance of user base by size and emphasizes the importance of localization. The project management issues such as responsibility assignment and modularity of the project positively correlate with the success of an OSS project (Midha & Palvia 2012).

At a general level the factors that influence competitiveness include licensing model, openness, applicability and targeting, continuance, product characteristics, volume, support, management of resources, expertise of local integrators, experiences and understanding, and functioning OS markets (Seppänen 2006). As discussed earlier, the type of license chosen affects both users and developers (Subramaniam et al 2009; Sen et al. 2012), but also the IPR management is important (Oksanen, Välimäki & Laine 2006b). The more open and flexible the development environment is, the more different possibilities and solutions it can offer for business. When the applicability of the product is wide and it is targeted well, the users are more likely to adopt it. The project should also guarantee continuity, not only continuity of the project management, or support-giving company, but of the whole project. If the continuity is uncertain, it is less likely that a user would adopt such software. (Seppänen 2006) Product criteria factors that are very important for the product in order to be successful include such factors as platform support, quality, usability, set-up costs, modularity, and integration possibilities with other software (Woods & Guliani 2005). Support is very important to the users, and is a critical factor for the success from the user side in OSS adoption (Seppänen 2006). Seppänen (2006) also talks about the community size: volume refers to the number of users and developers in the community, the higher the number, the more vital the community is. The last factors affecting success are related to management and markets. The awareness should be created and thus a market for OSS products. Increasing knowledge of the users and ensuring positive experiences for them are the key factors. Local operators also help the adoption. On the management side it is required that the structure of the community and its management are as important as the size of the community. The chain of command has to be clear and monitoring close. (Seppänen 2006)

The table below summarizes the different factors influencing OSS development projects.

Table 9 Critical success factors in open source software business

Factor	**Sub-factor**
Community management	External communities moderately coherent
	Internal teams very coherent
	Technological diversity moderate
	Number of direct and indirect contacts high
	Intrinsic benefits
Project management	Modularity
	High openness
	Ensuring continuance
	Establishing and communicating hierarchy well
Technology management	Good applicability
	Good quality
	Modularity
	Good usability
	Low set-up and usage costs
	Integration possibilities good
	Localization (language versions etc.)
License management	High know-how of licenses
Market management	Good product support
	Market creation

The views of the different authors on success factors in open source software business are summarized and categorized in five different categories; community management, project management, technology management, license management and market management. The community management is further divided into five sub-factors that refer to the coherence level of both internal and external members, indicating that new ideas must come from outside, but in order to ensure the functionality of the internal community the internal coherence should exists. Also intrinsic benefits are important to ensure in turn the sustainability and continuance of the community. Project management is divided into four sub-factors that stress communication, continuance and agile nature of the projects. Technology management is divided into seven sub-factors, which emphasize first of all quality through usability, version control, integration possibilities and localization, as well as low cost. License management indicates that good know-how is important. Market management is divided into good product support, which is a typical business model in OSS, and market creation.

3.3.3 CSF defined through risks and challenges

This study focuses on finding those success factors that are critical in the open source software industry, and therefore, for grasping an idea of the different critical issues, we take a look at the different challenges and risks generally associated with the industry and on the findings of other research relevant to the research purpose. These findings serve as a base for the personal interviews to be conducted in order to find a specific set of success factors that can be reviewed against the existing theories.

The study is limited to finding the CSF of the whole industry of OSS, not specific areas of OSS and not the whole software industry. Therefore it is important to be careful of which factors in theories should be taken into account. First the challenges and risks are reviewed, and they are those of OSS industry deriving from open innovation, because those are the ones differentiating OSS from traditional software industry.

The three basic challenges in open innovation were discussed in chapter 2; the challenge of diversifying internal innovation, balancing between external and internal innovation, and motivating the external innovation and contribution. The same challenges apply to open source software business. Along with maximizing the effectiveness of in-house R&D and managing the relation and roles of internal and external innovations, the key to any development in OSS software industry is the part of the contributors, the members of the development community (West et al. 2004). In the traditional business model (closed innovation model) the main goal is to be ahead of competitors and stay there by investing in in-house R&D and offering the best solution available (see chapter 2). In the different OSS business models identified above in chapter 3.1, there is one common interest regardless of the income model; the sustainability of the external development community. In case a software solution does not attract enough contributors, another complementary or substitute product with even just a small development community will step in and take over the market share of a product that is not being developed further. Even though in some of the nine business models the income is earned by selling a license, the reliance lies greatly upon the contribution of the community, and the companies that only sell OSS related services can lose their advantage of expertise in their respective OSS portfolio. (Daffara 2007) The key idea, therefore, is the motivation of the contributors. There are three ways the contributors get their motivation according to (West et al. 2004, 6):

- Direct utility; the individual or his employer gains from contributing directly
- Intrinsic benefit; the individual gets an intangible benefit or joy from contributing
- Signaling; the individual gains respect from other community members by showing his skills

The company can enforce these motivational factors by its own actions, e.g. by enhancing the development environment towards more intriguing for the developers, or granting annual awards or prizes, and thereby strengthen its position in the market.

Vainio, Vadén, Oksanen and Seppänen (2006) consider four aspects in sustaining an open source community; social, cultural, legal and economical. *Social sustainability* depends on the variety and nature of the community members, i.e. skilled vs. less-skilled developers, conflict resolution, decision-making and personal characteristics. The leadership should be strong in order to build coherent and scalable software because the multitude of developers translates to a multitude of uncontrollable modifications without a strong leadership. *Cultural sustainability* refers to possible cultural clashes deriving mainly from the shift of the communities from so called ‘hacker culture’ to a commercial and professional development culture. *Legal sustainability* can be threatened by several issues, mostly related to IPR conflicts. When OSS development was still a new way of making software, the idealism of freedom and transparent environment existed strongly, but along the development of the whole concept and the use of OSS for commercial purposes has brought along the need to have different IPR definitions, i.e. licenses. (Vainio, Vadén, Oksanen & Seppänen 2006, 4-10) Legal risks are closely related to licenses, and even though one might wonder why there is such a risk with products not sold with commercial licenses, there are many aspects that need to be addressed form legal perspective (Oksanen & Laine 2006a). The legal risk lies in license management related to adoption of OSS by a company. A large software package that is open source, might include incompliant licenses. This leads to IPR infringement issues. (Oksanen et al. 2006b) *Economic sustainability* refers also to the recent course the OSS development has taken; the use of OSS for commercial purposes. Many companies see the benefits of using OSS and some companies make OSS related business as their primary source of income, and therefore those companies are also prepared to pay for the development. The earlier ‘hacker culture’ has been overtaken by dedicated developers sharing the goals of their employers, which has changed the focus of the developer communities from merely ‘quality software’ to ‘quality software on schedule and within given framework’. (Vainio et al. 2006, 4-10)

Saastamoinen (2006) divides the risks in five categories at are related to OSS companies: IPR; patent and compliance issues; quality; responsibilities and guarantees; and change management. Risks related to IPR have to do with licenses. Many companies use extensive licenses to ensure the freedom of the code, but that makes it impossible to close the code they have produced by themselves from the competitors, which is considered a great risk in many companies. The license terms are often unclear and might cause problems if they are incompliant with another licensed code that is embedded, which means in practice that the reusability is controversial. The code can also contain parts that infringe a patent, that is not known, or the owner of the code might be unclear, when there is uncertainty whether the code can be used freely. Having the software comprised of parts from multiple sources, not knowing whether all the original sources and licenses are documented complicates the IPR issue greatly and makes it basically impossible to trust the freedom of the code. The current lack of court cases adds also to the uncertainty. Technical quality cannot be assumed, and might not be known in advance. There are faulty software in both proprietary and OS software side. On the other hand, OSS might have fewer bugs if the community is large, because more testers can find more errors, and such vast resources as a large OS project can offer are not possible in a proprietary company. Still, there are no guarantees that the software that is used works, unless it is acquired against payment from a company offering support or guarantee as a service. If the software does not function as required, the user is on its own, and might increase the initial costs. There might be changes in the community that are a great risk to the OSS company. For unanticipated reasons the development community can all of a sudden start to die, i.e. the developers stop developing and/or move to another project. That means that the company must try to get the community back to life or take over the development which means slowing down or major investment in resources. (Saastamoinen 2006)

Below is presented a table that summarizes the different factors that could influence the success of an OSS project based on literature review on different challenges and risks the project can face.

Table 10 Suggestions for critical success factors in OSS industry

Factor	**Sub-factor**
Innovation management	Motivating external contribution
	Clarifying internal/external roles
	Ensuring effective in-house development
Legal issues	Taking care of IPR management
	Securing against patent infringement
	Taking care of license compliance
Technology management	Extensive testing for quality and careful selection of code
	Getting code from most reliable and stable community

The factors discussed above based on risks and challenges found to exist in OSS business (see chapter 2) are summarized and include innovation management, legal issues and technology management. Innovation management is divided into three sub-factors that refer to effective management of internal and external resources. Legal issues are divided into three sub-factors, which emphasize the importance of knowledge of IPR and licenses. Two sub-factors of technology management suggest that it is important to select and test the code carefully, also keeping in mind the continuance.

3.4 Measuring success

Defining success is a difficult task that many authors have addressed. Stevens and Burley (1997) define successful innovation as follows:

> *"Commercially successful" as we use the term does not just mean that someone is buying the product or licensing the concept, but that the concept is providing economic profit to the parent company. In other words, the money returned is greater than all the money invested in creating that product, including the cost of capital (both depreciation and opportunity cost), raw materials and manpower used throughout the entire project. (Stevens & Burley 1997, p 16)*

The most common ways of measuring success are related to sales and growth (Hendersson & Cockburn 1994; Davidsson, Steffens & Fitzsimmons 2008). Other metrics used are for example marketing related, such as market share and competitiveness (Heinonen 2009). Firm growth is the most commonly used as measure of success. However, growth alone is not enough to explore the success of a firm, but instead it needs to be combined with profitability. Empirical tests show that the companies should first establish high profitability in the expense of growth and not vice versa. The companies will then be more successful in the long term. (Davidsson et al. 2008) The paper of Chakravarthy (1986) also shows that traditional measures based on the profitability are not enough. The paper claims that measurement should be done in two ways: assessing the quality of the company's transformations and measuring the stakeholders' satisfaction (Chakravarthy 1986). Peters and Waterman (1982) used two different ways of measuring the success in their famous book In Search of Excellence. First they used financial criteria that included: compound asset growth, compound equity growth, ratio of market to book value, average return of total capital, average return on equity, and average return on sales. The other measure they used was innovativeness and the ability to react to rapid changes in the firm's environment. (Peters & Waterman 1982) Edmunds (1982) gives a comprehensive set of measurement tools for SMEs in growth that uses very different type of measuring. The primary goal of the company is to increase the profits and the market share. The ways by which to measure the performance in order to manage the two are: flow of total funds, product-market plan, market performance, costs, assets, financial measures, management capacity, and taxation. (Edmunds 1982)

Looking at software development specifically, the authors DeLone and McLean (1992, 2002, 2003) distinguished six metrics for measuring a software development project success: system quality, information quality, use, user satisfaction, individual impact and organizational impact. System quality refers to the quality of the code and information quality to the quality of the documentation, i.e. how well the system structure and development work has been described. Use metrics mean the level of usage measured by different ways depending on the type of distribution method of the software: number of users, number of downloads, inclusion of software in other distributions, popularity of the web site or other measurable place for information, package dependencies, and reuse of code by external developers. User satisfaction can be measured by different user rating systems or by conducting user surveys. Individual and organizational impact refer to different economic and other type of impact on the individual and the organization. (DeLone et al. 1992, 2002, 2003) Crowston, Howison and Annabi (2006) studied FLOSS (free/libre/open source software) development success, and used the measures of DeLone et al. (1992, 2002, 2003) as basis for their study. However, they felt that there was a need for additional metrics for success measuring in FLOSS development projects. They added project output, process, and outcomes for project members. Project output is very often a milestone itself. Advancing from a semi-finished beta version of a software to a functioning and stable first alpha version is a significant step that indicates some degree of success of the project. This leads to developer satisfaction, in case the version can in fact be called alpha, and the goals that were initially set were met. Process refers to factors that have to do with the process of achieving the desired output. First indicator of success is the number of developers. FLOSS project depends on the community and the size of the community speaks of attraction towards the project, and as discussed earlier the attractive projects tend to be more successful than less attractive ones. Number of developers is not a measure on its own, if there is not enough activity generated by the community. The process can also be seen successful based on time between releases and the time to solve the bug issues and implement new requested features. Both releases and error-fixing demonstrate the level of activity in the project. Outcomes for project members include such as job opportunities, intrinsic benefits, salary, reputation building and knowledge creation. The more the benefits that come out of the project for a team member, the more successful the project is. (Crowston et al. 2006)

In this research the measures of profitability and growth will be used for measuring the success. The study uses qualitative methods for empirical data gathering and therefore very precise measures are unnecessary. The study will focus on the interview answers what comes to success measures.

3.5 Synthesis

The study is constructed in three main parts: the first part explains the main concepts, the second part is a literature view on key issues, and the last part is the actual study. The first part of the study focuses on two concepts: open innovation and open source. The path from the closed innovation model to the open innovation model is called the paradigm shift, and means that a company, group of companies or an entire industry has realized the deficiencies of overlapping R&D and innovation development, and the potential of a creation of ‘virtual laboratories’, where the minds of the brightest people are not a sole propriety of one company, but vice versa, benefit all. Some companies might benefit directly from open innovation, some indirectly, but the main idea is the effective use of all resources, because at the end, there is always a way how a company benefits indirectly from the development. Open source is a phenomenon of open innovation, and as named, means that the ‘source is open’ and anyone can use, develop and distribute it (cf. GNU). The study explains that open source is not in fact only software related, but there are other projects that can have free source; like open education and open content. Licenses are the key issue in open source software business. Licenses secure the freedom of the source code and determine the degree of freedom from different aspects.

In the second part the study links open source to the paradigm shift by explaining the difference between the traditional and the new open innovation business models in software business. The business models, or more precisely revenue generation, of open source software industry differ greatly from those in proprietary software industry. Open source software industry business models are mainly based on pure service on the software – not the software itself. (Koski 2006) Critical success factor is a concept that is the heart of this research. Critical success factors are those factors that affect the business substantially in the positive or negative direction (Hofer et al. 1987). The CSF are very firm-specific, although there are those that are also industry-wise (Ellegaard et al. 1993). Therefore the study has defined the CSF at three different levels in two axis: software industry level vs. open source software industry level, and on another axis open innovation vs. open source software. The study looked at software business CSF from three different viewpoints and found out that important factors in software business are a good project plan and project management, good product and resource management, and partnering. Software projects are usually long term, and therefore project management plays a critical role in the project. Product is what is sold, and in software business products are always defective to some degree, and so the differentiation can be made in the quality of the product including factors like agile methods, testing, and quality management. Resource management is also important; taking care of the resources so that there is enough resources, all resources are well-equipped and motivated. Open source software industry differs from the proprietary software industry as the literature review has shown. Based on literature the main differences comparing to proprietary software industry are related to community, communication and licenses. Community management includes motivation with intrinsic benefits, which is discussed more in detail in the last part where open innovation viewpoint is reviewed; communication capabilities and levels between the different operators; and number of those operators. The community is a more complex organization than a proprietary firm producing software, because the clear and classical hierarchy models are missing. Therefore proper and accurate communication is found to be very important. License know-how is also found to be an important factor in making success. Open innovation challenges faced in the open source software business are examined more closely from two different – but not contradictory – theoretical viewpoints. The literature review shows that the most important challenge specifically in the open source software industry is the motivation and contribution efforts of the developer community. Concluding from the four aspects in sustaining open source communities stated by Vainio et al (2006) and the three motivational factors distinguished by West et al. (2004), the main challenges in sustaining an open source software developer community today derive from the shift of open source software development from a pure hacker society, where the sustainability is achieved by intrinsic benefits or other personal immaterial rewards gained by the developer, towards a company-driven community, where the motivational factors lie mostly in direct financial benefits. As has been discussed, the paradigm shift seems to be inevitable, and therefore the open source software industry as part of open innovation will be developing towards an increasing commercialization, which indicates a change in the hierarchy, construction and nature of the developer communities. Thus, the challenges proposed by efforts to achieve open source community sustainability are constantly changing, which means that intrinsic and other immaterial benefits alone will not ensure the sustainability of the community, which is a prerequisite in the long run in order to be able to exploit an open source software successfully.

The study has looked at the different attributes of open source software industry, defined the success measures and listed different factors that might affect the success of an open source project. Three different tables were constructed based on existing literature and research: success factors of software industry, success factors of open source software industry and success factors in open source software industry based on risks and challenges of open innovation. The synthesis on the following page summarizes the literature review.

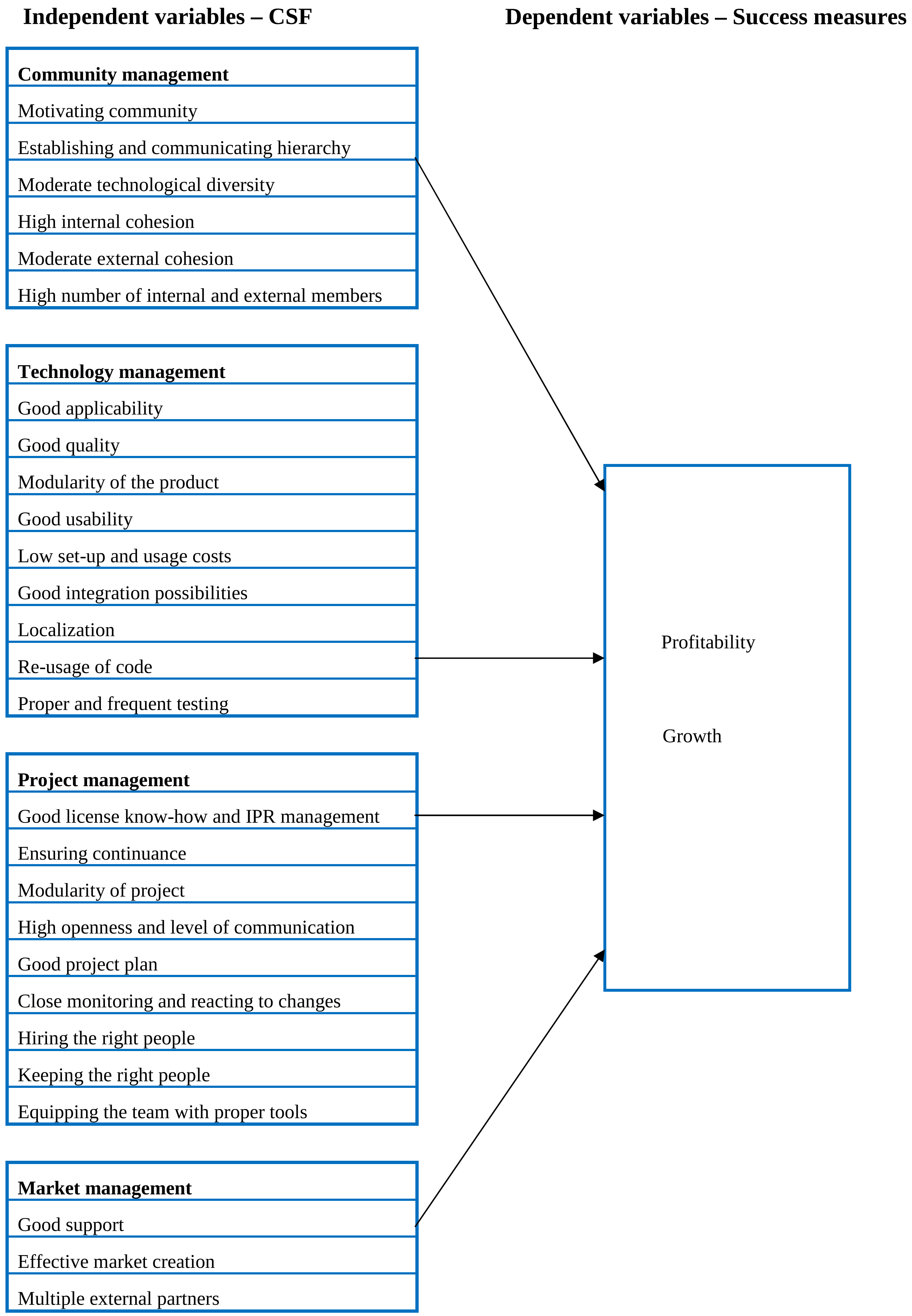
Independent variables – CSF
Dependent variables – Success measures
Community management
Motivating community
Establishing and communicating hierarchy
Moderate technological diversity
High internal cohesion
Moderate external cohesion
High number of internal and external members
Technology management
Good applicability
Good quality
Modularity of the product
Good usability
Low set-up and usage costs
Good integration possibilities
Localization
Re-usage of code
Proper and frequent testing
Project management
Good license know-how and IPR management
Ensuring continuance
Modularity of project
High openness and level of communication
Good project plan
Close monitoring and reacting to changes
Hiring the right people
Keeping the right people
Equipping the team with proper tools
Market management
Good support
Effective market creation
Multiple external partners
Profitability
Growth

Figure 6 Synthesis

4 METHODOLOGY

4.1 Research method

This study aims to find the key factors for successfully making business out of commercial open source software development. The study first concentrates on the existing literature and theories, in order to find out the different factors that might have an effect on OSS business in terms of successfulness. Three different ways are used for finding the factors in the literature review:

- Exploring the literature on software industry in general and make a summary of the different findings of CSF in software industry
- Exploring the literature on OSS industry in general and make a summary of the different findings of CSF specific to OSS industry
- Exploring the literature on open innovation and OSS related challenges and risks, conclude the CSF specific to OSS industry based on those findings, and finally make a summary of the different factors concluded

The three different summaries will form the synthesis and framework for the qualitative study. The synthesis thus includes all the possible factors that influence success of a commercial OSS project, the generalization being at the level of software industry.

There are two major types of research models: quantitative and qualitative. Quantitative approach can be said to be the traditional approach, whereas qualitative is more modern. Research intends to find out reality is like, and quantitative and qualitative researches offer a different kinds of realities, the other one objective and other one subjective. (Clarke 2005) Quantitative research model can be characterized as more objective, as the researcher has no direct or personal contact with the interviewee. The questions are pre-defined and very structured, and the process is deductive. Qualitative research is characterized as subjective. The researcher interacts with the interviewee constantly in the interview situation and therefore the process evolves depending on answers given. The qualitative process is inductive as the interview shapes the process and the theories are developed for understanding. (Creswell 2003)

This study has qualitative research approach. The reasons for qualitative approach are numerous and partly practical. Morgan and Smircich (1980) claim that qualitative methods are appropriate when the human beings can be seen as engaging in the shaping of the reality measured in the study, and when the world can be seen as dynamic. This research is a good example of such situation. The concepts under examination are not directly measurable and there is indefinite number of choices, thus quantitative approach would have to assume too much based on concrete and immobile reality. The qualitative nature allows more flexible data search, which is especially important in a search for such factors that the interviewee might have been unaware of. Qualitative research facilitates finding the most relevant data based on factors that impersonal questionnaire could not communicate. Qualitative method can better offer new data, which was not found in existing literature. In sum, qualitative approach can offer a subjective view of reality, which limits the generalization, but on the other hand, it can offer wider view than would have been possible with a quantitative approach. (cf. Alvesson 1996; Conger 1998; Padgett 1998)

Hirsjärvi and Hurme (2001) differentiate three types of interview methods: structured (lomakehaastattelu), semi-structured (teemahaastattelu) and unstructured (avoin haastattelu), where the conformity of the interviews diminishes when going from structured to unstructured. Structured interviews are pre-defined where the questions are set as a form that is followed strictly. The method does not allow going outside of the topic, but instead keeps the interview controlled. The method is the easiest of the three to conduct, but requires careful preliminary planning when setting the questions. The method is very close to quantitative research method, but the difference is that the interviewer is noting down the answers instead of the interviewee and a personal relationship is established. Semi-structured interview is a method between structured and unstructured interviews. The interviewer has pre-set themes of questions that are to be followed but emergent issues can be addressed, i.e. it is possible to explore issues that come up, but are not among the actual interview questions. Unstructured interview is more like a discussion than an interview. The interviewer has freedom to lead the discussion to a certain direction based on the interviewee’s answers. The questions are open-ended. This method is useful when all the answers cannot be anticipated, and when the number of interviewees is small. (Hirsjärvi et al. 2001) Although, unstructured nature of interviews may affect the validity of the research, and therefore it is important that the interviewer is knowledgeable on the subject and is able to guide the interviewee to the right subject without influencing the answers. The interviewer should also be able to make consistency out of the answers of the different interviewees. In this study the interviewer has a background in OSS industry and the knowledge therefore derives also from practice instead of only the literature review presented in this paper. The interviewer also knows three of the interviewees personally through professional intercourse, and can interpret those interviewees’ answers also in the light of personal characteristics, which diminishes the errors.

The data collection in this study is conducted through unstructured personal face-to-face interviews which according to Hirsjärvi et al. (2001) facilitate very in-depth communication. Qualitative interviews can be characterized as emergent, meaning that the questions and question setting can be influenced by the previous answers (Creswell 2003) In the interviews conducted in this study the questions are open-ended because the interviewees should not be guided towards any answers anticipated by the interviewer, and also it is possible that the interviewer has not been able to ask the right questions. Also, the interviews are unstructured so that unexpected ideas can be explored effectively. The number of the interviewees is small, which makes it possible to have long in-depth interviews.

The qualitative nature and the question-setting are also limitations of the study. It is important not to lead the interviewee towards any CSF found in literature. On the other hand, the interviewee might not simply think deep enough to find the possible underlying factors that explain the more obvious factors. Quantitative survey would be a good idea for future studies, but one has to be very careful in how many questions are asked and what those questions are. The more choices of different variables (CSF) there are, the more leading the survey becomes. The fewer choices there are, the more likely the interviewees are to answer with thought.

After constructing the theoretical part, the interviews are conducted. In the light of the theoretical framework the results of the qualitative interviews are then examined and compared. Finally conclusions can be made based on theoretical and qualitative analysis.

4.2 Data collection

Odendahl and Shaw (2002) point out that acquiring interviewees can be a difficult task. One good way is to raise the interest of the interviewee candidate. Sometimes a prominent actor in the fields has such interest in the topic that the person wants to be included in the data collection process – as an informant, that might also help to provide or suggest other interviewee candidates, and also provide personal knowledge in constructing the study. (Odendahl & Shaw 2002) Wilkinson and Young (2002) discuss snow-ball sampling, which means that one interviewee suggests a name of one or more other possible candidates, in which case it is possible to start with only one good interviewee. In this study snow-ball sampling was used, as well as raising professional interest of the interviewee. The author is a member of OSS organization, and thus has acquired some contacts through personal relationships and gotten further recommendations. The topic raised wide interest among most people the author discussed with, since the business side of OSS industry has only recently been studied more widely.

Personal interviews are a very effective method of data collection in qualitative research (cf. Odendahl et al. 2002). The personal interviews were held in the spring 2011, keeping in mind that they should be held quite close to each other time-wise, so that the fast development in the sector could not distort the answers in relation to one another. The interviews were individual and each was 1-1,5 hours long. Each interview was held in Finnish and recorded. There were 5 interviewees that held the following profiles at the time:

Table 11 Interviewee profiles

Interviewee	**Sex**	**Sector**	**Status**	**Previous relationship**	**Date**
1	male	Public	Director and OS specialist	Yes	16.3.2010
2	male	Public	Researcher and OS specialist	No	3.5.2011
3	male	Private	Managing director in OS company	Yes	19.4.2011
4	male	Private	Entrepreneur – OS company	No	29.4.2011
5	male	Private	Managing director in OS company	Yes	20.4.2011

The interviewees were chosen based on their good established reputation in the OSS community in Finland, as well as their different background relating to the organization they are part of. Out of 20 candidates five were chosen. The small number of interviewees compared to the number of total candidates was due to lack of time as well as the fact that the representation of the people in the field among the interviewees was sufficient.

All interviewees are specialists through either research or practical experience in open source or open source software. They are all Finnish, male and aged between 30 and 40 years old. Some of the interviewees chose to stay anonymous, and therefore none of them are named, instead they are numbered from 1 to 5. The interviewer had professional relationship with three of the interviewees beforehand.

The interviewees were sent a brief explanation of the topic beforehand, so they could prepare for the interview by thinking a little bit of the subject. They were not sent any questions in advance, in order for the answers to be spontaneous by avoiding any previous research to be done by the interviewees that might have influence on the answers. The interviews were conducted in the place by the interviewee’s choice, one in a cafeteria (number 4) and four others in their respective workplaces.

The interviews were unstructured, but generally the information that was sought for was:

- Background of the interviewee
- History of the organization the interviewee works for
- Relationship to OS and OSS
- Knowledge about business models
- Knowledge about success factors

- Opinion about business models
- Opinion about success factors
- Examples
- Suggestions for future regarding OS and OSS

The interview process can be highly influenced by social settings. It is important that different issues such as age, dressing, role, authority, etc. are taken into account beforehand. (Odendahl et al. 2002) Wilkinson et al. (1994) have identified three types of preparation to be made before the interviews: logistics preparation, preparation of physical properties and mental preparation. Logistics preparation means preparing the place of the interview and considering the means to get there. Preparation of physical properties means organization of the venue, dressing appropriately, props and recording. Mental preparation means considering beforehand how to manage the interview situation. The interviews in this study were held in the places of the interviewees' choices, and means to get there and preparation was different in each case. All the interviews were on schedule and there were no confusions. Making the interviewee feel comfortable and relaxed in the interview situation is important because only in that case the interviewee would have a free flow of thoughts (Wilkinson et al. 1994). In this case it was done by dressing semi-informally, according to the person's status and personality (if known beforehand). As some of the interviewees already knew the interviewer beforehand, the expertise or knowledge of the interviewee did not need to be addressed in all of the interview situations. In two cases, however, it was important to communicate the level of understanding – especially technical – of the interviewer to the interviewee, so that the interviewee could formulate the answers accordingly, and also know how deep in technical details would be possible to go to. Speaking "the same language" is essential when making the personal contact in the interview situation and building a relaxed atmosphere (Wilkinson et al. 1994). Tape recorder was used in the interviews and the interviewees were told of the recording beforehand and thus given the opportunity to decline.

4.3 Data analysis

Data analysis is the process in which the acquired data is organized in a way that it can make sense Creswell (2003). As data in qualitative research is made up of answers to open-ended questions, they cannot be numerically expresses as such. Therefore the analysis requires systematic process in order to fully understand the data and to answer the research questions. (Taylor-Powell & Renner 2003) Creswell proposes six steps in the data analysis process, which are presented in the figure below:

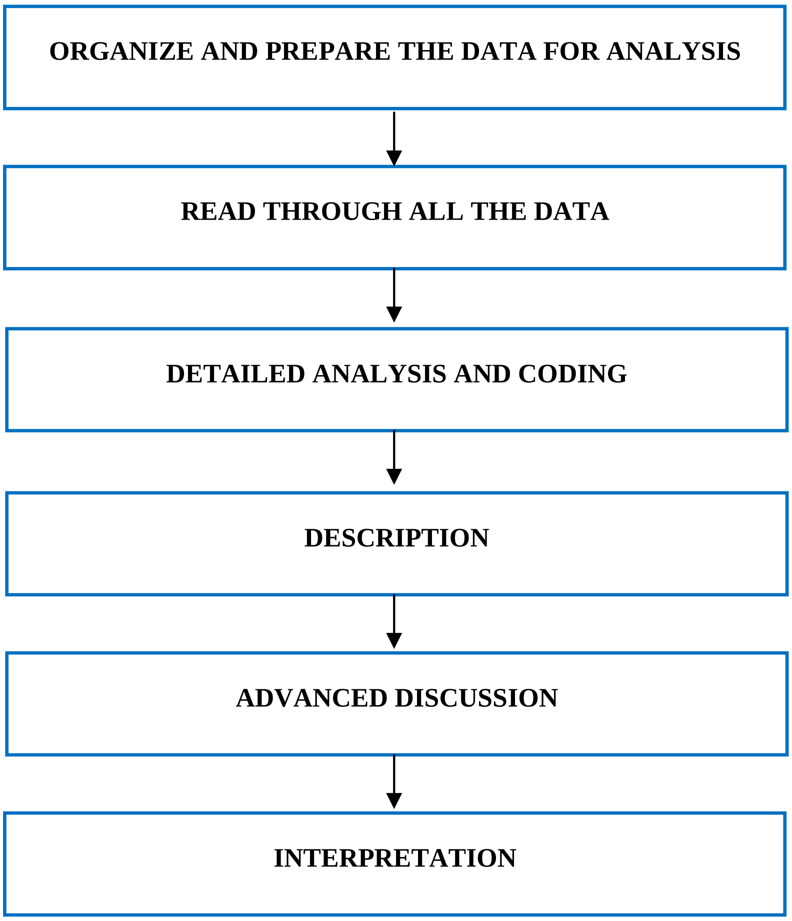
ORGANIZE AND PREPARE THE DATA FOR ANALYSIS
READ THROUGH ALL THE DATA
DETAILED ANALYSIS AND CODING
DESCRIPTION
ADVANCED DISCUSSION
INTERPRETATION

Figure 7 Data analysis process by Creswell (Cresswell 2003, 218-222)

Organizing and preparing the data for analysis includes transcribing the interviews, collecting the field notes, and making note of all information collected. Reading through all data is a step in the process where a general understanding of the data is acquired. Detailed analysis and coding is the next step, where first the data is categorized under themes or groups according to the meaning. This helps the researcher make a more detailed analysis on the data – what are the actual meanings and getting the sense of the whole data. The different categories should be clustered and iterated as far as possible. Description involves describing the setting, which means the people, places, etc in the setting. These can also be coded, and help in understanding the data deeper and choosing specific outcome categories in the coding step. Advanced discussion is the representation of the data collection and data in the research paper. It can be a description of the events, followed by discussion of the themes, illustrations, quotations, etc. The final step, interpretation, should discuss what is learned from the data and what are the issues that are the most meaningful. (Creswell 2003)

In this research the transcription was not made to full extent due to poor quality of some of the recording caused by background noise. However the sentences were written down as far as it was possible and tones of voices and reflections were noted, for example how strongly the interviewee stressed an issue. The field notes were also very much used in the process. Halcomb and Davidson (2005) make a case against the necessity of detailed transcription of the data, especially in mixed methods, where the interviews are not purely qualitative, but include some themes in the question setting. They argue that it is not always necessary to transcribe the recordings to full extent. Instead they propose another process for data analysis, which includes field notes as an integral part of the analysis. (Halcomb & Davidson 2005) The figure below illustrates the process of Halcomb et al. (2005):

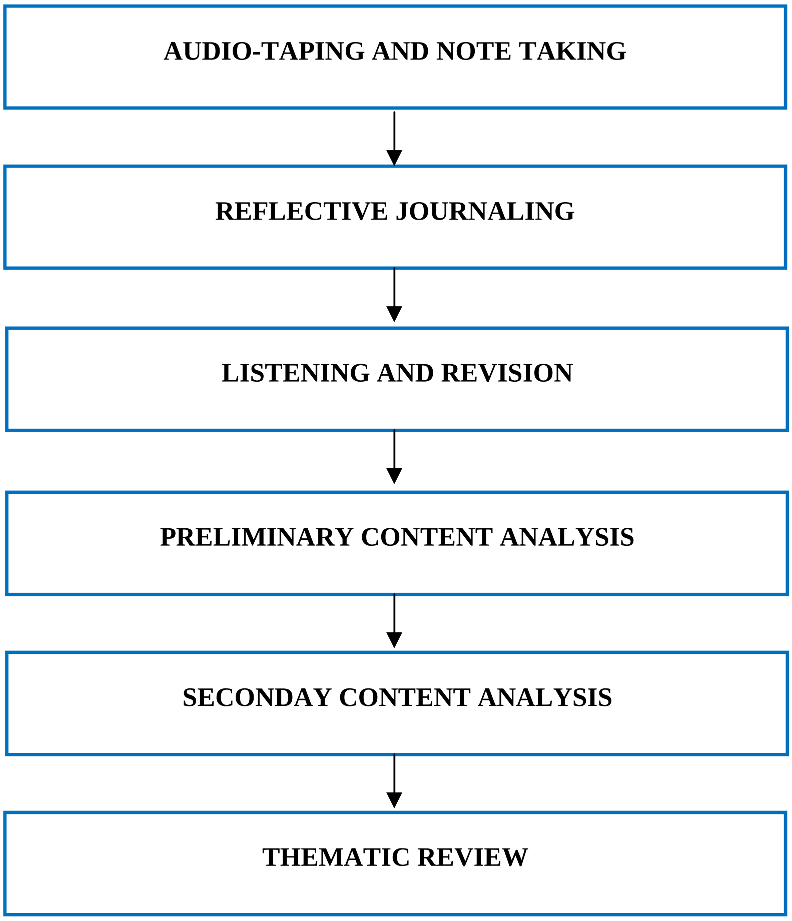
AUDIO-TAPING AND NOTE TAKING
REFLECTIVE JOURNALING
LISTENING AND REVISION
PRELIMINARY CONTENT ANALYSIS
SECONDAY CONTENT ANALYSIS
THEMATIC REVIEW

Figure 8 Data analysis process by Halcomb at al. (Halcomb et al. 2005)

Audio-recording supported by taking field notes is a good process for adding more context to the meaning of the interview data. Immediately after the interviews the researcher should review the interview carefully and possibly amend the notes with new ideas and reflections. The listening and revision means that after reflective journaling the researcher should carefully listen to the tapes, often several times, amend the field notes with the data on tape and also correct the field notes if necessary. This enables the researchers to get a full and clear picture on the content of the interviews as well as the context and reflections. Next the researcher should conduct the preliminary content analysis, once he/she is sure the data is accurate. This involves categorizing the data in themes, which can be referred to being the coding process. Secondary content analysis involves a second researcher to perform the listening and revision step, to ensure the preliminary analysis is accurate. Finally the thematic review involves re-listening to the recordings and making a final analysis of the content. (Halcomb et al. 2005)

Like already mentioned, in this research notes were taken during the recording, taking down the main messages added with other notes, e.g. the tone or attitude of the interviewee. The tapes were listened immediately after the interviews and notes were taken. The actual data analysis was performed approximately one year after the interviews were held, so the tapes were listened to a second time and more precise notes were taken, though not transcribed to full extent. The data acquired through field notes and the notes from the tape-listening were reviewed and categorized according to the synthesis based on literature review. Each item in the synthesis was gone through and correspondent answers were categorized under the item. Finally the outcome was a table where each of the factors regarding the research question had correspondent answers.

4.4 Trustworthiness

Trustworthiness in qualitative research differs slightly of that in quantitative research (Creswell 2003). Traditionally trustworthiness in quantitative research has been divided into four different categories: internal validity, external validity, reliability and objectivity. Internal validity is the extent to which there exist causal relation between the dependent and the independent variables. External validity means the degree to which the causal relationship can be generalized. Reliability refers to the quality of the measurement, and is usually tested by replication. (Lincoln & Guba 1985) Qualities of reliability are dependability, stability, consistency, predictability and accuracy (Kerlinger 1973). Objectivity is the opposite of subjective and refers to the methodology; the study can be seen as objective if multiple observers would come to the same conclusion as one. (Lincoln et al. 1985)

Due to the nature of quantitative study and thus the trustworthiness qualification above, Lincoln et al (1985) suggest four other means of measuring the trustworthiness of the study, that are more suitable for qualitative study using modern methods (naturalistic). The following table summarizes the means of establishing trustworthiness in qualitative study (Lincoln et al. 1985):

Table 12 Establishing trustworthiness in qualitative research (Lincoln et al. 1985)

Conventional	**Naturalistic**	**Methods**
Internal validity	Credibility	Prolonged engagement
		Persistent observation
		Triangulation
		Peer debriefing
		Negative case analysis
		Referential adequacy
External validity	Transferability	Contextual description
Reliability	Dependability	Detailed research outline
Objectivity	Confirmability	Triangulation
		Audit trail

Internal validity as conventional measure of trustworthiness is somewhat assuming knowing in advance, so Lincoln et al. (1985) have an equivalent measure for qualitative methods called credibility. Credibility rather answers to the question: How well do the findings represent the reality? There are several ways of enhancing the credibility. First, it is important to be well prepared. The researcher should carefully examine previous literature and findings of similar research and also get to know the topic and e.g. background of the interviewees well, i.e. be qualified to perform the inquiry. The interviewees should be selected as randomly as possible to ensure a representing sample of the reality, and the trustworthiness of their answers should be ensured by e.g. giving the opportunity to refuse and selecting those genuinely willing to participate, and through iterative questioning. Also superiors and colleagues should be consulted frequently. Transferability is the naturalist’s equivalent for external validity. In qualitative research it is very difficult to prove the findings applicable in wider selection, because qualitative research is usually specific to certain environments. Lincoln et al. (1985) suggest instead that with the findings it would be important to provide as much contextual information as required for the transfer to be able to be done. In other words, the researcher has to ensure a proper understanding of the research to the reader. Dependability refers to the ability to repeat the research, but not necessarily the outcome, which would be the case in conventional reliability. In order to achieve this, the research methodology should be described in detail, so that the exact same study could be performed. Confirmability is the equivalent of objectivity, and means that the data should not be influenced by the interviewer’s own perception, but instead all data should derive directly from the informants. The bias can be avoided by triangulation and recognition of the possible problem by the interviewer.

For this study approximately 20 people were asked for an interview and all of them were interested in the topic and gave a positive answer. Among all of them a few candidates were chosen for personal interviews based on their relevant position in the industry in Finland, their expertise in OSS and their background. It was possible to make careful selection, because almost all candidates that were asked for an interview gave a positive answer. The interviewees were chosen based on their expertise and status in OSS world. There were two main criteria:

- Established high status in the field (reputation)
- Different background from other interviewees in regards of field of business, position in the organization, private/public sector

The qualitative research interviewee group is homogeneous in demographics, which is a limitation in the results of the study. However, Finland is very advanced what comes to open source, and also the nature of open source dictates that it does not know national borders. Open source as a phenomenon is global and the circles are small. Also, as in IT sector in general, male is the dominant sex in the field. Thus, the nationality or sex of the interviewees does not distort the results, because the interviewees can be seen as typical experts of open source software. The interviewees were chosen from both private and public sector, in order to get the two different angles in the study.

The researcher has also been involved in OSS industry and studied OS beyond this research paper, so the preparation or qualification criterion is fulfilled. The knowledge of the researcher extends to not only the theoretical background of the subject in hand, but also to the structure of the field in Finland and the position of the interviewees chosen. This study has offered description of the background in order to understand the underlying factors leading to the research problem and the data collection. Also data collection method and sampling have been introduced, as well as description of the actual data collection with schedule. The interviewees were aware beforehand, or were made aware, of the background and level of knowledge of the interviewer, and a relaxed and comfortable atmosphere was created. The researcher knows some of the interviewees personally at professional level, which has its positive and negative sides. First, the relationship enables more comfortable setting for the interview, makes it easier to prepare, and also save time with background questions. However, the personal relationship might affect the interviewer’s perceptions, because some of the issues and opinions might have been assumed beforehand. The interviewees were clearly informed and instructed before the interviews, so they had some time to think about the issues before the actual questions, so that the most underlying factors could emerge. However, the interviewer was careful not to present any findings of the literature review before the interviewee had communicated all of his own experience and opinions in the interview. Confirmability is ensured by the fact that the interviewer was surprised of the findings and did not expect such an outcome, which means that the interviewer’s own expectations or possible influence on the outcome was diminished.

It is important to notice that the results are valid for a specific date. The software industry in general is constant change, especially because of the emergence of OSS. OSS business is a relatively new type of business that has yet not found its permanent place in the market. Therefore it can be assumed that the constant evolvement can change the outcome of similar research in an even one year time span. The study being tied to a certain time also has its ties to transferability and dependability, as according to conventional criteria for trustworthiness those two criteria would not be fulfilled. In naturalistic view the qualitative nature of the study permits the repetition and applicability being limited to research setting without repeating the same outcome.

5 SUCCESS FACTORS OF OPEN SOURCE PROJECTS

5.1 Main findings

The main outcome of the interviews is that mostly the interviewees opinioned that OS management does not differ from management of proprietary software or that of any other field of business.

> *Qualities of a good OS project are the same as in any other business* (Interviewee 3 2011)
>
> *General critical success factors are valid also in open source.* (Interviewee 3 2011)
>
> *There are no specific critical success factors in open source business… It does not matter what model we use here.* (Interviewee 5 2011)
>
> *No difference.* (Interviewee 4 2011)

Additionally, the answers to the question of success factors were:

- low price
- excellent product
- good business model development,

of which none is in the list of CSF in the synthesis.

In spite of the main opinion the interviewees mentioned a few factors in the synthesis, although they do not see them as part of the management that differentiates success, therefore they can be seen as *key* success factors. Next each of the CSF categories specified in the synthesis are reviewed in detail.

5.2 Community management

The theory suggests that community management is very important in making successful business, and its elements being key success factors. All the factors under community management are OS related, because community is a concept that does not exist in proprietary software development world. Below is presented a table which indicates the positive or negative response of each factor in the synthesis from the interviewees. A plus sign means a positive correlation and a minus sign a negative correlation in the opinion of one interviewee between the possible success factor in question and dependant variables.

Table 13 Results: community management

Factor of community management	**Response**
Motivating community	+ -
Establishing and communicating hierarchy	+ + + +
Moderate technological diversity	not mentioned by interviewees
High internal cohesion	not mentioned by interviewees
Moderate external cohesion	not mentioned by interviewees
High number of internal and external members	+ - + - -

Interestingly, the only factor that was clearly meaningful as a success factor – four out of five mentioned it – was *Establishing and communicating hierarchy*, which is the one of these factors that is also related to other fields than OS. Contradictory to the theory was the issue of the *High number of internal and external members*. Three out of five interviewees thought that the size of the community does not matter, and could be even better if the community is not too large, because that would make it easier to manage, which leads to *Establishing and communicating hierarchy*. One interviewee said *Motivating community* is important and one said the community takes care of its own motivation. Interesting discussions were carried out regarding the financial benefits versus intrinsic benefits for the community. Two of the interviewees claimed that the intrinsic benefits are only a means to get financial benefits, e.g. through higher expertise or reputation that leads to paid consulting work. One claimed – as according to existing literature – that intrinsic benefits are important and sufficient.

5.3 Technology management

Technology management can be seen as quite crucial in software business, because the product is in the core of the business. Below is presented a table which indicates the positive or negative response of each factor in the synthesis from the interviewees.

Table 14 Results: technology management

Technology management	**Response**
Good applicability	not mentioned by interviewees
Good quality	+
Modularity of the product	not mentioned by interviewees
Good usability	not mentioned by interviewees
Low set-up and usage costs	-
Good integration possibilities	+
Localization	+ +
Re-usage of code	not mentioned by interviewees
Proper and frequent testing	not mentioned by interviewees

In the part of technology management factors only *Localization* seems to have moderate importance according to the interviewees. In OSS development – in case the community is large – it is possible to reach a level of localization much more impressive than in proprietary software projects. If the community is global – as it very often is – the developers or community members have their own interest in localizing the software for their environment, and then it also gets done. The localization can be said to be almost automatic compared to the efforts in proprietary software development projects. The localization is therefore important, but that is the same for both proprietary software and OS software.

5.4 Project management

Project management was seen very important in the literature view by many authors (eg. Reel 1999). However, most of the factors were not even mentioned by the interviewees. Below is presented a table which indicates the positive or negative response of each factor in the synthesis from the interviewees.

Table 15 Results: project management

Project management	**Response**
Good license know-how and IPR management	- + +
Ensuring continuance	not mentioned by interviewees
Modularity of project	not mentioned by interviewees
High openness and level of communication	not mentioned by interviewees
Good project plan	not mentioned by interviewees
Close monitoring and reacting to changes	- +
Hiring the right people	+
Keeping the right people	not mentioned by interviewees
Equipping the team with proper tools	not mentioned by interviewees

Good license know-how and IPR management is seen a very important factor in in existing literature. The interviews showed it is not necessarily so. One interviewee claimed that it is not important at all in OS world, since the software is ‘free’ and the licenses are made to protect the freedom. However it seems from the programming aspect that there is importance, because the licenses affect the usage, or restriction of usage, for commercial purposes or closing the code Since code is re-used, all of a sudden the software can contain thousands of different pieces of code that are published under different licenses. Sorting out and managing the combination of code is difficult.

5.5 Market management

Since OS is fairly new phenomenon, it is no surprise that *Market creation* was a topic of discussion with some interviewees. Below is presented a table which indicates the positive or negative response of each factor in the synthesis from the interviewees.

Table 16 Results: market management

Market management	**Response**
Good support	not mentioned by interviewees
Effective market creation	+ + +
Multiple external partners	+

The main aspect according to three interviewees in market creation is firstly increasing the awareness and secondly educating the customer. Based on interviews the customers face the main challenge of not knowing what to choose, since the OS projects are infinite in number and not much reliable knowledge available. Additionally it is possible that OS has a bad reputation – not to mention the image of being ‘free of charge’, although the right term is ‘free of use’. This all confuses the customer and at the moment of the interviews in Finland there were no companies offering purely OSS selection services.

6 CONCLUSIONS

6.1 Theoretical contributions and managerial implications

Open source software is a new field of business that has been studied, but mainly on technical level. The business level research has started gaining foot only recently. Especially success factors specifically from the business perspectives have been neglected in research and lacking empirical data. The main question this study was intended to answer is: what are the critical success factors in commercial open source projects?

The theory shows that there are many different factors present in the course of an open source software project. According to theories, several factors influence the performance and successfulness of the project, and those factors are very often ones related directly to the open source nature of the business. They are factors that derive from the community management, technology management, project management and market management. The factors are specific to software industry and open innovation; some apply to business in general.

The interviews revealed that there is basically no difference at all between open source software industry success factors and general success factors. The two factors that were most clearly success factors were *establishing and communicating hierarchy* and *effective market creation,* both of which are general and not so much even software related. The interviewees did mention – but not stressed – a few software related factors having importance in success, but they also claimed that the recipe for success is either the same as in proprietary software business or the same as in business in general. Thus it can be concluded that the theory conflicts with this study in the part that there is a special set of success factors for a specific industry of open source software business, or even software business in general. Still, generalization cannot be made over all business sectors. The study suggests that the success factors are the same regardless of type of software the company is doing business with.

The interviewees mentioned three factors that make a business successful: *low price, good product* and *good business model development*. The factors are extremely general, and very logical. They can be made specific to most businesses by little fine-tuning, so if taken a certain angle to all the tree factors, they can be seen as OS specific. Low price is traditionally the first thing that is expected when thinking of OSS, although discussed earlier that free as related to OS means free-of-use, not free-of-charge. Good product in OSS is perhaps critical in a little bit different sense, since the background study has shown, the management of the product is different from proprietary software and requires more careful – or different kind of – consideration. Business models are very closely attached to the CSF in this research, since the business models in OSS are extremely different from those of conventional business (how to make money with something that is free?).

It is also possible that theory is more powerful than practice. It is possible that the managers do not see CSF identification as important as they should based on literature. Or maybe CSF identification in too narrow a sector *is* useless. As Robert Young, the CEO of Red Hat says:

> *In fact you make money in free software exactly the same way you do it in proprietary software: by building a great product, marketing it with skill and imagination, looking after your customers, and thereby building a brand that stands for quality and customer service. (Young 1999)*

Critical success factor as a concept is created to help managers focus on a few critical areas in their business and prioritize in a way that maximizes the company's success (cf. Rockart 1979). When the manager knows the CSF in that particular industry in that particular company, he/she can base decisions on issues that matter in regards of the outcome, and he/she is able to manage more effectively.

This study can help managers realize that there are OS specific factors that are not unambiguously creating success, but instead create controversy among informants that are from different kinds of organizations and with different technologies. The factors, that exist in literature and were synthesized in the end of the literature review in this study, might depend on the specific environment of the company. Although the findings presented indicate those OS specific factors make no difference, it is important to be aware of the different factors present in the business, but make one's own conclusions on which are the specific key factors in one's own external and internal environment.

This study suggests that general success factors are the ones that apply in open source software industry. Thus good general management is enough for any manager and there is no need to be an expert in open source in order to lead the company to success. Rather, there is a need for a strong manager that has a solid background in several different fields and that way learned the best management style that is not too narrowed down to any specific field.

6.2 Suggestion for future research and limitations of the study

It would be interesting to see if the results would be the same if a similar research setting would be used. As qualitative research method permits the repetition factor of trustworthiness yielding different results The use of different interviewer and interviewees in a different point of time when the industry has developed further might change the outcome.

Another way of doing the research could be using triangulation, i.e. using two empirical methods in addition to literature review. The scope of this study did not permit additional quantitative study, but making such based on qualitative research results, could prevent the limitations of qualitative study what comes to generalization (see the trustworthiness chapter), and diminish the subjective nature of the research, which could yield more precise results.

The three general factors mentioned by the interviewees: low price, good product and good business model development deserve a closer look, especially focusing on different types of business. Are the factors different in different fields, and merely generalized by the interviewees? The outcome of this research gives the general critical success factors in open source software business management, but maybe they can be iterated to have specific meaning in specific industries.

7 SUMMARY

Open source and especially open source software has been gaining foothold during the past decade. Some authors even argue that a complete change towards openness in all sectors of business is inevitable. The software business in any case has gone through tremendous change from the traditional strong copyright model towards open source business models. This has lead to new revenue creation models that can be assumed to change the management style and requirements. This study aims to find answers to the following questions:

- What are the factors influencing commercial open source software projects?
- What is the relation between those factors?
- Why certain factors explain more of the success than other factors?

The theoretical part of the study explains the background for the research. It helps the reader to understand the concepts needed for the empirical part of the research. Important phenomenon is the paradigm shift from closed innovation to open innovation, which is the basic idea for open source software. Open innovation is an initiative that can be related to any field of business or research. It implies that the resources should be used completely, so that they can be exploited also in the external environment, instead of being guarded internally.

Open source is one manifestation of open innovation. Open source can also be other than software related, but this study concentrates on the software business only. Open source means that the source, whether it being software, knowledge or content, should be free, specifically free-of-use, not necessarily free-of-charge. The openness of the source enables the external parties benefiting and further developing the original version.

Open source software is an embodiment of open innovation and open source. It is software where the source code is open and freely available for modification and re-use. The degree of the freedom depends on the specific software license of which some are stricter and some more allowing. The main difference between a proprietary software license and an open source software license is that proprietary license is aimed to secure the code for the developing organization itself, and open source license exists for securing the freedom of the code for everyone.

Critical success factors define the focus that should be taken when managing a business or a project. The factors regarding open source software are divided into community management, technology management, project management and market management. The four main factors are divided into more specific factors that derive from the theories presented in this study, constructed directly from other authors and indirectly from challenges in open innovation.

Empirical study shows, that none of the open source specific factors are important in open source projects. Instead the findings show that open source software should be managed just as proprietary software business or any other type of business. Three basic critical success factors that were mentioned were low price, good product and good business model development, which were not included in the synthesis, where different factors deriving from literature were presented. Factors in the synthesis that had meaning were establishing and communicating hierarchy and good market creation, which are also quite general. In sum, the theory somewhat conflicts with the empirical findings as it seems that critical success factors in open source software projects are the same as in any other business model.

8 REFERENCES

Aarts, Emile (2005). Ambient Intelligence Drives Open Innovation. *Interactions*. Vol. 12 (4), 66-68.

Albaum, Gerald – Strandskov, Jesper – Duerr, Edwin (2002) *International Marketing and Export Management*. Pearson Education Limited.

Alvesson, Mats (1996) Leadership Studies: From Procedure and abstraction to reflexivity and situation. *Leadership Quarterly*. Vol. 7(4), 455-485.

Amit, Raphael – Zott, Christoph (2001). Value creation in E-business. *Strategic Management Journal.* Vol. 22 (6/7), 493-520.

Arce, Enrique – Flynn, Donal (1997) A CASE tool to support critical success factors analysis in IT planning and requirements determination. *Information and software technology.* Vol. 39 (5), 311-321.

Benussi, Lorenzo (2006) *The evolution of free/libre open source software*. <http://opensource.mit.edu/online_papers.php?lim=1000>, retrieved 29.9.2007.

Bonaccorsi, Andrea – Rossi, Cristina – Giannangeli, Silvia (2004) *Adaptive Entry Strategies Under Dominant Standards. Hybrid Business Models in the Open Source Software Industry*. SSRN Working Paper.

Bretthauer, David (2001) *Open Source Software: A History*. University of Connecticut.

Bruno, Albert V. – Leidecker, Joel K. (1984) Identifying and Using Critical Success Factors. *Long Range Planning*. Vol. 17 (1), 23-32.

Chakravarthy, Balaji S. (1986) Measuring strategic performance. *Strategic Management Journal*. Vol. 7 (5), 437–458.

Chesbrough, Henry W. (2003a) The Era of Open Innovation. *MIT Sloan Management Review*. Vol. 44 (3), 34-41.

Chesbrough, Henry W. (2003b) *Open Innovation: The New Imperative for Creating and Profiting from Technology.* Harvard Business School.

Chesbrough, Henry W. (2004) Managing Open Innovation. *Research Technology Management*. Vol. 47 (1), 23-26.

Clarke, Roger (2005) Research Models and Methodologies. HDR Seminar Series. <http://www.uow.edu.au/content/groups/public/@web/@commerce/documents/doc/uow012042.pdf>, retrieved 27.4.2012.

Conger, Jay (1998) Qualitative Research as the cornerstone methodology for understanding leadership. *Leadership Quarterly*. Vol. 9 (1), 107-121

Creswell, John (2003) *Research Design: Qualitative, Quantitative and Mixed Methods Approaches.* Thousand Oaks, CA.

Crowston, Kevin – Howison, James – Annabi, Hala (2006) Information systems success in free and open source software development: Theory and measures. *Software Process—Improvement and Practice*. Vol. 11 (2), 123–148.

Daffara, Carlo (2007) *Business models in FLOSS-based companies*. <http://opensource.mit.edu/online_papers.php?lim=1000>, retrieved 29.9.2007.

Davidsson, Per – Steffens, Paul – Fitzsimmons, Jason (2009) Performance Configurations Over Time: Implications for Growth- and Profit-Oriented Strategies. *Entrepreneurship Theory and Practice*. Vol. 33 (1), 125–148.

DeLone, William – McLean, Ephraim (1992) Information Systems Success: The Quest for the Dependent Variable. *Information Systems Research*. Vol. 3 (1), 60-95.

DeLone, William – McLean, Ephraim (2002) *Information Systems Success Revisited*. In: Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS 02), 238-249.

DeLone, William – McLean, Ephraim (2003) The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. *Journal of Management Information Systems.* Vol. 19 (4), 9-30.

Dess, Gregory G. – Robinson, Richard B. (1984) Measuring Organizational Performance in the Absence of Objective Measures: The Case of the Privately-Held Firm and Conglomerate Business Unit. *Strategic Management Journal*. Vol. 5 (3), 265-285.

Edmunds, Stahrl W. (1982) *Performance Measures for Growing Businesses*. Van Nostrand Reinhold Company.

Ellegaard, Charlotte – Grunert, Klaus G. (1993) The Concept of Key Success Factors: Theory and Method. *Perspectives on Marketing Management.* Vol. 3, 245-274.

Esteves, José Manuel (2004) *Definition and Analysis of Critical Success Factors of ERP Implementation Projects*. Universitat Politècnica de Catalunya, Barcelona, Spain.

Field, Tom (1997). When bad things happen to good projects. *CIO.* 15 Oct., 55-62.

Free Software Foundation. <www.gnu.org>, retrieved 22.4.2011.

Freund, York P. (1988) Planner’s Guide: Critical Success Factors. *Planning Review*. Vol. 16 (4), 20-23.

Gassmann, Oliver and Enkel, Ellen (2004) *Towards a Theory of Open Innovation: Three Core Process Archetypes*. <http://www.alexandria.unisg.ch/publications/274>, retrieved 30.8.2007.

GNU. <http://www.gnu.org/philosophy/free-sw.html>, retrieved 1.10.2007.

Goldman, Ron – Richard, P. Gabriel (2005) *Innovation Happens Elsewhere - Open Source as Business Strategy*. Morgan Kaufmann, Boston.

Gordjin, Jaap – Akkermans, Hans (2003) *Value Based Requirements Engineering: Exploring Innovative e-Commerce Ideas*. Vrije Universitet.

Grewal, Rajdeep – Lilien, Gary L. – Mallapragada, Girish (2006) Location, Location, Location: How Network Embeddedness Affects Project Success in Open Source Systems. *Management Science*. Vol. 52 (7), 1043–1056.

Hagel, John III – Brown, John Seely (2006) *Creation Nets: Harnessing the Potential of Open Innovation.* <http://www.johnhagel.com/creationnets.pdf>, retrieved 1.4.2008.

Halcomb, Elizabeth – Davidson, Patricia (2005) Is verbatim transcription of interview data always necessary? *Applied nursing research.* Vol. 19 (1), 38-42.

Hamel, Gary (2000) *Leading the Revolution*. Harvard Business School.

Hars, Alexander – Ou, Shaosong (2001) *Working for Free? – Motivations of Participating in Open Source Projects.* Proceedings of the Hawall International Conference on System Sciences.

Hedman, Jonas – Kalling, Thomas (2003) The business model concept: theoretical underpinnings and empirical illustrations. *European Journal of information Systems.* Vol. 12 (1), 45-59.

Heinonen, Laura (2009) *On the pursuit of growth in technology-based companies : the role of public financing in the start-up process of Finnish drug development companies*. Turku School of Economics.

Helander, Nina – Aaltonen, Timo – Mikkonen, Teemu – Oksanen, Ville – Puhakka, Mikko – Seppänen, Marko – Vadén, Tere – Vainio, Niklas (2007) *Open Source Software Management Framework*. eBRC Research Reports 38, Tampere.

Hendersson, Rebecca – Cockburn, Iain (1994) Measuring competence? Exploring Firm Effects in Pharmaceutical Research. *Strategic Management Journal*. Vol. 15 (8), 63-84.

Hirsjärvi, Sirkka – Hurme, Helena (2001) *Tutkimushaastattelu – Teemahaastattelun teoria ja käytäntö.* Yliopistopaino, Helsinki.

Hoch, Detlev J. – Roeding, Cyriac R. – Purkert, Gert – Lindner, Sandro K. (1999) *Secrets of Software Success*. Mc Kinsey & Comany Inc.

Hofer, Charles W. – Schendel, Dan (1978) *Strategy Formulation: Analytical Concepts*. St.Paul, West.

Hyvönen, Eero (2003) *Ohjelmistoliiketoiminta*. WSOY.

Jing-Helles, Jing (2008) *Service-based Open Source Business Models in the Chinese Market*. ServOSS-seminaari 3.6.2008, Helsinki.

Jonker, Jacobus (2004) *The Strategic Identification and Integration of Critical Success Factors to Achieve International Competitiveness for South Africa as a Tourism Destination*. University of Pretoria.

Joutsen, Atte (2007) Presentation at MindTrek-conference 3.10.2007.

Juga, Jari – Uotila, Tuomo (1993) *Success Factors of the Sawmilling Industry in Finland*. Turku School of Economics and Business Administration.

Kerlinger, Fred (1973) *Foundations of Behavioral Research*. Holt, Rinehart and Winston. New York:

Khandewal, Vijay – Ferguson, Jeff (1999) *Critical Success Factors (CSF) and the Growth of IT in Selected Geographic Regions*. Hawaii International Conference on System Sciences.

Koenig, Joel (2005) Seven open source business strategies for competitive advantage. IT *Manager's Journal*, May 13, 2004. <http://www.itmanagersjournal.com/articles/314?tid=85>, retrieved 4.2.2009.

Koski, Heli (2006) Do Open Innovation Models Boost Software Firms' Productivity? In: *Essays on OSS practices and sustainability*, ed. by Helander Nina - Antikainen Maria, eBRC Research Reports 36, Tampere.

Lakhani, Karim R. - von Hippel, Eric (2002) *How open source software works: ”free” user-to-user assistance*. MIT Sloan School of Management.

Lerner, Josh – Tirole, Jean (2000) *The Simple Economics of Open Source*. National Bureau of Economic Research.

Liang, Chaucer (1999) Critical Success Factors for reinventing the academic library. *Reference Services Review*. Vol. 27 (2), 127-133.

Lincoln, Yvonna – Guba, Egon (1985) *Naturalistic Inquiry*. Sage, CA.

Locke, John (2004) *Open Source Solutions For Small Business Problems*. Charles River Media.

Lynch, Richard (2000) *Corporate Strategy*. Prentice Hall, London.

Lytras, Miltiadis D. ed. (2006) *Open Source for Knowledge and Learning Management: Strategies Beyond Tools*. Hershey, PA, USA: Idea Group Publishing.

Midha, Vishal – Palvia, Prashant (2012) Factors affecting the success of open source software. *Journal of Systems and Software*. Vol. 85 (4), 895–905.

Morgan, Gareth – Smircich, Linda (1980) The Case for Qualitative Research. *Academy of Management. The Academy of Management Review.* Vol. 5 (4), 491-500.

Nasir, Mohd Hairul Nizam – Sahibuddin, Shamsul (2011) Critical success factors for software projects: A comparative study. *Scientific Research and Essays*. Vol. 6 (10), 2174-2186.

Odendahl, Teresa – Shaw, Aileen (2002) Interviewing Elites. In: *Handbook of interview research: context & method*, ed. by Gubrium Jaber – Holstein James. Thousand Oaks, CA, 299-315.

Oksanen, Ville – Laine, Juha (2006a) Software Patents and Open Source – How to Mix Water and Oil? In: *Essays on OSS practices and sustainability*, ed. by Helander Nina - Antikainen Maria, eBRC Research Reports 36, Tampere.

Oksanen, Ville – Välimäki, Mikko – Laine Juha (2006b) Open Source Risk Mitigation and the License Compliance Software Tool. In: *Empirical Insights on Open Source Software Business*, ed. by Helander Nina – Mäntymäki Maria, eBRC Research Reports 34, Tampere.

Open Source Initiative. <http://www.opensource.org>, retrieved 22.4.2011.

Open Science Resources. <http://www.openscienceresources.eu>, retrieved 1.4.2012.

Osterwalder, Alexander (2004) *The Business Model Ontology: A Proposition in a design Science Approach.* Université de Lausanne.

Osterwalder, Alexander – Pigneur, Yves – Tucci, Christopher L. (2005) Clarifying Business Models: Origins, Present, and Futire of the concept. *Communications of the Associations for Information Systems.* Vol. 16 (1), 1-25.

Padgett, Deborah (1998) *Qualitative Methods in Social Work Research: Challenges and Rewards.* Thousand Oaks, Sage.

Peng, Gang (2009) Critical Success Factors for Open Source Innovation: The Case of Open Source Software Development. *Issues in Information Systems*. Vol. 10 (2), 157-164.

Peters, Tom – Waterman, Robert (1982) *In search of Excellence: Lesson's from America's Best Run Companies*. Harper & Row.

Pigneur, Yves (2004) *An ontology for defining e-business models*. UBC workshop, Jan 9, 2004. <http://www.slideworld.org/viewslides.aspx/PPT-An-ontology-for-defining-e-business-models-Slideworld-Medical-Search-Engines-ppt-214757>, retrieved 25.4.2011.

Pinto, Jeffrey – Prescott, John (1988) Variations in Critical Success Factors Over the Stages in the Project Life Cycle. *Journal of Management*. Vol. 14 (1), 5-18.

Project Open Source / Open Access. University of Toronto. <http://open.utoronto.ca/index.php?option=com_content&task=blogcategory&id=28&Itemid=58>, retrieved 28.9.2007.

Rajala, Risto – Nissilä, Jussi – Westerlund, Mika (2007). Revenue Models in the Open Source Software Business. In: *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*, ed. by St.Amant Kirk – Still Brian, Information Science Reference, New York.

Raymond, Eric S. (1998) *Goodbye, "free software"; hello, "open source"*. <http://www.catb.org/~esr/open-source.html>, retrieved 22.4.2011.

Raymond, Eric S. (1999) *Keeping an Open Mind*. <http://www.catb.org/~esr/writings/openmind.html>, retrieved 24.4.2011.

Raymond, Eric (2000) *The Cathedral and the Bazaar*. <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>, retrieved 22.4.2011.

Reel, John (1999) Critical Success Factors In Software Projects. *Software*. Vol. 16 (3), 18-23.

Rigby, Darrel – Zook, Chris (2002) Open-market Innovation. *Harvard Business Review*. Vol. 80 (10), 80-89.

Rockart, John (1979) Chief Executives Define Their Own Data Needs. *Harvard Business Review*. Vol. 57 (2), 81-93.

Rockart, John – van Bullen, Christine (ed.) (1986) *A Primer on Critical Success Factors. The Rise of Managerial Computing: The Best of the Center for Information Systems Research.es-Irwin, 1981*. McGraw-Hill School Education Group.

Rosen, Lawrence (2004) *Open Source Licensing: Software Freedom and Intellectual Property Law*. Prentice Hall.

Saastamoinen, Matti (2006) *Avoimen lähdekoodin lisenssit kaupallisessa liiketoiminnassa*. Tampereen Yliopisto.

Sen, Ravi – Singh, Siddhartha – Borle, Sharad (2012) Open source software success: Measures and analysis. *Decision Support Systems*. Vol. 52 (2), 364-372.

Seppä, Arto (2006a) *Open Source in Finnish Software Companies*. Elinkeinoelämän Tutkimuslaitos (ETLA). <http://www.etla.fi/files/1447_Dp1002.pdf>, retrieved 4.2.2009.

Seppä, Arto (2006b) Software Companies in Europe: Initial Look at the Importance of Patents and Licenses to Software Firms and Some Firm Characteristics Comparisons. In: *Empirical Insights on Open Source Software Business*, ed. by Helander Nina – Mäntymäki Maria, eBRC Research Reports 34, Tampere.

Seppänen, Arto (2010) *Ohjelmistokehityspalveluiden liiketoiminta- ja sopimusmallit*. Kajaanin ammattikorkeakoulu.

Seppänen, Marko – Mäkinen, Saku (2005) *Business Model Concepts: A review with Case Illustrations*. Engineering Management Conference, 2005. Proceedings. 2005 IEEE International. Vol. 1, 376-380.

Seppänen, Marko (2006) Thoughts on Competitive Strategy and OS. In: *Empirical Insights on Open Source Software Business*, ed. by Helander Nina – Mäntymäki Maria, eBRC Research Reports 34, Tampere.

Singh, Param Vir – Tan, Yong – Mookerjee, Vijay S. (2009) *Network Effects: the Influence of Structural Social Capital on Open Source Project Success*. CMU Tepper working paper.

Stallman, Richard (2010) *Free Software – Free Society*. Free Software Foundation Inc.

Stevens, Greg A. – Burley, James (1997) 3,000 raw ideas = 1 commercial success! *Research Technology Management.* Vol. 40 (3), 16-28.

Subramaniam, Chandrasekar – Sen, Ravi – Nelson, Matthew L. (2009) Determinants of Open Source Software Project Success: A Longitudinal Study. *Decision Support Systems*. Vol. 46 (2), 576-585.

Taylor-Powell, Ellen – Renner, Marcus (2003) *Analyzing qualitative data.* University of Wisconsin-Extension.

Tübke, Alexander (2004) *Success Factors of Corporate Spin-Offs*. Springer, United States.

Tähtinen, Jaana (2001) *The Dissolution Process of a Business Relationship*. University of Oulu.

Vainio, Niklas – Vadén, Tere – Oksanen, Ville – Seppänen, Marko (2006) Elements of Open Source Community Sustainability. In: *Essays on OSS practices and sustainability*, ed. by Helander Nina - Antikainen Maria, eBRC Research Reports 36, Tampere.

Viskari, Sari (2006) *Managing Technologies in Research Organization: Framework fo Research Surplus Portfolio*. Research report 176, Lappeenranta University of Technology.

Västilä, Tarja (2010) Täältä tullaan maailma! *Kauppalehti.* 18.10.2010, 12.

Ward, Ben – Lincoln, Tim (ed.) (1990) Planning for Profit. In: *Managing Information Systems for Profit*. John Wiley & Sons, 103-146.

Weber, Stephen (2004). The Success of Open Source. Harvard University Press.

West, Joel – Gallagher, Scott (2004) *Key Challenges of Open Innovation: Lessons from Open Source Software*. <http://www.cob.sjsu.edu/west_j/Papers/WestGallagher2004.pdf>, retrieved 30.8.2007.

West, Joel – Vanhaverbeke, Wim – Chesbrough, Henry (2006) *Open Innovation: A Research Agenda. In: Open Innovation: Researching a New Paradigm*. Ed. by Chesbrough Henry – Vanhaverbeke Wim – West Joel, Oxford University Press.

Wilkinson, Ian – Young, Louise (1994) Improvisation and Adaptation in International Business Research Interviews. In: *Qualitative Methods in Organizational Research: a Practical Guide,* ed. by Cassell Catherine – Symon Gillian. Sage, London, p. 207-223.

Wolf, Karsten D. (2007) Presentation at MindTrek Conference 3.10.2007.

Woods, Dan – Guliani, Gautam (2005) *Open Source for the Enterprise: Managing Risks, Reaping Rewards.* O'Reilly.

Young, Robert (1999) Giving it Away: How Red Hat Software Stumbled Across a New Economic Model and Helped Improve an Industry. *The Journal of Electronic Publishing,* Vol. 4 (3). <http://quod.lib.umich.edu/cgi/t/text/textidx?c=jep;view=text;rgn=main;idno=3336451.0004.304>, retrieved 19.2.2008.