

UTRECHT UNIVERSITY

MASTER BUSINESS INFORMATICS

THESIS REPORT

---

# A Comparative Case Analysis of Open Source Software Business Models.

The creation of a blueprint for Open Source Software start-ups.

---

*Author*

Zeena SPIJKERMAN, 5766664  
z.spijkerman@students.uu.nl

*Supervisor*

Dr. Slinger JANSEN  
slinger.jansen@uu.nl

*Second supervisor*

Dr. Fabiano DALPIAZ  
f.dalpiazz@uu.nl

July 11, 2018



## **Abstract**

A misconception about Open Source Software is that there is no such thing as an open source business model (Letellier, 2008). This misconception has endured for a long time because the history of the software industry has been driven by standardized proprietary software. Nowadays, more companies are generating considerable revenues through Open Source Software (OSS), and the research question we can extract from this is: how do businesses generate revenue with OSS? A comparative case analysis of 10 B2B OSS businesses in combination with a literature review and in-depth interviews will answer this research question. With this research we aim to determine the success factors of OSS business models based on empirical data by applying the Software Business Model Framework. The results are formed into a blueprint and guideline for OSS focused start-ups which lead them in the establishment of a successful OSS business. Additionally, investors can use the blueprint as a prediction of the future of businesses. The expert-interviewees have shown several similarities between the business models. Overall, we can confirm that the sample consists of commercial open source businesses which apply a hybrid combination of Software Business Model components.

**Keywords:** Open Source Software (OSS), Business Models, Software Business Model Framework, OSS start-ups.

# Table of contents

<b>Introduction</b>	<b>3</b>
<b>1 Research Approach</b>	<b>4</b>
1.1 Problem Statement . . . . .	4
1.2 Research Questions . . . . .	4
1.3 Research Strategy . . . . .	5
1.3.1 Literature study . . . . .	5
1.3.2 Case study . . . . .	5
1.3.3 Analysis . . . . .	7
1.3.4 Relevance . . . . .	8
<b>2 Business Model Analysis</b>	<b>9</b>
2.1 Business Models Defined . . . . .	9
2.2 Open Source Business Models . . . . .	10
2.2.1 Use-Value Models . . . . .	14
2.2.2 Indirect-Sale Value Models . . . . .	15
2.2.3 Direct-Sale Value Models . . . . .	16
2.3 Value Creation . . . . .	17
2.4 Open Source Licensing Policies . . . . .	19
2.5 Measurements of Success . . . . .	20
2.5.1 Business Model Success . . . . .	21
2.6 OSS Start-up Strategies . . . . .	22
<b>3 Case Study Analysis</b>	<b>24</b>
3.1 The Interviewees . . . . .	24
3.1.1 Incentives . . . . .	24
3.1.2 The Start-up Phase . . . . .	26
3.1.3 Suggestions for Entrepreneurs . . . . .	27
3.2 Comparison of OSS Business Models . . . . .	29
3.2.1 Strategy . . . . .	29
3.2.2 Revenue . . . . .	30
3.2.3 Upstream . . . . .	31
3.2.4 Downstream . . . . .	33
3.2.5 Usage . . . . .	34
<b>4 Findings</b>	<b>36</b>
<b>5 Discussion</b>	<b>40</b>
<b>6 Conclusion</b>	<b>42</b>
<b>References</b>	<b>43</b>
<b>Appendices</b>	<b>49</b>

<b>A Interview protocol</b>	<b>50</b>
<b>B Software Business Model Framework</b>	<b>52</b>
<b>C Software Business Model Framework components</b>	<b>54</b>
<b>D Open Source Software Licenses</b>	<b>56</b>

# Introduction

Starting from the mid-90s, several small open source businesses changed the software industry by offering a cooperatively produced collective good instead of proprietary software (Hippel & Krogh, 2003). Since then, there has been increasing interest among academics and practitioners in Open Source Software (OSS) (Lee et al., 2009). Initially created by the hacker movement (E. S. Raymond et al., 2000), the OSS phenomenon has now metamorphosed into a more mainstream and commercially viable product (Fitzgerald, 2006) with ground rules defined by the Open Source Initiative (*The Open Source Definition*, 1999). When companies recognized this new type of software as a way to generate revenue, new business models arose. OSS was starting to be used as a new business strategy to reduce costs and make maximum profits by a large stream of software companies (Krishnamurthy, 2005). These companies became well-known by offering cooperatively produced software (Hippel & Krogh, 2003) for free within an already existing corporate market. Due to this cooperative approach to product development, open source is often not seen as a business approach but more as a technology model (Krishnamurthy, 2005). Nevertheless, different types of business models have been applied where different types of stakeholders and external factors are playing a role in the start-up phase of OSS producers, distributors and service providers. Currently, companies are focused on the Open Source concept and create considerable revenues through open source software and services (Krishnamurthy, 2005) but with different offering profiles. An example is Red Hat and Linux who not directly make money from the open source programs but price complementary services (Lerner & Tirole, 2001). The OSS business model and its revenue logic are not always as obvious to perceive (Androutsellis-Theotokis et al., 2011) and there is little to no research performed that looked into the building blocks of successful OSS business models. A large amount of business literature is devoted to the definition of a business model, but this research specifically focuses on OSS businesses and will therefore use a more open source oriented definition of a business model and its characteristics.

Onetti & Capobianco (2005, p.224) recognize the lack of a comprehensive theoretical framework about OSS business models and that this is due to the relative newness of the phenomena. This paper contributes to the field by exploring the way open source businesses have entered the software ecosystem and created a foundation for following companies. Additionally, knowledge is contributed to the software business domain by looking specifically at the current business models of B2B focused OSS businesses. A comparative case study is conducted by reviewing the business models and factors for success of 10 OSS businesses. The business models are defined and conceptualized by application of the Software Business Model Framework of Schief (2013). The outcomes of a literature study around the subjects of OSS consortia, their business models, the value exchanges within them, and how success could be measured, are used for a thorough analysis of the determinants. These interviews with expert employees provide information about the success of these OSS businesses and will be used to create an OSS business model blueprint and a guideline for OSS start-ups. Chapter 1 provides details on the research approach. The literature review is performed in Chapter 2 comprising an overview of existing business model literature, definitions of business model success and start-up strategies. Chapter 3 is focused on the analysis of the case study. The findings section conclude with several statements on OSS business models and offers an blueprint for OSS start-ups. Finally this research finishes with the discussion and conclusion.

# Chapter 1

## Research Approach

### 1.1 Problem Statement

The body of knowledge in OSS research lacks focus on the building blocks constructing business models of successful OSS businesses. This research tries to establish the determinants that make OSS businesses thrive by looking at previous success-stories. In this research the definition of OSS businesses as described by Bonaccorsi et al. (2006) will be used: “firms that supply, in various ways, open source based products and services and release them under Open Source licenses.” This definition will simultaneously be used with the Open Source Definition, originally acquired from the Debian free Software Guidelines (see table 1.1). We aim to further investigate and build a foundation for both OSS and business literature. Literature based on empirical data focusing on OSS entrepreneurs is scarce. This leaves this vulnerable group of starters without sufficient guidelines while entering an upcoming market.

### 1.2 Research Questions

This paper addresses the current OSS landscape where the focus remains on the business models of OSS businesses and therefore the main research question will be: **Q.1. What are the success determinants for open source software business models?** To answer the first research question, this research consists of two components:

#### 1. The Open Source market

This first component shows how the open source market is currently defined. Background stories of the B2B OSS businesses create a foundation for the rest of the research. From there on, the comparison of the successful OSS Businesses is initiated. To perform a comparative case analysis a classification of business models is needed. Each business model is defined by using the Software Business Model Framework (Schief, 2013). This “[..] comprehensive, industry-specific and standardized software business model framework” (Schief, 2013, p.61) allows us to categorize and define the variety of OSS business models (see section 1.3.2.3). The framework will be elaborated on for every OSS business to create a comparable set to analyze.

#### 2. Determinants of success

The characteristics that are being analyzed will be defined in this section and are part of the method. The first question is answered by analyzing the outcomes of the first component and subsequently used to create the guideline for entrepreneurs. To prove which exact combination of characteristics has been the key to success for OSS businesses, the following question is answered: **Q.2 What synergies between business model characteristics create a blueprint for business success?**

The answer to this question is based on the outcomes of the expert-interviews and OSS business model focused literature. The outcomes are gathered to create a guideline for new business to use

and is simultaneously applied as validation of the research. The following research question will generate an artifact usable for example investors and OSS developers: **Q.3. Which factors lead to potential business success for an open source business?** The artifact is presented in a blueprint usable by practitioners and is a foundation for further research.

## 1.3 Research Strategy

For this qualitative research, a multiple-case study is selected because it enables the exploration of phenomena within, in this case, the open source ecosystem (Baxter & Jack, 2008). The use of a variety of data sources ensures that the unknown field is explored to eventually determine the explicit success determinants of OSS business models. The primary data used for this research is collected from a comparative case study which and is backed by a literature study. Subsequently, internal validation is performed by the interviewees.

### 1.3.1 Literature study

Yin (2009) states that before a case study is performed in the field, initial theory and understanding of the theory behind what is being studied is necessary. It is known that "A thorough, sophisticated literature review is the foundation and inspiration for substantial, useful research" (Boote & Beile, 2005, p.3). Therefore this research starts with an enhanced literature study to gain knowledge about Open Source Businesses, their used business models, and possible success factors. The literature study also gives an idea about the determinants of success. The literature review is presented in Chapter 2 providing a business model analysis in a tabular literature overview covering OSS business models. This same chapter contains the e3 value models which depicts the value exchanges between the different stakeholders interacting around an OSS product. Additionally Chapter 2 depicts on how the success of OSS business models could be measured and which components of the model should be taken into consideration in the determination of success-factors. Finally, start-up strategies are discussed to finally be incorporated in the OSS guideline.

### 1.3.2 Case study

The literature study is combined with a comparative case study which grounds the current dynamics that are present within predefined settings (Eisenhardt, 1989). The source of evidence is based on individual depth interviews (Yin, 2009) within the sample of OSS companies. The final determinants of success rest on a comparative case analysis of the interview transcripts of 10 OSS companies which are chosen based on pre-determined sample criteria. The case study approach is based on the three phases of the Case Study Protocol(CSP) constituted by Pervan & Maimbo (2005) which is based on research by Eisenhardt (1989). The authors describe the CSP as a guideline for data analysis containing the procedures for conducting research and is also used as a research instrument (see Figure 1.1).

#### 1.3.2.1 Sample Criteria

The second stage of phase one of the CSP depicts the selection of the cases where a specified population is defined. Therefore in this section we have pre-defined sample criteria for the selection of the cases. For case studies to give significant results, random selection of the sample is neither necessary, nor preferable (Eisenhardt, 1989). The size of the sample for this research is controlled by theoretical and practical considerations (Robinson, 2014). Theoretically, the size of the sample influences the generalizability of the research, therefore a big sample size (around 20) is preferable. Practically, by convenience sampling the sample size is smaller. This is due to response time of the open source companies and the number of interviews that have to be performed within the time constraint of this research. The sample consists of OSS businesses that are chosen based on particular search criteria. We are interested in companies that follow a certain quality standard in their business and share the interest in OSS. The OSS company that fits within the sample;

Phase	Stage	Activity	Reason
Phase One	Getting started	<ul style="list-style-type: none"> <li>Define of research question(s)</li> <li>Possible a priori constructs</li> </ul>	<ul style="list-style-type: none"> <li>Focuses efforts</li> <li>Provides better grounding of measures</li> </ul>
	Selecting cases	<ul style="list-style-type: none"> <li>Neither theory nor hypothesis</li> <li>Specified population</li> <li>Theoretical sampling</li> </ul>	<ul style="list-style-type: none"> <li>Retains theoretical flexibility</li> <li>Constrains extraneous variation and sharpens external validity</li> </ul>
	Crafting instruments and protocols	<ul style="list-style-type: none"> <li>Multiple data collection method</li> <li>Qualitative and quantitative</li> </ul>	<ul style="list-style-type: none"> <li>Triangulation strengthens grounding of theory</li> </ul>
Phase Two	Entering the field	<ul style="list-style-type: none"> <li>Iterative data collection and analysis</li> <li>Flexible opportunistic data collection</li> </ul>	<ul style="list-style-type: none"> <li>Speeds analysis</li> <li>Facilitates emergent themes</li> </ul>
	Analysing the data	<ul style="list-style-type: none"> <li>Within-case</li> <li>Cross case analysis</li> </ul>	<ul style="list-style-type: none"> <li>Gains familiarity with data and preliminary theory generation</li> <li>Looks beyond initial impressions</li> </ul>
	Shaping hypothesis	<ul style="list-style-type: none"> <li>Iterative tabulation of evidence for each construct</li> <li>Replication logic across cases</li> <li>Search for the cause, ie the 'why' behind relationships</li> </ul>	<ul style="list-style-type: none"> <li>Sharpens construct definition, validity and measurability</li> <li>Confirms, extends and sharpens theory</li> <li>Builds internal validity</li> </ul>
Phase Three	Enfolding the literature	<ul style="list-style-type: none"> <li>Comparison with conflicting literature</li> <li>Comparison with similar literature</li> </ul>	<ul style="list-style-type: none"> <li>Builds internal validity, raises theoretical level and sharpens construct definition</li> <li>Improves generalisability, raises theoretical level and sharpens construct definition</li> </ul>
	Reaching closure	Theoretical saturation	Ends process when marginal improvement becomes small.

Figure 1.1: Case Study Protocol by Pervan & Maimbo (2005) adapted from Eisenhardt (1989)

1. is registered as a company,
2. is B2B; meaning the business is providing OSS to other businesses,
3. is a software vendor; meaning it creates and offers (open source) software,
4. made the code of the software freely available,
5. hosts an open source community,
6. generates revenue.

In order to develop software under the Open Source name there are some requirements as mentioned in the Open Source Definition as stated in table 1.1 (*The Open Source Definition*, 1999). This means that besides the company criteria chosen by the authors, the companies should follow the rules set by The Open Source Definition. Following these criteria, the sample consists of the following companies: Project Open, Mulesoft, SUSE®, Prism, GitLab, GeoCat B.V., Typo3 GmbH, xTuple, PowerDNS and Red Hat.

### 1.3.2.2 The Interview

Phase two of the CSP characterizes the iterative data collection and analysis, which in this research starts by conducting interviews within the sample. The expert-interviews are a combination of open questions and the predefined elements from the Software Business Model Framework of Schief & Buxmann (2012). The interviews are semi-structured and held with practitioners in the OSS field. The participants of the interviews are chosen because of their experiences which reflects the scope (Cooper et al., 2006) and their ability to answer the interview questions. The interview comprises two parts. The first part is based on 10 open questions (see Appendix A.) focused on the background of the company, the entrepreneurial aspects and the interviewees' view on success. These answers provide background information about the OSS company and is used to create a guideline for future OSS entrepreneurs. The open part of the interview was also used to give extra feedback on the questions and the background story of the interviewee and his role. The second part of the interview is based



The Open Source Definition	
1	Free redistribution
2	Source code
3	Derived works
4	Integrity of The Author's Source Code
5	No Discrimination Against Persons or Groups
6	No Discrimination Against Fields of Endeavor
7	Distribution of License
8	License Must Not Be Specific to a Product
9	License Must Not Restrict Other Software
10	License Must Be Technology-Neutral

Table 1.1: Open source criteria generated from *The Open Source Definition* (1999)

on the work of Schief & Buxmann (2012) and gives insight in the particular characteristics of the analyzed business models.

### 1.3.2.3 Software Business Model Framework

A method used to define the characteristics of software business models is the Software Business Model Framework (SBMF) (Schief, 2013). Schief conceptualizes business models used in the software industry and describe it in a standardized manner, gathered in a framework. The author recognizes a literature gap in the field of business model concepts specifically in the software industry and mentions that the existing concepts are limited in scope compared to the generic ones. The proposed framework is trying to fill this gap but does not account for all the software industry-specific economic properties. In their research, Schief & Buxmann (2012) state that a business model is composed of a number of strategy elements, and that their model make the strategic choices explicit. An extensive literature research resulted in 213 conceptual connections between economic properties and affiliated software business model components. Nevertheless, their Business Model Framework was based on the three studies of Osterwalder (2004), Morris et al. (2005) and Krumeich et al. (2012). The concept of the business model ontology by Osterwalder (2004) is well spread in the academic world and therefore used as a base for their framework. The other two studies are used for their literature reviews on various business model concepts. The Software Business Model Framework is composed of 5 groups that in turn contain 20 elements that are recommended as guidelines to characterize a business model (Schief & Buxmann, 2012). The 5 groups are based on an extensive literature research and come together as: strategy, revenue, upstream, downstream and usage. Figure C.1 in the appendix gives an overview of the groups and their sub-components. The elements have been empirically validated with the outcomes of expert interviews with managers of 9 software companies in Germany. The goal was to prove the comprehensiveness of the framework and therefore the companies were chosen due to the similarity of their business models (Schief & Buxmann, 2012). The framework will be used to analyze and perform the comparative case analysis. The use of this framework enables us to compare the business models of OSS companies on the same level and define the determinants of success. Ideas of what these determinants might look like, can be found in Chapter 3. The SBMF will be used as a base for the practical part of the research. This means that the questions of the questionnaire are based on the before mentioned and empirically validated 5 building blocks of the framework. This framework is used because of its ability to classify, structure and analyze any given software business model (Schief, 2013).

### 1.3.3 Analysis

The second stage of phase two of the CSP is the analysis of data within- and cross-case. The interviews are recorded and transcribed to eventually be analyzed with the NVivo tool (see *What is NVivo?*

for more information). The answers of the stakeholders are coded within the tool based on the categorized questions and SBMF components. With the use of this research method, qualitative rigor is what is aimed for. Rigor is needed to establish four components as stated by Lincoln and Guba in research by Thomas & Magilvy (2011): Credibility, transferability, dependability and confirmability. These four components contribute to a detailed research and the ability to replicate it (Thomas & Magilvy, 2011). This comparative case study is based on the comparison of the completed SBMFs by assembling all of them in a single table. We use the tree-map visualization technique to efficiently use the display space while mapping the full hierarchy of the chosen components (Johnson & Shneiderman, 1991). The tree-maps create the foundation of an overview of similarities and differences and will be combined with the coded answers extracted from the interviews. All-together the data is analyzed to derive a blueprint for Open Source start-ups. The data is analyzed to perform the third step of the CSP where the findings are used to sharpen the construct definitions. Moreover, in this stage the data is internally validated by the interviewees. Phase three of the CSP is finalized by referencing to existing literature and closure of the research.

### **1.3.4 Relevance**

The combination of interviews and a literature study produces qualitative data that is used to model the determinants of success. The guideline is used as a rule of thumb of the actual determinants and can be applied by Open Source start-ups. Gagliardi et al. (2011) determine the usefulness of guidelines as "the syntheses of best available evidence that support decision making" and is also seen as a way to "promote proven benefit" (Grimshaw et al., 2004, p.2). The guideline can be used to determine the propensity for growth of OSS business models of entrepreneurs and can be used to consider alterations in these business models. The guideline is created according to the sections from Schief and Buxmann's 2012 Software Business Model Framework (Figure B.1) to cover each part of software business models. Lambert & Davidson (2013) see that most studies collect data to understand the business model concept, but see that there is no empirical research that aims to predict firm success. This study adds empirical data which can be applied by investors to see whether business models are well constructed and lead to success. Additionally, inside expert information reveals information and tips that have yet not been collected. Finally, the data extracted from the case study can be compared and reviewed composing a blueprint for Open Source start-ups. This blueprint adds to business model theory and is applicable by OSS practitioners.

## Chapter 2

# Business Model Analysis

### 2.1 Business Models Defined

Business models have been in the spotlight of academics and practitioners for a long time, at least 1177 articles have been published since 1995 (Zott et al., 2011; Burkhart et al., 2011). Because of this large amount of research, it seems impossible for academics to create one unified definition of a business model (Morris et al., 2005; Shafer et al., 2005; Zott et al., 2011; Burkhart et al., 2011). Research seems to use the term in a way that it particularly fits the purpose of the studies (Zott et al., 2011). The business model concept is generally used as a management method to envision and analyze existing business ideas and to plan future decisions (Burkhart et al., 2011; Shafer et al., 2005). Innovative business models can disrupt a market, and continuous revision is needed to become successful (Schief, 2013). According to Casadesus-Masanell & Ricart (2010, p.205) a business' strategy coincides with its business model, and the business model appears as a reflection of its realized strategy. The authors state that an observer should be able to know the business' strategy by looking at its business model. This shows that a business model is a powerful tool for a wide range of disciplines (Shafer et al., 2005) and is used in this study to grasp the revenue streams of OSS businesses, contributing to specific software business model literature. The most recent literature review on the concept of business models is performed by Zott et al. (2011). In their research the authors found that the existing definitions partially overlap, and that researchers find a great amount of interpretations. Their review concluded that three phenomena are mostly used to describe business models (p.15): 1. e-business and the use of information technology in organizations; 2. strategic issues, such as value creation, competitive advantage, and firm performance; and 3. innovation and technology management. A consensus is leaning towards a definition that contains value proposition and creation as the core of a business model (Schief, 2013). Therefore this study uses the definition that contains this component stated by Osterwalder (2004):

A business model is a conceptual tool that contains a set of elements and their relationships and allows expressing a company's logic of earning money. It is a description of the value a company offers to one or several segments of customers and the architecture of the firm and its network of partners for creating, marketing and delivering this value and relationship capital, in order to generate profitable and sustainable revenue streams. (p.15)

The Software Business Model Framework by Schief (2013) will be used as a method to analyze the OSS businesses and therefore it is important to decompose the definition of Osterwalder (2004) into the components of the framework. The literature review performed by Schief (2013) indicates value as the center of gravity in a software business model. The definitions found and analyzed for their literature review are not industry-specific, therefore Schief (2013) addresses four key areas of business models found in literature and specifically used for the software industry: value proposition, financials, value configuration and customers. The combination of these four areas form a foundation of software

business models and leverage as the set of elements expressing a company's logic of earning money (Osterwalder, 2004, p.15).

## 2.2 Open Source Business Models

Despite the recognition of software business models, there is still a misconception around OSS stating that there is no business model specifically for OSS vendors (Letellier, 2008, p.6). In addition to that, Open Source is often wrongly categorized as a solely cooperative technology approach to product development instead of a business approach (Krishnamurthy, 2005). Nevertheless, there is enough evidence that Open Source has evolved into a real business approach. Because OSS is distributed freely, conventional ways of generating revenue need to be adjusted. Making money with free software may not be easy, but the challenge is not bigger than with proprietary software (Young, 1999). Businesses are constrained to develop new business models but not all OSS products are on the same level of profit potential (Bonaccorsi et al., 2006). We have seen that in the traditional software business models, companies provide all of the value to the customers and the revenue is realized in return through traditional license fees (Hecker, 1999). Hecker states that in OSS business models, the value is provided by different stakeholders like developers that are attracted to working on a project. Therefore, OSS demands different business models that are able to leverage the potential of the open source product (Onetti & Verma, 2008). The interest in the different types of business models for OSS has grown, but this field has not yet been completely clarified (Onetti & Verma, 2008). There have been a lot of attempts to classify and gather all types of OSS business models (Onetti & Verma, 2008), but there is still no consensus. Rosenfall (2012) attempted to create a literature overview of OSS business models categorized based on the literature of E. Raymond (1999). Because there are many ways to generate revenue with OSS, an overview with a simple taxonomy of three components as used by Rosenfall (2012) is mostly accessible. E. Raymond (1999, p.2) makes a distinction between two kinds of economic value software has;

- Use value; the use value of a program is its economic value as a tool, a productivity multiplier, works as a non-profit intermediate good.
- Sale value; the program's value as a salable commodity, the value as a final for-profit good.

Both values distinguish several business models, two under the category "use value", five known and two speculative under the "sale value" category. Figure 2.1 presents an overview of the OSS business models Raymond depict in his groundbreaking research. Based on Rosenfall's 2012 literature review we have found additional work of researchers depicting OSS business models. The business models from figures 2.2 and 2.3 show both differences and common characteristics. The following section will describe the common assumptions on OSS business models as shown in the literature overview.

Use-Value Funding Business Models		Example
Cost-sharing	Competing software users that cooperatively fund OSS development with pool programming resources.	Apache Web Server
Risk Spreading	Opening up software, lowers costs and spreads risks. The collaborative community is funded by multiple independent revenue streams.	Cisco
Indirect Sale-Value Business Models		
Loss-Leader/Market positioned	Using OSS to create/maintain a market position for proprietary software that is generating a direct revenue stream.	Netscape opening the Mozilla browser
Widget frosting	Hardware manufacturers opening up their software to create another profit center by making their drivers, configurations and utility tools free. "Future proofing" with OSS.	Apple opening Darwin (core of MacOSX)
Give away the recipe, open a restaurant	Open source software to create a market position for services.	Red Hat
Accessorizing	Sell accessories for OSS. At the low end, mugs and T-shirts; at the high end, professionally-edited and produced documentation.	O'Reilly & Associates
Free the future, sell the present	Open the proprietary software under certain conditions. Software is released in binaries and source with a closed license, but with one that includes an expiration date on the closure provision.	Aladdin Enterprises
Free the software, sell the brand ( <i>speculative</i> )	Open a software product, retaining a test suite and/or brand name and add services or third-party software (Rosenfall, 2002, p.53).	Sun Microsystems with OpenOffice
Free the software, sell the content ( <i>speculative</i> )	Brokerage, the client and server are open but the content is proprietary (Rosenfall, 2002, p.53).	N/a

Figure 2.1: Business models taxonomy according to E. Raymond (1999)

Author(s)	Use-Value Funding Models		Indirect- Sale Value Models							Direct-Sale Value Models			
Raymond (2001)	Cost-sharing	Risk-spreading	Loss-Leader/ Market Positioned	Widget Frosting	Give away the recipe, Open a restaurant	Accessorizing	Free the future, sell the present	Free the software, sell the brand	Free the software sell the content				
Hecker (1999)			Loss-Leader	Widget Frosting	Support seller (software franchising)	Accessorizing (software franchising)	Sell it, free it	Brand licensing (software franchising)	Service enabler	Dual licensing (Hybrid 1)	Restrict modifications (Hybrid 2)		
Lerner & Tirole (2002)				Code release	Symbiotic subsidising				Intermediates				
Stallman (2002)			Selling media, not the software		Consulting services					Niche and specialty OSS distributors			
Wichmann & Spiller (2002)	OSS development and Community enablers		Original Linux distributors	OSS distributors and complimentary product	Service and support providers	OSS conference organizers							
West (2003)				Partly open	Opening parts								
Dahländer (2005)	Sponsoring projects	Founding projects		Using projects						Using projects		Using projects	
Goldman and Gabriel (2005)				Bundle OSS with other software + hardware	Consulting services	Ancillary items	Newest version proprietary, previous OSS		Software with OSS as platform	OSS and proprietary for OEM		Subscription services	
Krishnamurthy (2005)			Distributor (1)		Distributor (2/3) + third party Service Provider				Third-party service provider (software & service)	Software producer (Non-GPL and GPL model)			
Perens (2005)	End-user business + contractors + Service businesses			Open source with proprietary accessories + hardware vendors	Open source + services					Mixed OSS and proprietary software	Proprietary Open Source		

Figure 2.2: Overview Business models taxonomy based on research of E. Raymond (1999) and Rosenfall (2012) part 1/2.

Author(s)	Use-Value Funding Models		Indirect- Sale Value Models							Direct-Sale Value Models			
Bonaccorsi et al. (2006)			Pure OSS business model								Hybrid business models		
West & Gallagher (2006)	Pooled R&D	Pooled R&D		Selling complements	Spinout				Donate complements				
Perr et al. (2010)	Community source/consortia	Consortia	Professional services/consulting	Device Hybrid/proprietary extensions	Support					Dual license	Hybrid/proprietary extensions	Subscription	Device
Daffara (2007)			Installation support	Badgeware	Consulting services		Training, single or multiple version offering; Split OSS/commercial products	Platform providing: selection, support, integration & services on set of projects	Integration	Twin licensing	Legal & technical certifications	Subscription	
Rönnback et al. (2010)		Risk-sharing							In-direct model (services)				
Chesbrough & Appleyard (2007)	Community source				Support, Professional services/consulting					Dual license	Proprietary extensions/add-ons	Subscription	Device
Fitzgerald (2006)	Leveraging community development		Loss leader		Value-added service enabling	Accessorizing		Leveraging the open-source brand		Dual licensing			
Koenig (2004)			Patronage	Optimization	Hosted service; Consulting, Transaction, SaaS		Embedded			License based: Dual license			
Chang, Mills & Newhouse (2007)	Community	Macro R&D infrastructure			Central Support					Split licensing	Value-added close source		
Onetti & Verma (2008)					Reciprocal					Dual licensing	Academic		

Figure 2.3: Overview Business models taxonomy based on research of E. Raymond (1999) and Rosenfall (2012) part 2/2.

## 2.2.1 Use-Value Models

### Community Source

In the category Use-Value Funding Models we see one OSS business model that is mentioned by Spiller & Wichmann (2002), Perr et al. (2010), Chesbrough & Appleyard (2007), Fitzgerald (2006) and Chang et al. (2007) which is based on the Community Source. The business model "OSS development & Community enablers" as mentioned by Spiller & Wichmann (2002) is based on two types of actors: marketplaces and conference and trade fair organizers. Marketplaces like SourceXchange and Cosource.com function as a matching place for potential buyers and sellers. The buyers are organizations or individuals that are looking for software improvement and the seller is the OSS developer community. The potential of the exchange is based on the assumption that many developers prefer to decide themselves on what product they want to work on, which is impossible working as an employee. The marketplace also improves the development process by offering the provision of a project manager and productivity tools. Additionally, demand aggregation is an acknowledged advantage, multiple buyers with a similar problem can aggregate their funds on the marketplace to obtain a OSS solution. Spiller & Wichmann (2002) state that till 2002 as far as known, no company in this business became profitable. The authors see that conference organizers are the stakeholders that enhance interest in OSS projects and arrange meeting opportunities for the OSS community and business partners. The authors have seen that pure marketplaces and exchange models have failed because the value added is possibly not sufficient to bear a business model purely on a matching function. In these business models, revenue can solely be generated from the demand side because OSS developers are presumably not willing to pay for the service.

Perr et al. (2010) agree on the business model where a community source or consortia of enterprises deliver shared OSS. They see the advantage captured from the organizational side where software license costs are reduced by the contribution of joint development. Spiller & Wichmann (2002) discuss that there might be an actor that drives the consortium as a leader. According to the authors, the consortium vendors allocate a fraction of the budget for the resource commitments that are involved in a shared community source projects. This business model is recognized as a "collaborative return on investment" and many public organizations have already been involved (Perr et al., 2010, p.448). Chesbrough & Appleyard (2007, p.65) refer to the business model categories of Perr et al. (2010) and state that the community source business model encompass users with almost identical needs to pool their resources and address the particular need. Fitzgerald (2006) explains companies' duty to leverage the talents of the OSS community, which eventually will allow them to increase development productivity and the benefit of free work. The author does not describe precisely the way of how it is enabling the revenue models of organizations but sees it as an essential step in the creation of an OSS business model. The community business model of Chang et al. (2007) distributes the costs of sustaining the product/service with the creation of a community of users and industry partners. These actors collaborate on the development work and maintenance. An example which the authors mention is the Apache Software Foundation with its decentralized community of developers. According to the authors, a weakness of this business model is that it relies on community donations. Riehle (2009) and Popp (2012) characterize the community source in the "community open source model" Popp (2012) states that if the OSS is controlled by a community of stakeholders where even pre-sales and sales activities are freely offered we speak of community open source (Popp, 2012). The source code of community OSS is only available under one license, so dual-licensing is not possible. This creates the opportunity for anyone to enter the market and generate revenue from the project without being disadvantaged (Riehle, 2009, p.3). According to Riehle (2009), the contributors to community OSS used to be volunteering developers and has seen a movement towards the representation of non-profit foundations. This is also applied by one of the stakeholders in this research, where legally the foundation is the owner of the project but since it is controlled by its members, it still represents the community that keeps working on the project for free.



## 2.2.2 Indirect-Sale Value Models

### Services & Support

Originally, OSS companies focused on offering services to companies that are unable to install, implement and integrate software themselves (Popp, 2012). Perr et al. (2010, p.443) state: "Among the earliest open source business models were models focused on professional services and consulting." Usually, requests for support are sent to OSS forums and mailing lists since this is free of charge, but like in a lot of cases, support can also be part of an OSS business model and purchased from a third-party provider (Fitzgerald, 2006). Koenig (2004) illustrates in his research that support and consulting business models are being applied by many OSS companies and can be very profitable. In the following section the authors' perceptions and findings on service and support business models will be discussed.

Daffara (2007), Rönnbäck et al. (2010) and Stallman (2002) listed the types of services a company could offer: installation support, integration, training, consultancy, legal and technical certifications. Hecker (1999) created two business model categories: "support sellers" and "service enablers". Accordingly, support sellers are the companies that generate revenue from media distribution, branding, training, consulting, custom development and post-sales support (Hecker, 1999, p.49). The author describes service enablers as companies that create and distribute OSS primarily to support access to revenue-generating online services (p.49). Spiller & Wichmann (2002, p.48) see a variety of services that is offered by companies with disparate experiences and backgrounds. The authors distinguish two types of companies:

1. Companies that have a background in Linux or other specific OSS products establishing services based on their technical and product knowledge. Examples are Linux distributors (e.g. Red Hat, SUSE and Ubuntu) and independent OSS service companies.
2. Companies that have special process knowledge in how to provide a service related to IT in general (e.g. KPMG, PwC and Accenture).

The OSS-related services and support Spiller & Wichmann (2002) name are: consulting, systems integration, support, maintenance, remote administration, training and application management. They see that companies from the second type stated above, have a critical factor in their success, which is often their level of Linux expertise. In finding success, the importance of product know-how stands across process know-how in the separate service fields (Spiller & Wichmann, 2002). Figure 2.4 shows the areas of success in the OSS-related service business as mentioned by Spiller & Wichmann (2002, p.50).

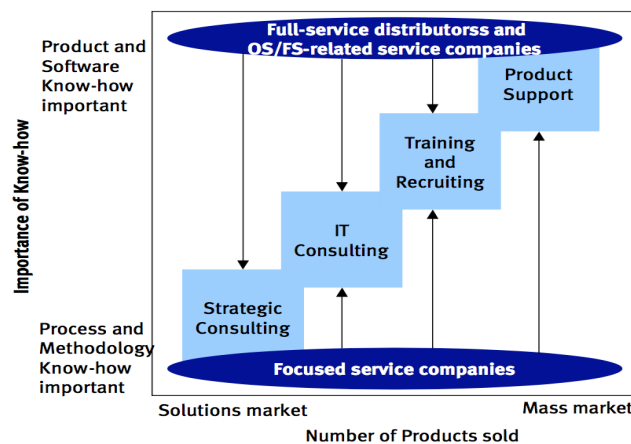


Figure 2.4: Product know-how vs. process know-how (Spiller & Wichmann, 2002)

The figure shows that for strategic consulting, methodology and process know-how are highly important while product know-how is less important or can be easily acquired. In contrast, for product

support offering companies primarily need product know-how while support process know-how can be acquired. Spiller & Wichmann (2002) state that companies with OSS background will primarily be successful with offering support and training, for which product know-how is important. Perr et al. (2010) mention that service providers are the ones that smooth out the rough edges of OSS, where immature code is not customized. The authors see that these service providers facilitate the disruption of established software markets by promoting the wide distribution of OSS (p.443). This in turn increases the capture of customers and enables up-scaling of service offerings. Goldman & Gabriel (2005) makes a distinction between two business models that are focused on offering service. In the first one, the authors mention consulting services that leverage the OSS product. The second one is focused on a service that is based on the OSS like a subscription service that updates the customers' software with assured code (Goldman & Gabriel, 2005).

Krishnamurthy (2005) identifies Third Party Service providers as companies that have service as their only revenue stream. The author explains that these companies are competing with the OSS developer community to provide customer service, and are not focused on selling the product. Perens (2005) also identifies service businesses that participate in the development and maintenance of OSS. The author describes that service businesses create solutions which are based on the integration of multiple OSS programs with a connecting OSS specialized for the distinct customer. Chang et al. (2007, p.2) refer to a central support model where a central body provides robust releases and support for OSS products that are important for the strategy of its community. To conclude this section, the authors listed the advantages and disadvantages of a business model that is based on support contracts. Advantages:

- It ensures long-term sales and profits because large organizations often require support for their used software.
- It is a predictable and dependable stream.
- The model ensures a stable and large number of clients and contracts because of high subscription renewal costs.
- Provides users more options and different levels of support for a variety of organizational needs.

Disadvantages:

- No need to pay for support because it is OSS and free information is available.
- It has to ensure a large number of already available users/ an existing base of customers that need support or added value.
- Support architectures are easy to clone from existing ones.

Chesbrough & Appleyard (2007) indicate that the business models support, subscription and professional services can also be found in the proprietary side of the software industry. The business model 'support' is applied when the company generates revenue by selling customer support contracts. The authors acknowledged another business model called 'professional services/consulting' and is based on revenue by offering professional services, training, consulting or customization of the OSS. The authors mention a variety of services which shows that there are many possibilities for OSS business to generate money with this business model.

### 2.2.3 Direct-Sale Value Models

#### Dual Licensing

Companies might want to use two types of licensing side by side (Hecker, 1999) for one product, this is called dual licensing, twin licensing or split licensing. Companies that adopt this business model base their source of revenue for a large part on license fees and focus on the type of license the software is being distributed under (Chesbrough & Appleyard, 2007). The company can offer the different versions and let it depend on the class of user or the intent to distribute the software and create derivative works (Perr et al., 2010, p.445). Koenig (2004) describes dual licensing as a business policy that lets the customer choose between two options: a commercial license or open source license. If the customer chooses to license the software under an open source license like the

GPL, it allows him to sub-license the product's source code to numerous levels of sub-licensees. This is only possible when it is re-licensed in the source code format on identical terms (Onetti & Verma, 2008). Organizations also offer free use of the software sometimes, but adds limitations in use or a fee. The fee is usually applied when the customer wants to use features that are not enabled in the software under an open source license. The commercial license is available under more conventional licensing terms because the customer does not have to publish the modifications of the source code (Onetti & Verma, 2008). Koenig (2004, p.3) and Chang et al. (2007, p.6) discusses several advantages of dual licensing for the organization that provides the software:

- Improved customer awareness and faster adoption.
- Stronger competitive positioning.
- The creation of a large base of users to find bugs & recommend improvements.
- High level of flexibility for users and organization.
- Allows customers to use and customize the software for further sales.
- Increases number of users and potential sales.

Some users might be happy with using the free license and providing support themselves and some will want to create extra assets (Perr et al., 2010). According to the reviewed literature, OSS businesses should focus on the customers that want extra assets. The Indirect- and Direct-Sale Value Models focus on the realization of the commercial needs of customers, which is based on the vision of OSS companies that share this commercial background. Riehle (2009) and Popp (2012) depicted this type of companies by categorizing them under "commercial open source" businesses, meaning that the "open source software projects are owned by a single firm that derives a direct and significant revenue stream from the software" (Riehle, 2009, p.1). The main difference with the community source model are the control and ownership structures built into the business models (Riehle, 2009). The next move was towards the distribution for OSS like SUSE is doing for Linux. Currently we can find hybrid business models where companies offer services next to a packaged OSS distribution or enterprise versions.

## 2.3 Value Creation

Wieringa & Gordijn (2005) acknowledge the core of a business model to be composed of value exchanges which represent the offered valuable services of businesses and other stakeholders. According to Popp (2012) value is generated by software vendors providing OSS and the customers using it. Riehle (2009) states that the cause of overall exponential growth of and value creation for the open source ecosystem is the growth in number of viable OSS products. The different stakeholders recognized in the sales process of an OSS product and the value exchanges can be expressed using the e3-value model (Gordijn & Akkermans, 2003). This model enables us to express the specific stakeholders and value exchanges of a business model, in this case an open source business model. Modelling the value chain with a conceptual modelling approach makes it easier to recognize the different OSS business models and the advantages for each stakeholder that applies, and is part of the business model. Huemer et al. (2008) describe the e3-value model as a tool which maps complex multi-steps scenarios using the existing technique called "use case mapping". Within these use case maps, a scenario path indicates the direction, the start and the end of a path. The start stimulus represents a specific need of an actor while it ends after the final segment of the path (Huemer et al., 2008). Furthermore, AND- and OR forks are used to model sub-paths. Figure 2.5 illustrates the value exchanges between the stakeholders that are recognized in existing OSS business models. The figure shows all possible interactions and is a combination of the previous discussed models, which can be combined into a hybrid model. The e3 value model is built on the pre-defined elements of Gordijn & Akkermans (2003) which will be explained in the following sections.

First, the *actors* or *market segments* exchanging value in a business model are defined. Following the guidelines of the e3 value model of Gordijn & Akkermans (2003, p.48), "an actor is perceived by his/her environment as an economically independent (and often also legal) entity". The authors define the market segment as a: "concept that breaks a market (consisting of actors) into segments

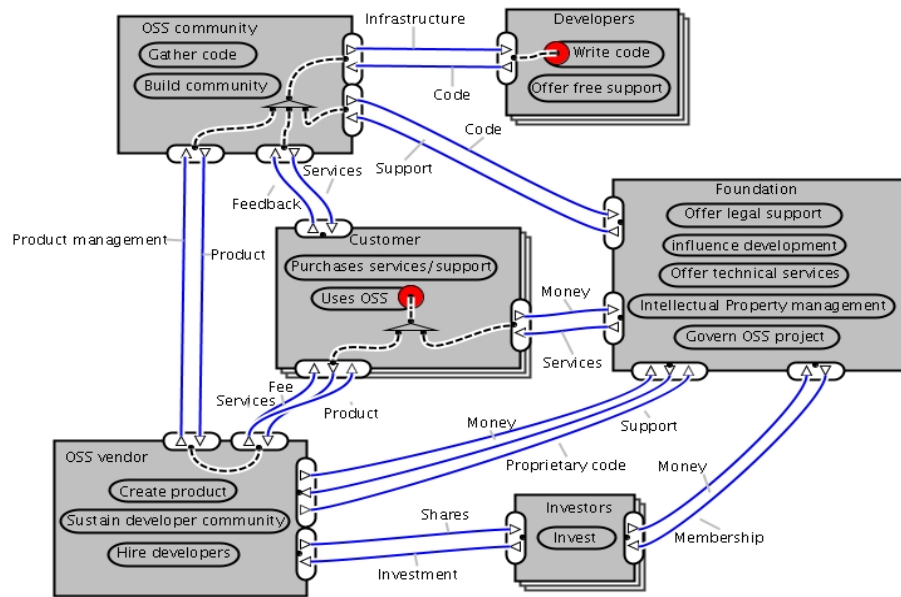


Figure 2.5: E3-value model of OSS business model stakeholders and their value exchanges.

that share common properties". The following actors and market segments are recognized who each execute activities:

- **Developers:** The developers write the code as the base of an OSS product. Additionally, they offer free support through the OSS community.
- **Investors:** Investors play an important role money-wise for OSS start-ups and non-profit foundations offering funding for the development of OSS.
- **Customers:** The customers are the end-users of the OSS and purchase the product and close services and/or support contracts with OSS vendors.
- **OSS vendor:** The OSS vendor providing the product, services and/or support.
- **Foundation/ Association:** The non-profit software foundation works as a collaboration enabler between the OSS community and the commercial OSS vendor (O'Mahony, 2005).
- **OSS community:** The community operates like a hub since it directs the value directed towards the OSS vendor, the customers and possibly an OSS foundation.

According to Gordijn and Akkermans, the activities are performed for profit or to increase its utility. Subsequently, *value objects* are exchanged between the actors, which can be services, products, money or even consumer experiences (Wieringa & Gordijn, 2005). The *value ports* are used to portray whether an actor provides or requests a certain value object (Gordijn & Akkermans, 2003). These individual value ports are grouped in *value interfaces* which show the specific value object the actor is supposed to exchange. Figure 2.5 shows a value exchange between the community and the OSS vendor where the community exchanges the value object "product" since they develop a product based on the value object "code" offered by developers. According to Riehle (2009), the community creates the value by developing a product for the OSS vendor leveraging a faster time-to-market and low development costs. In return, the OSS vendor helps the community to market it and offers "product management" to make it marketable as a professionally developed product (Riehle, 2009). Additionally, the OSS vendor often provides a platform for the OSS community to communicate and exchange code in order to sustain the community activities. OSS vendors might receive monetary investments from investors in exchange for company shares. These vendors take advantage from the tight connection with the OSS community by hiring the best developers from this community to work full-time on their product.

The "foundation" actor offers developers several services like legal support and intellectual property management. The foundation is a separate entity which is able to host and govern a software

project (Rosenfall, 2012) when the vendor is offering the services to the end customer. The company exchanges value in return in the form of sponsorships and support to both the community and/or a foundation (O'Mahony, 2005). O'Mahony (2005, p.408) explains that assets like proprietary code, financial resources and hardware can be donated by the vendor to a foundation, and in return some foundation offer a membership as a sponsor with an advisory role. The same role is apparent for investors who want to support OSS foundations by offering money or intellectual property. The customer has two possibilities when it comes to choosing the source of support and/or services. The OSS vendor offers paid support and/or services while the OSS community offer this for free. It is up to the customer whether they want to sign a SLA with a vendor or to find help from the community. Here, the foundation often works as a middle-man between the customer and the community. The OSS community receives value from the feedback given by the customers which in turn makes it possible to offer the users and vendor a better product.

## 2.4 Open Source Licensing Policies

In order to develop software under the Open Source name there are some requirements as mentioned in the Open Source Definition shown in table 1.1 (*The Open Source Definition*, 1999). OSS licenses are ones that conform to the the latest published version of this definition (Hope, 2007) which emphasizes the key elements of "free distribution, readily available source code and the right to make derivative works" (Kennedy, 2001, p.348). Hope (2007, p.8) determines that the owner of the OSS should: "1. Refrain from keeping the source code secret and 2. grant an IP license to others so that they have the legal right to access and manipulate copyright-protected aspects of the code". This means that all OSS licences disclaim warranties and often try hard to limit liabilities (Kennedy, 2001). Kennedy indicates that OSS licenses are different from traditional licensing concepts as it uses a license to give the user more freedom instead of more restrictions. The choice of a OSS license is very important from both the legal and commercial perspective (Pal & Madanmohan, 2002). The OSS license affects the business' activities and therefore OSS business often define their business model around the selected license (Onetti & Verma, 2008). Onetti and Verma have studied the effect of licensing on the opportunities of funding and acquisition where potential investors are likely to inspect the business' actual use of OSS and its compliance with the selected license.

OSS business-owners can perceive more difficulty in choosing a license compared to proprietary software vendors who just create custom made licenses, because they have less flexibility (Onetti & Verma, 2008). The licensor of OSS could be a single developer, a group of developers or a corporation (Lerner & Tirole, 2005). The choice of the most fitting license is based on a complicated set of motivations of the software owner and the preferences of the developer community (Lerner & Tirole, 2005; Krishnamurthy, 2005). According to Onetti & Verma (2008) the choice should also be based on the business' motivations, project characteristics, its nature and the predetermined audience-impacts. Businesses implicitly choose a business model when they select a certain type of license for their OSS (Onetti & Capobianco, 2005), but the question is which license directs towards a chosen business model. This can be based on the benefits businesses choose to have. Developers usually are for the non-commercial benefits like peer recognition and career advancement, while a lot of businesses have commercial incentives driven by the money that could be made of OSS products and complementary services (Lerner & Tirole, 2001).

Lerner & Tirole (2005) created a model of the available types of OSS licenses to capture the rationale behind the choice for a specific license (see figure D.1 in appendix). The authors described all Open Source initiative-approved licenses plus some that are not approved and determined their restrictiveness. Kennedy (2001, p.15) divided the licenses into four other categories naming: the GNU General Public License (GPL) or "free" licenses, the BSD set of licenses, the Mozilla Public License and at last other non-GPL and commercial licenses. The author states that the most important distinction between the before mentioned licenses is "in the way the license addresses the issue of permitting derivative works to be later made proprietary". These categories resemble the most used licenses within the open source community (Fitzgerald, 2006). The choice between all the different licenses might be hard, but the focus lies on what the licensor wants to do with the software and to decide

whether the non-commercial or commercial incentives are more important. Additionally, the exact differences and their business implications are difficult to grasp and often need help from legal entities. Consequently Open source licenses are often subject to misinterpretation (Rosen, 2004), so for detailed explanation we will redirect to the official website of the specific license or the book of L. Rosen.

Onetti & Verma (2008) found cases where OSS businesses change licenses over time or even remove license constraints for business reasons. In these cases, the change was needed because of the poor fit of the license with the business goals of the company. The fact that often the initial licensor is the person who created the project, lacking a legal or business background, adds to this reason. The patent licenses described in research by Rosen (2004) and described in this section are hardly consistent with business models that are focused on selling software. The secret according to Rosen (2004) is that "licensors can make money on what the open source doesn't grant" (p.231). According to Rosen it is therefore more rewarding for businesses to consider the exclusions from licenses rather than the grants looking for new revenue streams. Kennedy (2001) states that the most important exclusion is the trademark or brand identity, which are excluded, explicitly or implicitly, from all open source licences. Another way of generating money based on licensing is dual-licensing or split-licensing, as discussed in section 2.2.3. Onetti & Verma (2008) classified licenses based on the business models of OSS companies to review the correlation between a company's license and business approach. This is done by looking at the company's revenue drivers, ranging from license fees to services, and the license the company's business model is based on differentiating between copyleft, GPL like licenses and non-copyleft, BSD like licenses. Onetti & Verma (2008) agree on the fact that it is hard to assign OSS businesses to specific types of business model categories, because a lot of the businesses adopt hybrid business models. Nevertheless, the authors believe that focusing on the revenue drivers helps to understand a business' strategy and what its business rationale is. Comino et al. (2007, p.1) state that "while projects distributed under highly restrictive licensing terms (like GPL) have a significantly smaller probability of reaching a stable release, applications towards sophisticated users have a larger probability of evolving in the development status". To conclude, the Open Source license has to be aligned with the terms of a business model of a business and its developer community which creates the product.

## 2.5 Measurements of Success

The current generation of OSS businesses is gradually focusing more on gaining competitive advantage and being successful in the market. The success of OSS businesses in its ecosystem is important because for an ecosystem to function effectively, Iansiti & Levien (2004, p.2) state that every domain it consists of should be healthy. Despite of this recognition, success is difficult to measure, it is a subjective definition (Thomas & Fernández, 2008). Nevertheless, a lot of research is performed on the topic and many ways to measure success are constituted. Business success differs from OSS project success since, generally speaking, the creators of the projects are not profit oriented. The success of OSS project development is discussed by several researchers (Crowston et al., 2003; Midha & Palvia, 2012; Comino et al., 2007; Singh et al., 2008; English & Schweik, 2007; Stewart et al., 2005). Nevertheless, measuring success during the process of development does not encompass the impact of OSS in the market. Therefore it is also important to review when software is successful market-wise, and how OSS businesses became successful in the past. We need to address a combination of business success literature and open source project success literature to create an encompassing overview of which business models work best for OSS entrepreneurs. Successfulness of OSS businesses is a less discussed topic in literature, therefore we will try to come to a conclusion what OSS business success might constitute.

OSS businesses are often established within an existing open source community working on a specific Open Source project. Therefore the measurement of OSS project differs from business success. Midha & Palvia (2012) look at the success of OSS projects and determine which factors can lead to this. The authors explain that OSS projects evolve and transform over time, needing a longitudinally examination of the possible success factors over time. The two measurements Midha & Palvia (2012)

use are *project popularity* and *developer activity* in the software development context. Other researchers measure the success of a OSS project according to community activity, like the amount of free contributions (Feller et al., 2002; Crowston et al., 2006; Stewart et al., 2005) and volunteers fixing bugs or adding features (E. Raymond, 2001). Lee et al. (2009) came to five determinants for OSS success and emphasize the significant relationships among them. The authors conclude that *software quality* and *community service quality* are the most important components to decide whether OSS is successful in terms of user satisfaction.

For this research, mostly business success is relevant because of the examination of business models it the prosperity of growth of OSS businesses. Thomas & Magilvy (2011) gathered three success criteria within this group: business continuity, met business objectives and delivery of benefits. Thomas & Magilvy (2011) and Simpson et al. (2004) address that business success is hard to define because it is determined to be multidimensional construct. Nevertheless, Thomas & Magilvy (2011) showed that when success criteria within a business are formally defined and measured, the outcomes of IT projects are improved over time and the resources are better applied. This research uses success criteria as mentioned by the stakeholders and grouped them into the following categories: project management success, technical success and business success. Vilkinas et al. (2011) created a literature overview of criteria for business success and mention subjects like customer satisfaction, making profit and growth of benefit. Overall, financial criteria are often seen as the most convenient measure of business success (Walker & Brown, 2004). Headd (2003) shares this idea and determines the success of businesses by establishing the fact that the business is making profit. Business success can also be brought back to the simple definition where it is equivalent to continued operations, or as Watson et al. (1998) call it: survival. The author's research showed that businesses with more resources - meaning that the business is large, having employees and good financing - are more likely to survive. But survival is not useful alone since businesses can also close while they are successful (Headd, 2003), and using only financial criteria omits the intangible goals of business owners. Simpson et al. (2004, p.484) argue that success determined by an outside influence lacks the relevance if the entrepreneur does not see him/herself and the owned business as successful. Based on this idea it might be correct to base success on whether the goals of the business are achieved or are working towards it (Beaver, 2002). This is consistent with Jennings & Beaver (1997) who state that success of small businesses is "the sustained satisfaction of principal stakeholder aspirations". This definition can also be applied to medium and large businesses since success for small businesses give them the possibility of growth. For this research we can state that business from the sample are successful based on their propensity for growth and their met business goals.

### 2.5.1 Business Model Success

The determinants of success of OSS businesses is based on the success of their business models because business models are often linked to competitive advantages (Wirtz, 2011). In turn, these competitive advantages lead to future business success. According to Zott et al. (2011) the performance of businesses is positively related to business models that are novelty-centered and that are connected with an applicable market strategy. Sanchez & Ricart (2010) see that interactive business models provide a sustainable competitive advantage because they offer value to the community. The previous section explained how business success often is measured and this section defines what determinants are used to create business model success. This section defines the focus point of business models that have been successful according to academics.

Bonaccorsi et al. (2006) see that the size of the company does not explicitly favour the adoption of Open Source and that companies in their entering phase overall combine Open Source and proprietary software in a hybrid business model. Krishnamurthy (2005) states that OSS products having the highest profit potential also have high relative product importance and customer applicability. The author sees that these companies have large developer communities supporting the OSS company and the product while having outstanding direct and indirect marketing support. Linux is an example that is used with high relative importance, because the product is an operating system with high customer applicability because the product can be installed on every computer (Krishnamurthy, 2005). Additionally, Krishnamurthy mentions that certification of developers on a OSS product of-

ten leads to strong revenues. These developers creating OSS play another important role since their personal success is often related to business success (Walker & Brown, 2004). Schief (2013) states that leveraging external workforce for the product development of OSS is a way for OSS companies to foster their agility and firm growth. According to Lee et al. (2009) the individual net benefits are also influenced by the use of OSS and its user satisfaction. The authors state that in turn, this user satisfaction is significantly influenced by the quality of the software and the community service.

Rubenstein (2017) listed four business models that according to his study have shown to be successful for OSS companies. The author mentions the services and support business model (explained in previous section) and uses Red Hat as an example of a company that became successful by offering free software while charging for the services a company often really needs. The author uses Mozilla as an example because the Mozilla Corporation generates revenues from partnerships with big companies like Yahoo, Google and Amazon which pay big money to be included as a built-in search option in the open source Firefox browser (Rubenstein, 2017). Advertisement partnerships are recognized as a business model for OSS companies that are moving towards a more economical model. MySQL, together with many other OSS companies, is an example of a large successful OSS business that focuses on paid additional features. Their two versions of the software each provide different features focusing on whether the customer is a large enterprise; which needs extra specific features or whether the user is non-commercial who typically does not need any extra features and therefore can use the open source version. This "versioning" of a product is a recognized way of generating revenue (see section 2.1 for a detailed explanation).

## 2.6 OSS Start-up Strategies

Business models change over time and become better as business is going well. The perfect business model is not built in one day an start-ups might have to try different models in the first phase of establishing a business. Software start-ups often face difficulties in the initial phase because of tenacious competition and time-pressure from the sometimes chaotic and rapidly evolving market (Paternoster et al., 2014; Bosch et al., 2013). Lichstam & Johansson (2016) recognize an intersection between start-ups and OSS since they are both sources of innovation. Start-ups are often resource constrained (Bosch et al., 2013) and OSS offers a combination of low cost labour and innovation sources (Lichstam & Johansson, 2016). There are several strategies for companies that want to change from proprietary to open source software, but there are no clear strategies for start-ups that want to generate revenue with the establishment of an OSS business. The software industry is currently experimenting with new business models based on collective creativity as in the OSS ecosystem (Chesbrough & Appleyard, 2007). The ecological approach shows that new ventures are dependent on processes in the bigger ecosystem, or also called "macro-population" (Li & Liu, 2013). The forces of the environment have shown to be influential in new venture founding and the survival of start-ups (Fernhaber et al., 2007). Entrepreneurs have to be attentive to new opportunities in the ecosystem to be able to withstand in the environment (Li & Liu, 2013).

Fogel (2005) has studied how OSS is produced and which steps will lead to a successful OSS product. The author divides the OSS development process into two tasks that need to be accomplished: the acquisition of both 1. users and 2. developers. A known issue in open-oriented organizations is how to attract and keep the participation of these users and developers (Chesbrough & Appleyard, 2007). According to Fogel (2005), the interaction between the two attends complexity to an OS project's initial presentation. The strategy to do so, is to reward according to the amount of time both groups put in. The effort and reward should correlate reliably otherwise the business risks losing the interest of most stakeholders and might stop to devote effort to it (Fogel, 2005). The presence of interesting subjects for all parties grants the initial phase of supply of the collective good, the phase where cooperation costs overpower benefits (Bonaccorsi & Rossi, 2003). After this first phase, more stakeholders will see that the contribution is profitable for other stakeholders and therewith create a virtuous circle. The collective event becomes eventually self-sustaining, creating a new equilibrium in which the stakeholders choose to collaborate (Bonaccorsi & Rossi, 2003). The hard thing is that OSS start-ups have to compete for contributors and acquire large groups of developers who can engage



within the community (Chesbrough & Appleyard, 2007).

Chesbrough and Appleyard introduce a new strategy for OSS entrepreneurs called the "Open Strategy". The authors imply the importance of finding a way to profit from innovation activities in open initiatives and mention this can be reached through deployment, hybridization, complements or self-service to sustain their business over time. The hybridization is recognized in research performed by Bonaccorsi et al. (2006). The authors explored the entry strategies of software firms in the Open Source field based on empirical evidence (p.2). Their study concludes that the Hybrid Business model where both proprietary and open source license models are applied is mostly used by business entering the OSS ecosystem. Their research showed that the orientation towards Open Source has a positive impact on the entry phase and can be associated to the intensity of motivations that are typical of OSS developers. To the authors' surprise, the size of the companies does not favour the adoption, stating that small companies enjoy more benefits of being Open Source. Chesbrough & Appleyard (2007) add that if the open strategy is performed effectively, it will balance "the powerful value creation forces that can be found in creative individuals, innovation communities, and collaborative initiatives with the need to capture value in order to sustain continued participation and support of those initiatives" (p.73).

Start-ups are exploratory in their first phase but have to run on tight schedules to fulfill the wishes of funding companies and launching dates (Bosch et al., 2013). Entrepreneurs are not able to spend years on their idea anymore while burning venture capital and figuring out the perfect strategy (Blank, 2013). It is important for entrepreneurs to be efficient and systematic (Bosch et al., 2013) and a strategy that recognizes this is the Lean start-up, which is seen as a less risky strategy for start-ups to succeed (Blank, 2013). Lean is often perceived as a part of agile software development, practices that became popular around the year 2000. Both lean principles and agile software development focus on the strong presence of customers' contribution and short feedback cycles used for the creation of customer value (Bosch et al., 2013). The lean principles of failing fast and continuous learning create quickly a minimum viable product open for customer feedback (Nobel, 2011). The strategy helps start-ups to be efficient by minimizing development efforts while maximizing its value and systematically validates if the product is generating the supposed customer value (Bosch et al., 2013). It also favors iterative design and fast creation which makes it easier for the business to move along quickly to the market (Nobel, 2011). Entrepreneurs usually prefer technologies that can accommodate the change in their new product and its management (Giardino et al., 2014). The lack of resources is therefore often taken care of by the use of OSS, which is part of the original idea of Lean start-ups as established by Eric Ries. Ries mentions that Lean is meant as low-burn by taking advantage of, for example, open source agile software (Ries, 2008). So the use of OSS in Lean start-ups is apparent but, the application of the Lean principles for OSS start-up businesses is not. Reviewing the principles of Lean, we can say that it is possibly a working strategy for OSS start-ups but the lack of literature on the application of the Lean start-up strategy specifically by this kind of businesses, leaves us without evidence. Nevertheless, the principles are applicable in the creation of a guideline for OSS start-ups.

# Chapter 3

## Case Study Analysis

Software and business literature has shown many variations of OSS business models. A comparative case study of 10 OSS business businesses and their business models gives us empirical evidence on what business models of successful businesses look like. This chapter consists of a description of the interviewees, their incentives to participate in the open source ecosystem, their start-up phase and the comparison of their business models with the use of the Software Business Model Framework of Schief (2013). Altogether, the analysis gives us a view on the current OSS landscape and the possibility to generate a guideline for start-ups.

### 3.1 The Interviewees

The sample size is constructed by certain company criteria (see 1.3.2.1). Due to time constraints it was not possible to interview employees with the same role, and therefore differ. Due to specific business model information the outcomes of this research are anonymised. Table 3.1 shows the profiles of the interviewees of the comparative case study in random order.

#### 3.1.1 Incentives

The incentives of developers to contribute to open source projects is a widely researched topic (Oreg & Nov, 2008; Von Krogh et al., 2003; Lakhani & Wolf, 2003). Reasons range from intrinsic motivations, self development and intellectual motivation to extrinsic motivations like better job opportunities. The motivations of developers differ from motivations of firms, but are according to Bonaccorsi & Rossi (2006) heterogeneous. With the right intrinsic and extrinsic motivations it is possible for entrepreneurs to become a successful member of the OSS ecosystem. Question 3 of the interview points towards the reasons of companies offering OSS rather than proprietary software. This was part of the interview because little attention is given to the motivations of firms engaging in the open source ecosystem (Whichmann, 2002). Nevertheless, Bonaccorsi et al. (2006) performed a case study under 146 Italian open source based companies stating that firms' incentives are mostly extrinsic and built on the monetary reward. The authors recognize 3 types of incentives in literature; economic, social and technological. These types of incentives were also recognized in the answers of the stakeholders. An example is company I. that started as a closed source software company but saw that this would not help the company to grow. They decided to go Open Source because it would be hard to compete with the other bigger, closed source DNS companies, and because of the helpful input of the users. Company D. also adds that the nice thing of Open Source is that they do not have to buy licenses of other companies, you are independent of the licence en price policies of the larger software companies which is an important economic incentive (Lerner & Tirole, 2005; Bonaccorsi & Rossi, 2006). They also see that OSS has the same quality as proprietary software and that it is easy to customize based on the clients' wishes. Company H. took "the chance to act as a disruptor in a very staid, unresponsive, complacent vendor marketplace" by offering an open source solution. There already were

Company	Type of software	Year of establishment	Country of Headquarters	Role of interviewee
A.	Project Management Tools	2015	Spain	CEO
B.	Integration platform & ESB	2006	U.S.A.	Account/ Development manager
C.	Linux distribution	1992	Germany	Regional Director EMEA North
D.	Application Service Provider	2003	The Netherlands	Founder
E.	Git-repository management	2011	x	Product Manager
F.	Government geographic data publishing	2007	The Netherlands	Software Engineer
G.	Content Management System	2016, project 1998	Germany	CEO
H.	ERP+CRM	2001	U.S.A.	CEO
I.	Domain Name system server	1999	The Netherlands	Product Manager
J.	Linux distribution	1993	U.S.	Account manager/ Sales

Figure 3.1: Case study company details

many similar proprietary software companies of their kind and the interviewee noticed that certain challenges were better to be addressed in a Open Source way.

An incentive that a few of the stakeholders share is a social one, sharing the thought that software simply should not be a proprietary good (Feller et al., 2002). The founders of company F. and A. have sympathy for the Open Source idea of sharing knowledge. They both also already had experience with the Open Source concept and therefore were able to create a product around it. The founder of company G. feels like it is necessary to give back when you have the ability to write good code. Company E. was built with the same reason, without thinking about profit the founder just wrote really good code and was already present in the strong culture of doing everything Open Source. Usually when this is the case, the developer has a personal problem that needed to be solved and eventually took the chance to create a company to serve others the working software.

Company C. took the chance to make an open source product available for the corporate market. The interviewee saw that the demand was growing, and that not only developers were interested in the software. Company J. sees this as a positive change, because offering OSS to the corporate market raises the adoption rate of the software. This adoption rate is high because the software is free of charge, which is beneficial for the companies using the software and the software company itself because they get more feedback, which in turn enables growth. This same company sees the economic value when the software is free and complementary services are paid for (Hawkins, 2004). The interviewee adds that people who like what they are doing create better things than people that have to, meaning that open source developers create software out of love for the common good. A final message which all interviewees agree on: " *Open Source is here to stay and to grow even bigger*".

### 3.1.2 The Start-up Phase

According to the interviewees, the start-up phase of an OSS company can be difficult and differs from a proprietary software company. Every OSS company has had a different course where some were growing without any help and some were acquired to receive financial backing. The history of the companies differ in different ways. The first difference has to do with funding, some of the stakeholders have received angel investments, some venture capital and some just put personal money into the open source project. Companies A., D., F. and H. have not received any money from people or firms outside of the company. The CEO of company A. admits that the start-up phase was quiet hard, putting in personal money to make it work. The interviewee states: *"So there was some theoretical knowledge about [business] models and then we were just trying. And then the customers and everything just evolved"*. Company A. started purely with offering services, and five years after the establishment of the company, open source enterprise software was offered. The interviewee tells that after a year he was able to finance himself but also that the company wanted to stay small. For company A. it helped that they are in a niche sector which led to being the market leader after three years. Company H. also only invested personal capital and took donations from friends and family to be able to establish the company.

Company F. was never really focused on money and therefore has been growing slowly. The Interviewee sees the benefits of creating an Open Source based product: *"[...] I think that particularly the open source movement really facilitates the start-up market."*. Accordingly, the start-ups are facilitated by the open source community because they offer freely usable software. He mentions that if a start-up has to start off with paying some vendor a lot of money to create the software, the start-up would have a lot less business. Using OSS makes the business model of an entrepreneur much more relevant. Interviewee E. mentions that they did not earn any money in the beginning due to the fact that the company did not have an enterprise product. Therefore company E., and also company C. have received venture capital which has shown effect; particularly in the early stages of a company's development, facilitating the development of a real product (Hellmann & Puri, 2002, p.194). It is also recognized that angel investments attract venture capital (Madill et al., 2005) creating more interest for the company in the market. Company J. was at a certain point incorporated as a limited liability corporation, this means that the members of the company are not personally liable for possible debts or liabilities. After that, they joined an American seed accelerator which offers funding and support. Company J. admits that it was not an obvious decision to take funding since they already had hundreds of paying customers, but they saw the opportunities the accelerator could initiate. Company J. was established in 1993 and after the company took venture capital in 1998, the company went IPO in 1999. The investment was used to develop an enterprise product which was a great move to create more revenue.

There is another recognized option to grow as an OSS company, which is being acquired by a bigger company with more experience. This was the case for companies I. and C. Company I. started in the late 90's and is incorporated into a larger organization in 2015. This merge was set in motion to get legal, sales and marketing support while the original team could focus on the technology. The same happened to company C., which was acquired several times by different companies. Company C. was founded in 1992 and acquired in 2003 when both the partner and sales organizations of company C. were integrated. The company was acquired by another organization in 2011, but stayed a separate company. In 2014 the merger was merged with another organization which meant a new parent for company C. The original company became a semi-autonomous business unit, retaining its leadership unit while having shared resources available. Having a larger parent organization above them, offered them financial backing and security. After this final acquisition, a grow of more than 18% in revenue was a fact. According to the CEO, this growth was due to both the financial backing and the expansion of their product portfolio.

The first couple of years of company G. were different. Since the initial release of the project in 1998, company G. was not a company but just a project. The project was getting bigger every day and with the growing amount of contributors, a clear structure was needed. In 2005 an association was established to offer a legal body to help sustain the project. The interviewee tells about the association and the point when change was needed: *"And this worked out pretty well but at a certain point everybody*

realized that you are still relying on volunteers to do the work within the association. So the next logical step was to either employ people within the association or to found a dedicated company to do that." So action was taken, and around two years ago a company was founded by the community, being 100% subsidiary of the association. Being the CEO of the newly founded company, the interviewee states that they have very strict rules imposed by the association. The company purely focuses on service products and "everything surrounding that no one has been supplying yet".

### 3.1.3 Suggestions for Entrepreneurs

To be able to create a model for entrepreneurs that have the aspiration to start an OSS company, question 10 of the interview aimed at tips given by the experts from the OSS market. According to interviewee F., an important aspect of how company F. creates revenue is based on the fact that they keep working on their core products. Nevertheless, the interviewee recommends start-ups to use libraries that are already available to create your own product and to build your own marketing idea around it. Or shortly, take available code from the web and create a special offer. The fact that the software is being used helps in the improvement of the product. On a certain moment in the development phase, it will be time to invest in the core product. This is needed to improve the core layers in your own software product and to keep the whole community rolling. The interviewee has a critical view on start-ups because he sees that some do not "grow up" at a certain point in time, staying for a long time in the initial development phase.

Interviewee of company E. thinks that it is most important to communicate with the clients, and if you do not have any clients as a start-up: just find one client. This aligns with what interviewee H. says what is important; identifying the right kind of customers and making them happy. The interviewee adds that it is important that a business model covers the costs and provides decent profit margins. Also, engaging the open source community and being honest and transparent with all stakeholders is important for company H.

Interviewee E. has a simple suggestion, stating: "Just create a product that 1 person thinks is good, a product that 1 person likes and iterate on this product until they get to the point they are very happy with it and want to give you money for it.". The other option suggested by interviewee E. is to attract venture capital, but they are often not interested in an OSS product. Nevertheless, according to interviewee J. venture capital can be attracted and obtained by giving lots of presentations for possible investors. What the interviewee sees, is that successful companies which originate from the open source world, first created an open source project which became very popular and then started to build a business around it. The interviewee concludes that it might be better to establish a start-up around a proprietary product to earn money and open the software in a later stadium. He states: "I think that that is a important realization, you can make things open source only one time." There is the possibility to change the license, but every time you do that the community will not like it. After the company created an enterprise product, the challenge was to price it right. It is easy to make your product very cheap, but company J. never lost any clients by raising the price of their product. Nevertheless, the pricing should be appropriate and according to the quality of the product. Interviewee B. states that, if they look back at the open source product in the initial phase, it was too good which caused the problem that enterprises did not want to make the step to using their paid enterprise edition. Because of their ambition to grow, the company decided to put less focus on the open source version and more on the enterprise software. The interviewee says that in the end it is important to focus on what the client currently and in the future wants and to work around those ideas.

Interviewee J. agrees with the before mentioned idea of creating a product which is interesting for the market. You should differentiate your company from others and the interviewee sees three ways to do so:

- **Comparative differentiators;** The other company has features A, B and C, so we have features A, B and C.
- **Unique differentiators;** We can do what the other company cannot.
- **Holistic differentiators;** These are the aspects like the company culture and how you are perceived in the market. This comes up later in the sales processes after the acceptance of a client

to start using the software. These differentiators are less seen in the begin phase of the acceptance because there is a need for awareness on the vision of the company and the changing technology trends.

The interviewee sees the importance of operating in an ecosystem where you, as a company, are operating with different kinds of partners. It is important to enter such an ecosystem to broaden the adoption of your technology. According to the interviewee, very important partners are the Independent Software Vendors (ISV's) which are companies that produce software that needs to be certified. The interviewee also recognizes the benefit of having an OSS product in such ecosystems in terms of the integration and licensing of a software product. Because the first time software is created, it has to run on something at the client site, on a cloud or an intern system. The software has to be tested on different layers with which it should cooperate. It is beneficiary to have an open source product because clients first start looking at your free version of the software which is typically easy and fast. After a while, you will be able to approach the client and offer support to install the larger version of the software and ask money for it. Companies that have open source versions of a proprietary software product are dominating with their community and development models. Another strategy, also used by company J., is to buy smaller companies. The strategy of company J. is to buy proprietary software organizations which can be used complementary to their own software, and make them Open Source. They give everything that is licensed-based and thus intellectual property, to the community. The company believes that the speed of innovation lies within the boundaries of the community, and not necessarily within the boundaries of the company. The choice between build or buy is according to the interviewee easily made because it is faster to buy companies then to build the functionality that you need. To predict the future of an open source company, the interviewee of company J. suggests to look at certain elements of a start-up. There are a lot of Key Performance Indicators you can use to see if a company is serious, but with Open Source it seems very important to determine how open they are. This means you should firstly look at the number of code contributions, and secondly at how often the code of the major communities, that is used for the product, is downloaded.

The interviewee of company D. thinks that the importance prevails in the quality of the software, which should be high and created in-house. This will keep the clients with you in the long term. The interviewee of company A. sees that this is where their problem lies, that their software is too complex which makes it hard to find developers for. This led to a community that did not kick-off in a way you would expect of a normal open source community. Again, the quality of software is seen as the most important aspect of an OSS business. Because *"[...] if the software breaks, your business stops"*. According to interviewee C. the power of Open Source lies in the fact that everybody can add something in his or her own way. For a company it is important to have a clear mission and vision for parties that want to connect with the community and the product. The interviewee says it is important to book little success-moments where you make sure you keep innovating and standardizing. Interviewee G. thinks that it is most important to find a balance between the open source mindset and the business mindset. This balance is important because it can be hard to produce OSS and at the same time make a living for everyone that is working on it. The solution is to make sure you have a good developer and a sales person. These two persons should work together on the same product, with the same vision and respect towards each other. To keep track of the open source origins, the equality between developers and business focused employees is key. The interviewee concludes that the business oriented employees need to keep the open source mission in mind, and the developers need to see the necessity of creating revenue.

An issue recognized by companies A. and H. is that prospective customers did not trust the software due to the fact that it is Open Source. Therefore it is important to explain what the company's vision is in a less technical and more business way. On the other hand, interviewees E. and G. see that the software landscape has changed and that customers need less explanation about what Open Source entails and that companies are confident enough to use OSS.

## 3.2 Comparison of OSS Business Models

The data compelled from the interviews is based on the five building blocks of Schief's 2013 Software Business Model Framework: Strategy, Revenue, Upstream, Downstream and Usage. This section shows the outcomes of the comparison for every section of the framework. The outcomes gives an overview of the business models of the OSS companies from the sample. Each section of the framework is explained shown in Figure C.1 in the appendix. For each compartment and sub-compartment the stakeholders have chosen at least one and/or more answers based on the business model of their company. The following sections are based on the answers of the interviewees and notable correlations will be put in perspective. As explained in section 1.3.3 the data is visualized in tree-maps to efficiently create an overview of the interview outcomes.

### 3.2.1 Strategy

The first section *Strategy* is subdivided in the *value proposition*, *investment horizon*, *value chain*, *degree of vertical integration* and the *number of cooperation partners* of a company. The value proposition is aiming at the competitive advantage of a company's offering. Three companies agree on the importance of being an innovation leader, which is achieved by offering new and disruptive software and services (Schief, 2013, p.72). Three companies focus on the functionality of their product with its available features. The value proposition on which most companies agree on is the quality of their product, meaning that they aim for high consistency and dependability. Being an innovation leader in the open source ecosystem means that even under conditions of market failure, innovation is achieved by the collaboration of the community working on a public good (Hippel & Krogh, 2003). Hippel and Krogh see that OSS business models "present a novel and successful alternative to conventional innovation models" (p.212). Only one company has a value proposition focused on their image in the market or tries to differentiate through their price-scheme.

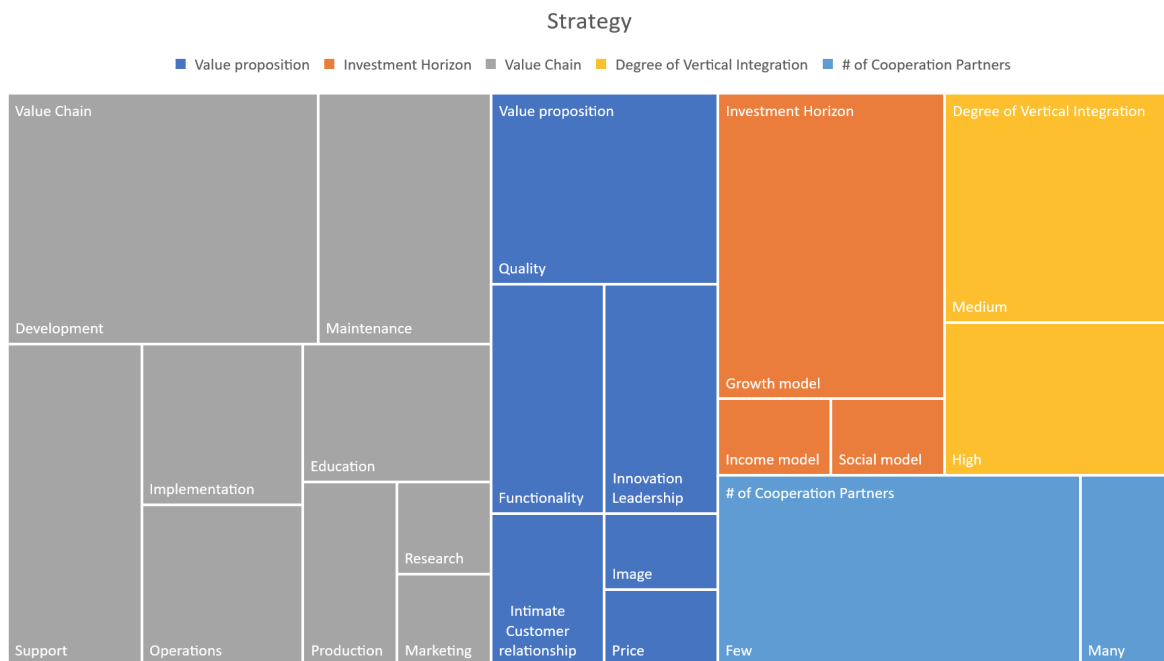


Figure 3.2: Tree-map of the SBMF outcomes - Strategy

In the *investment horizon* there is a clear time strategy that most company's use: the growth model. According to Morris et al. (2005, p.731) the growth model is based on a start-capital including investment with later on a reinvestment in order to grow the value of the firm. This model aims at growing to achieve capital gain for new investors. The growth model is in line with the definition of success

we established in Chapter 2. Yet, the definition of growth is different for some of the companies. Interviewee of company F. addresses that they want to grow in number of employees but do not have a clear strategy to do so. Company E. on the other hand, has a clear strategy to grow in terms of revenue, trying to go public at the end of 2020. Company B. also measures growth in terms of revenue, wanting to have generated a billion dollar revenue for 2020. Company I. also tries to grow but just to meet the vision of the company and not to create such revenues. Company D. envisions growth in the number of partners and users. Usage is also measured by company G. which is trying to acquire more users to be visible in the market and to advertise their services. Company F. uses the Income model, working up to a point where they can sustain the business without making a lot of profit. Only company G. follows a social model, meaning that they are not focused on making profit but on specific clients such as governmental- and other non-commercial institutions.

The *value chain* is based on the main software value chain activities a company may perform (Schief, 2013, p.72) and is seen as the center of gravity of a company's business model (p.45). The tree-map in Figure 3.2 shows that most companies focus on the development of their software. This is part of the upstream activities which are performed until the software is ready to be offered on the market. According to Schief (2013, p.55), sub-activities of the development value chain activities are: Requirements engineering; software design; coding; subsystem testing; subsystem integration; system testing; user documentation and provisioning. Having development as the most important value chain activity, shows that the background of the companies originates from the open source field. OSS development is the main activity of the community due to the fact that OSS is freely accessible to all (Hippel & Krogh, 2003). Since software is their main product, which is in this case free of charge, the companies have to keep the development in pace to meet the growth expectations. When the software is good, more users will use it and more users will know about the company. This again relates to the numbers of users of services and support options. So next to the development of the actual product, support that is offered to the clients is a value chain activity that is important for companies A., C., F., G. and H. This support is separated by Schief into primary and development support. The primary support is the support offered to the users while the development support is reviewing code, which needs deep technical knowledge. Mostly, the support offered by the sample is primary support which is offered in various types like technical support and installation support.

The *degree of vertical integration* depends on the performance of the value chain activity. Rothaermel et al. (2006) assessed whether the activity was conducted within or outside of the boundaries of the company. The calculation is represented by qualitative judgment where the more activities are performed within the boundaries, the higher the degree of vertical integration is and the other way around. This means that most companies perform either all of their activities within the boundaries and some are outsourced to partners. An example is the development of a certain cloud solution to a partner, where the server space could be maintained outside the companies' borders. The *number of cooperation partners* is also based on a qualitative judgment. Eight out of ten companies has a few cooperation partners, this implies that the companies outsource some of the value chain activities to partners. This could be the case when the software company has certain consultancy partners who do the offering and implementation, or companies that offer services around an existing product. Some companies have many cooperation partners, this is possible when the software product is a platform that is used as a base for application development performed by other companies (Schief, 2013).

### 3.2.2 Revenue

The second component is the *Revenue* of the participating companies. The sub-component *sales volume* is based on the number of solutions that are sold and is based on qualitative judgments (Schief, 2013). As the stakeholders are successful businesses with aspirations to grow, Figure 3.3 shows that most of them have sold many solutions and therefore have a high sales volume. For all companies the revenue source is build up from direct paying users instead of revenue through advertising or commission. This is partially due to the fact that service agreements are offered for long-term and based on direct sales (Hoch et al., 2000). The *pricing assessment base* tells how the prices of the solutions are defined. %50 of the companies use a hybrid combination of both usage-based and usage-independent pricing. Company B. bases their prices on the through-put when the platform is used. Company I. bases the



price on the number of subscribers which the client connects to their product. The pricing assessment is often based on the cost-structure of the software: if the vendor is offering a platform the pricing is similar to the one of company B., but when the vendor is offering applications it is more common to price the product usage-independent.

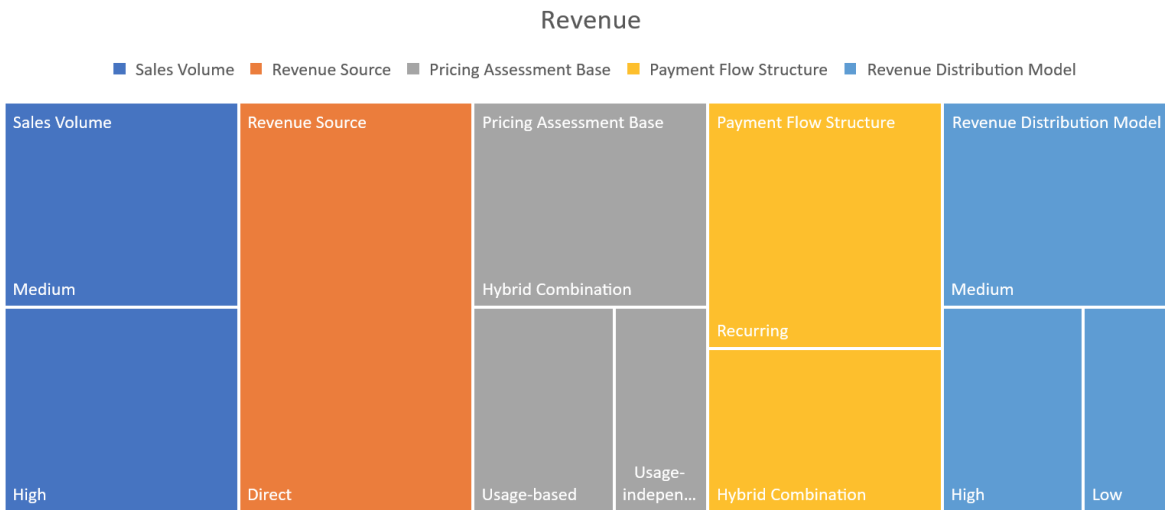


Figure 3.3: Tree-map of the SBMF outcomes - Revenue

The fourth sub-component is the *payment flow structure* and decides the point in time when the customers has to pay for the solution (Schief, 2013). The outcomes show that the majority of the companies have chosen for recurring payments which are similar to subscription fees. Explained by the interviewee of company I., a subscription model is necessary because developers have to be paid on a steady base. Besides that, companies need a constant cash-flow to pay the partners and for other services. According to Dubey & Wagle (2007) a subscription model is the cause of lower revenues during the growth phase because the payments are spread, but over a longer period of time it is beneficial because the streams become steadier and the operating costs of new subscribers become minimal. The interviewee of company B. states that their clients pay for the use of their platform with a yearly-subscription fee because that gives them the time to make sure that the client is satisfied wanting to renew the contract after a year. Only one company has an upfront payment flow structure, and two have a hybrid combination of upfront and recurring structures. Company E. offers their clients yearly contracts because it creates more revenue. The interviewee also states that it is very motivating for the sales department to close large deals, and will even try to offer multi-annual contracts. The final component is the *revenue distribution model* which is defined by the percentage of the companies' revenues that is being shared with other stakeholders (Schief, 2013), the qualitative judgment 'medium' shows that the revenues are not shared with many stakeholders. This is an advantage for the companies since they are able to reinvest in the value chain activities within in the company.

### 3.2.3 Upstream

The *Upstream* component defines the development of a product and/or service. The first sub-component, the *software stack layer* classifies the types of software the company is offering, where in this case study the majority offers application software. The *platform* is the technical platform behind the offered solution. Most companies offer solutions that are installed on servers, this is a traditional and reliable platform still used by many businesses (Schief, 2013). What also got our attention is that a lot of the companies provide cloud computing solutions which gives a clear view on the current trends in the software ecosystem. According to Endo et al. (2010, p.3) companies are attracted to the development of open source cloud solutions because "it allows for resources to be provisioned according on a demand basis [...]".

The next component is the *license model* the software is based on. Since this research is focused on open source business models it is expected the companies apply either Copyleft Licenses (e.g. GPL), Permissive Licenses (e.g. BSD) or a dual model of both (Schief, 2013). Six out of ten companies use the GPL license for their software, this corresponds with the fact that GPL is the most used license for OSS (Stewart et al., 2006). OSS companies often use one of these two types of licenses because of their popularity (Onetti & Verma, 2008) and because of its restrictiveness. The GPL license limits commercial adoption of software in a very restrictive way. Onetti and Verma state that the copyleft licences like GPL, are used by OSS companies that have a so-called reciprocal business model where the revenue come from professional services (p.20). In order to use the GPL license, it is important that the revenue is not originating from licensing fees. The GPL license allows modification of the code but redistribution of the modified software is only allowed if this version is also distributed under the GPL license. This means that perceived usefulness could be reduced because it may "[...] be reduced by restrictive licenses in that such licenses limit potential users' ability to employ the code in conjunction with software distributed under a less restrictive license." (Stewart et al., 2006, p.130). This explains the number of cooperation partners with which the OSS companies work together with.

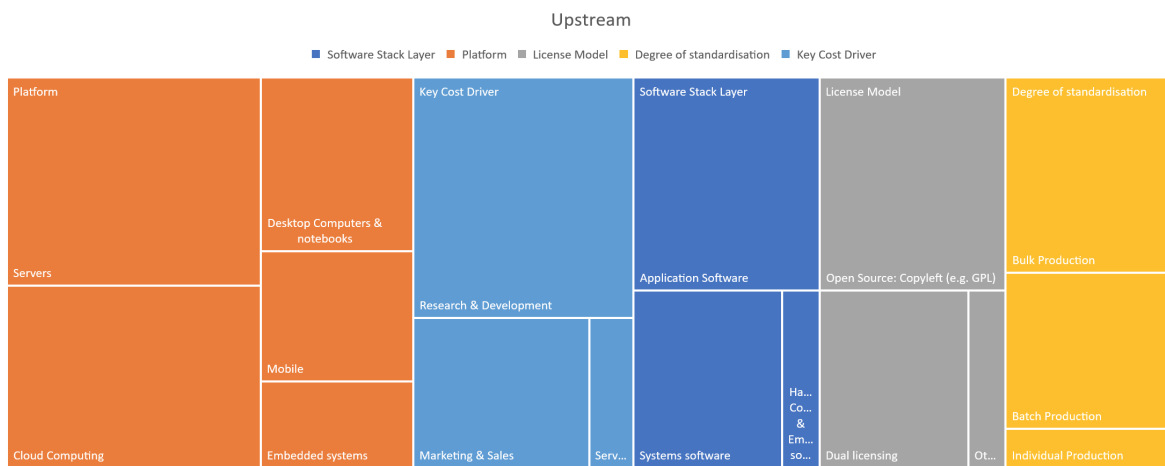


Figure 3.4: Tree-map of the SBMF outcomes, Upstream

Companies cooperate to make sure that the products are compatible and integrated. Interviewee of company G. says that the advantage of this is two-fold; they let the industry-partner build the integration and make sure it is working in the major version of the OSS software and in return there are proper integration and services available. Four OSS companies offer both open source solutions and a proprietary solution next to each other. This complies with the dual license business model where a company markets the software product with the choice of either an open source licensed software product or a commercial/proprietary one (Onetti & Verma, 2008) as described in Chapter 2. Company E. uses the MIT open source license, which is a non-restrictive license with open possibilities to change and use the code (*The MIT license*, n.d.). The *degree of standardization* shows whether a company offers standardized or customized software (Schief, 2013). Schief states that individual production means that the product is customized based on specific customer needs. Software which is produced in bulk can be reused by any customer. Five companies produce their software in bulk and four in batch. Company E. addresses that they never make something specially for one customer. When the customer indicates that they want a certain function, the interviewee asks which problem they want to see fixed and sees if he can create something that is interesting for more clients, so in bulk. Only company D. produces individual products for specific clients. Company I. creates a revenue stream of extra modules that companies want that are not available in the open source version. These are usually industry specific modules that a group of their target users need, so it is not fully customized but focused on a certain (paying) group of users. The interviewee of company C. explains that 90% of their software is standardized but that they sometimes have to be flexible because their

clients are essential in their business model.

The final sub-component focuses on the *dominating cost driver* within the company. Eight out of ten companies mostly spend money on Research & Development, so the personnel costs for the development of the software. This complies with what is stated in Chapter 2 that the majority of the open source developers are being paid for their work. The Interviewee of company G. admits that they hire developers from the open source community to spend a certain amount of their time on the product. The interviewee says that the developers like this because they usually use their spare time to write code but now use that time to spend it with their families. This shows that OSS companies use free labour of the community and combine this with the work of in-house developers. They do this to ensure that the project is not dependent on the work of the community. Besides the research and development costs, companies B., G. and J. admit that they also spend a lot of money on marketing.

### 3.2.4 Downstream

The *Downstream* component is focused on the customers by characterizing the target market and distribution areas. The *localization* of the company shows the geographical areas a company is targeting, which is in this case mostly all areas. But the internationalization of companies is a gradual development where the distant markets are gradually targeted (Moen et al., 2004). This can also be drawn from the history of the OSS companies, many of them started offering their services and solutions in the country of the headquarters. The Internet created this border-less business platform (Benjamin & Wigand, 1995) and the high internationalization can be seen as a signature of software companies (Hoch et al., 2000).

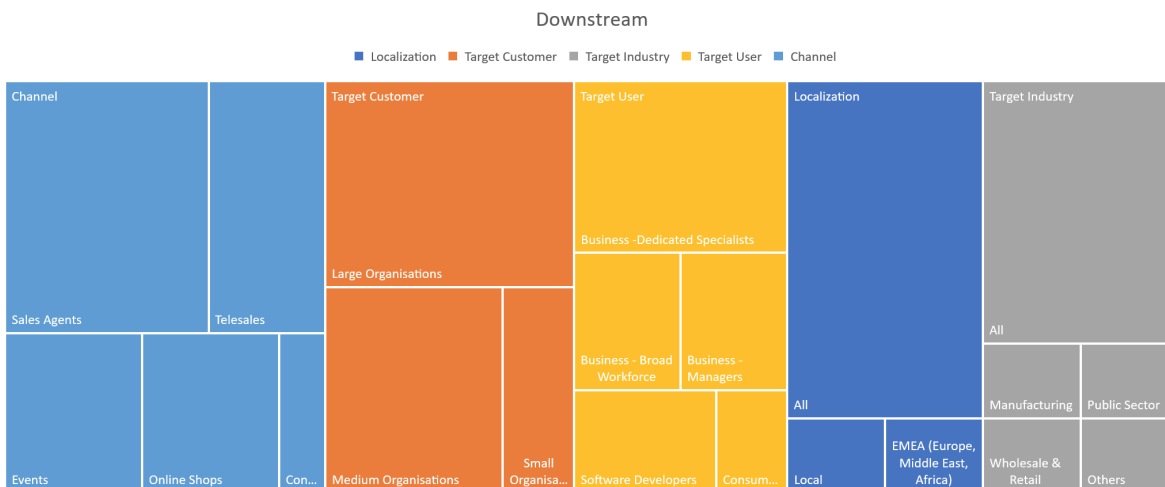


Figure 3.5: SBMF outcomes - Downstream

The *target customer* is categorized into small (up to 50 employees), medium (51-250) and large (>250) sized companies. Figure 3.5 shows that the sample mostly focuses on large organizations. Company E. explains that as a B2B company you earn the most money with offering your product to large clients. That is also why many of the OSS products have an open source solution and a separate enterprise product to offer companies that are willing to pay. Company B. adds that they offer a product which is a solution to problems of large complex companies and therefore have created a market around these large sized customers. The interviewee of company I. sees that the open source product is often used by smaller companies, while their focus lies on the larger ones agreeing with company E. that these companies are willing to pay. This money is not only necessary to grow in revenue but also to develop the enterprise products. Interviewee A. states that you need success with smaller companies to convince the larger ones, therefore the company is currently focusing on small and medium sized companies. Company J. admits not having an explicit target customer due to the fact that each customer has a different need. To comply with these needs, company J. has many

different solutions and service offerings which are usable for a broad range of customers. Although most companies are focusing on the large customers, they mostly agree on the fact that the types of customers differ.

The *target industry* component is obtained from the Standard Industrial Classification (SEC, 2013). The option "All", which is mostly picked, conceptualizes companies which sell horizontal solutions that can be applied in any chosen industry (Schief, 2013). This is due to the fact that most of the companies develop application software usable for clients in different fields. Having a broad target industry creates a large available market with the possibility of a constant growing revenue. According to the author, the *target user* is the initial user of the software and specifies for what type of user the software is designed for. The types of users are derived from Cotterman & Kumar (1989, p.1316) which ranges from consumers to developers. Therefrom, the three business focused users mentioned by Schief are: Broad workforce, Dedicated specialists, Managers and additionally the users "consumers" and "software developers" are portrayed. Most software is designed for dedicated specialists who have specific knowledge about certain processes and need a specific type of software to perform their roles. The *sales channel* through which the software is marketed to prospective customers is mostly through sales agents. The companies all have specific sales departments which are focused on acquiring new international customers. Next to the options provided by Schief, most companies admit being very active in the field of digital marketing through online advertisement.

### 3.2.5 Usage

The final component *Usage* is made up of service oriented components which, according to Schief, are needed during the use of any software solution. The *implementation effort* is based on the effort that is required to install and configure the software. The software of the stakeholders seems mostly to be medium since the customers will not always be able to install the software themselves and might need some assistance. Some of the interviewees admit that especially the open source versions seem to be harder to install if the user has no prior knowledge of software.

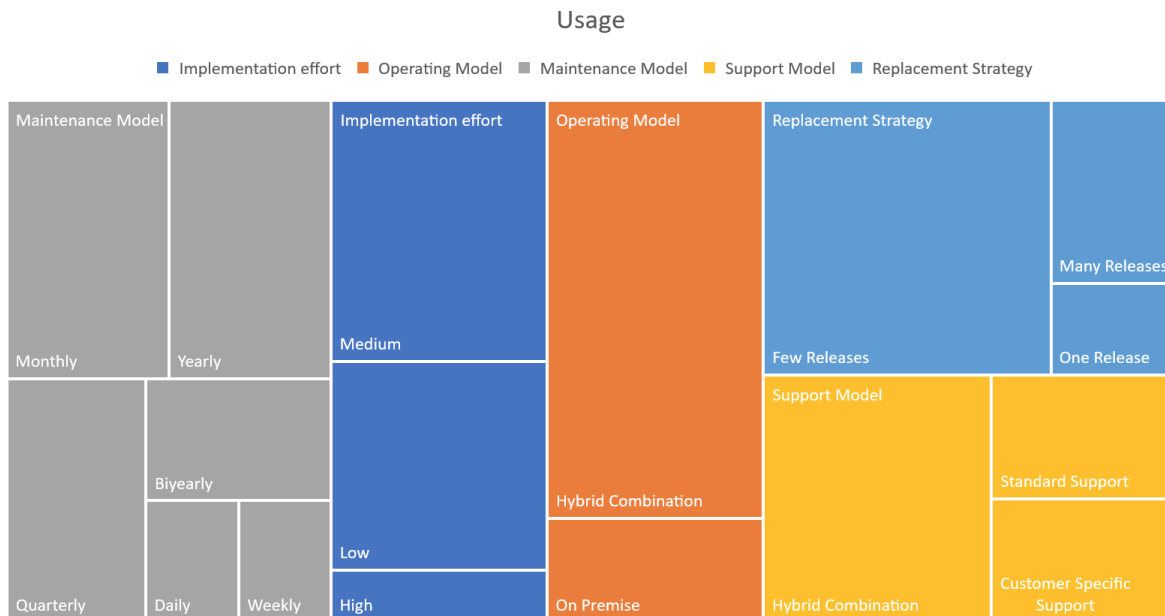


Figure 3.6: Tree-map of the SBMF outcomes - Usage

The *operating model* on which the software is deployed differentiates between two main deployment models: on premise and on demand. On premise needs installation and execution on the local systems of the customers while on demand is based on a central hosting platform offered by the software company (Buxmann et al., 2013). The hybrid combination of both on premise and local systems

is the most often chosen option because the software companies of the sample offer different types of software which all have a specific deployment model. The installation and integration services are sometimes needed and can be paid for as an extra service. The *maintenance model* is based on the release frequency of the software. In this case a release represents a major update of a software version. As shown in Figure 3.6, the answers are scattered due to the amount of software solutions the companies offer, but overall most of them have a monthly or yearly major release. Company C. offers maintenance contracts that are paid for every month.

A company's *support model* depends on the support contracts signed by the customer (Buxmann et al., 2013). Standard support is the same for all customers while the customer specific support model offers a customized individual support contract. The majority of the sample offers different types of support contracts. This mainly has to do with the fact that OSS companies cannot ask money for the software, therefore business models of OSS companies are primarily built around software related services (Whichmann, 2002). Customer specific support is also often needed because of the diversity of the customers and is based on the underlying Service Level Agreements (SLAs) offered by the software vendor. Prices are also based on these service, like for company E. who offers an SLA of 4 hours for the more expensive subscriptions and a cheaper one for the SLA of 24 hours. Company I. noted that large customers have big integration projects preceding the installation of the software and therefore need "professional services" to adjust it to their software architecture and networks. These large customers are specifically looking for software companies that offer these support and services contracts so they do not have to do it themselves. The interviewee adds that this is the biggest difference with their open source version of the software: the open source version has to be installed by the user and if a problem occurs, they have to look for a solution within the forums maintained by the developer community. The final sub-component is the *replacement strategy* and is based on the number of available software product releases at a time (Jansen et al., 2011, p.2-4). No company just deals with one release, only two companies have many releases, and the majority has few releases which means some major release versions are used by the customers.

# Chapter 4

## Findings

This final chapter describes the main findings of the case study with discussion in perspective to existing knowledge about OSS business models. After examining the transcripts and frameworks we can answer the three research questions and make statements based on the knowledge we have gained. The first question to be answered is: **What are the determinants of success of open source business Models?** Many definitions of business success are discussed in literature and accordingly success is mostly measured by revenue, but can not be solely used as an indicator. Based on the investment horizon component of the framework we can state that the sample is looking for growth in company size, number of customers and community size. This disagrees with the findings of Bonaccorsi et al. (2006) who states that Open Source is a production paradigm that does not support company growth. Since the sample consists of mostly medium and large companies we can state that these companies have grown since the start-up phase while offering an OSS product. Therefore this research bases success on whether the goals of a business are met or being worked towards by growing.

The second research question is: **What synergies between business model characteristics create a blueprint for business success?** Data based on interview transcripts and completed Software Business Model frameworks show several correlations between combinations of business model components. First we recapitulate the main findings from the completed SBMFs. In terms of strategy, the OSS companies have quality as their value proposition which correlates with development as their most important value chain. The degree of vertical integration of the development is medium and is shared with few cooperation partners. All-together the investment horizon leans towards a growth model. The sales volume of the sample differs between medium and high but comes directly from the source. The pricing assessment base is a hybrid combination of usage-based and usage-independent pricing. We recognize a subscription model in the recurring payment flow structure. The revenue distribution is medium, which means it is not shared with many shareholders. The definition and development of the OSS is covered in the upstream component of the SBMF. We see that mostly the companies offer Application software, with on the second place Cloud Computing software. The platform on which it is offered is a server and the product is usually standardized in bulk. The license model used on the open source part of the offered product is a Copyleft, like GPL, model. Nevertheless, some of the companies offer both proprietary and OSS simultaneously by applying a dual licensing model. The key cost driver is based on costs spent on Research & Development. The downstream component is focused on the customers of the OSS companies. The localization is mostly worldwide and the type of customers are typically large organizations (>250 employees) from varying industries. The target end-users are mostly Business dedicated specialists. The marketing channel is maintained by sales agents. The usage component is composed of the offered services. Implementation is one of these services, and the effort implementing the software is medium according to most interviewees. The operating model is a hybrid combination of on premise and on demand, based on what the customer prefers. Also the support model is a hybrid combination, the choices for customers are standard- or customer specific support. The OSS has few releases and mostly monthly or yearly maintenance moments. By applying the SBMF we can state that *OSS business components*

can be drawn from the Software Business Model Framework.

The business models recognized in the data can be compared to OSS business models found in literature. As covered in chapter 2 we can depict three types of OSS business models: Direct-sales value models, Indirect-sales value models and Use-value models. A review of the characteristics of these types of business models enables us to compare them to the case study data. After reviewing the interview data it is noticed that none of the companies from the sample use a Use-value model. Nevertheless, we have recognized several elements of the Direct- and Indirect-sales value models in the business models of the OSS companies. Therefore we can state that *OSS business model components found in literature are recognized in practice*. Direct-sales value models are focused on the dual-licensing of software, meaning that a proprietary license is used next to an open source license. Companies B., H. and I. use a dual license model to offer companies either an open source version or a commercial enterprise version of their software. Based on this outcome we can partially agree with the assumption found in literature that *dual licensing is an often chosen business model for OSS businesses*. According to Popp (2012) there are two ways to apply this type of license: 1. offering a dual product with dual licensing or 2. a dual license for identical products. The author states that customers sometimes choose a commercial license to ensure they get a warranty or liability from the OSS vendor. This idea concurs with the choice of OSS license of the sample (GPL) whereas we disagree with the statement found in literature that *OSS business use less restrictive open source licenses*. The use of a copyleft license limits commercial adoption of software in a very restrictive way and fits in Onetti and Verma's 2008 Reciprocal business model. The authors allocate this model to OSS business that use a copyleft license while generating revenue from professional services. This revenue does specifically not originate from licensing fees but adopting this model does not mean it is a pure service company (p.21).

Within the Indirect-sales value models, a consistent part of the business models of the sample is based on the offering of an OSS core, and additional modules or functionalities that are specifically interesting for medium- and large sized enterprises. This is recognized by Chesbrough & Appleyard (2007, p.65) who describe these companies that choose an OSS license to help proliferate the product and concurrently provide "enterprise" versions to paying customers which are generally more stable. This business model focuses on the development of proprietary extensions or add-ons and are identified in the sample. This versioning of an OSS product is different than the dual licensing as mentioned in research discussed in Chapter 2, because the license stays Open Source, so free of charge. Nevertheless, the extra modules that are not covered in the open source version are available after an overcharge is paid by the customer. This is also applied on the specific services and support models offered for enterprises. The so-called enterprise editions form a contradictory business model category called "commercial open source business models" (Riehle, 2009; Popp, 2012). The definition of companies that apply this model is given by Riehle (2009, p1.): "Commercial OSS projects are owned by a single firm that derives a direct and significant revenue stream from the software." In the commercial open source business model, commercial OSS companies foremost focus on providing services around the software product (Popp, 2012). Research by Popp (2012, p.34) shows that services like maintenance and integration are provided based on a commercial support contract and specified in a SLA next to the OSS. Using a commercial OSS business model gives the company a competitive advantage against proprietary software companies since they get to market faster with a product offered at a lower cost but with excellence services (Riehle, 2009). It is notable that the majority of the sample offers their clients SLAs next to the open source or built in a commercial version. We can see a correlation between the medium implementation effort and companies that offer services, meaning that clients that have no background in IT are willing to pay for the installation of the OSS. These offerings of enterprise editions including implementation, integration and/or support seem to form, what the interviewees call, a subscription model. This model is based on the correlations between a recurring payment flow, a monthly maintenance model for a bulk product marketed within a broad target industry and international localization. Additionally, the companies that share these same focus points offer customer specific support or a hybrid combination. Two companies of the sample operate as original Linux Distributor as mentioned by Whichmann (2002, 42) who package and sell their own version of the Linux software in various packages. The payment flow structure used by the sample is either recurring or hybrid which is a characterizing feature of a subscription. Research by

Zhang & Seidmann (2010) states that the subscription model lowers cost of ownership and therefore that starters do not need large up-front investments. The author sees the benefit for the customer in the constant access to up-to-date software at a predictable cost (p.40). Furthermore this business model is not recognized in any other software business model literature or OSS literature.

Additionally, we see that the degree of standardization is distributed where five companies offer bulk software and four batch. What got our attention is that four out of five bulk software is cloud computing software. The same correlation cannot be found between bulk software and servers but what we can see is that three out of four batch software has a recurring pay structure. All-together, the SBMF data shows that the sample has corresponding components that form a particular business model. This combination can be mold into a blueprint for a hybrid open source business model (see Figure 4.1). The companies from the sample can be categorized as commercial open source vendors. Popp (2012, p.36) states that these vendors do not have to implement the same business models to be categorized as such and to be successful, but they can build a unique hybrid business model by selecting a subset of known business model components. This specific combination of business model components is based on the success of the sample and forms the proposed blueprint revealing the building blocks of their open source business model. This figure affirms the statement that *a blueprint can be created for the establishment of an OSS business.*

Strategy				
<b>Value proposition:</b> Quality	<b>Investment horizon:</b> Growth model	<b>Value Chain:</b> Development	<b>Degree of vertical integration:</b> Medium	<b># of cooperation partners:</b> Few
Revenue				
<b>Sales volume :</b> Medium High	<b>Revenue source:</b> Direct	<b>Pricing assessment base:</b> Hybrid combination	<b>Payment flow structure:</b> Recurring	<b>Revenue distribution model:</b> Medium
Upstream				
<b>Software stack layer:</b> Application software	<b>Platform:</b> Servers	<b>License model:</b> Open source - Copyleft	<b>Degree of standardisation:</b> Bulk production	<b>Key cost driver:</b> Research & Development
Downstream				
<b>Localization:</b> All	<b>Target customer:</b> Large organisations	<b>Target industry:</b> All	<b>Target user:</b> Business-dedicated specialists	<b>Channel:</b> Sales agents
Usage				
<b>Implementation effort:</b> Medium	<b>Operating model:</b> Hybrid combination	<b>Maintenance model:</b> Monthly Yearly	<b>Support model:</b> Hybrid combination	<b>Replacement strategy:</b> Few releases

Figure 4.1: A OSS business model blueprint.

The answer to question two adds convenient knowledge to the final research question. Question three is: **Which factors lead to potential business success for an open source business?** This question is answered by a combination of the constructs of the blueprint and the insider tips directed to



entrepreneurs given by the interviewees. We conclude with the most affirmative answers :

- Find a good balance between business oriented employees and open source developers.
- Use the open source community to find developers that can work in-house on your product.
- Funding is not necessary but creates possibilities, otherwise look for incubators.
- Create a clear vision for your company and establish your differentiators.
- Stick to the open source idea, this will keep the community close to the company.
- When the community does not kick-off, start with offering proprietary software and make it open source after a while.
- Communicate frequently with your clients to establish their needs.
- Decide your revenue stream: Services, support, dual licensing, enterprise editions etc.
- Medium and large customers are willing to pay large amounts of money for services, so do not sell yourself short by offering services for low prices.

We can align the stories of the interviewees as stated in Chapter 4 with the following statement: *It is possible to create an OSS start-up without having an existing open source community.* We have seen that companies from the sample started offering a proprietary product and opened the product later in the development process of the product. Nevertheless, not all interviewees agree on the benefits of Open Source and whether offering an OSS product will be sufficient for start-ups to grow. Nevertheless, these companies have grown into successful companies and Bonaccorsi et al. (2006, p.19) confirm that Open Source "makes it possible for small firms to be innovative and find sustainable revenue streams". As depicted in Chapter 3, literature covered several strategies to establish a start-up where the blueprint and guideline are there to be used as an aid in this process. The sample data shows us that not all companies from the sample have received investments. Additionally, none of the stakeholders has mentioned investments as a prerequisite for a start-up to become successful. Therefore we disagree with the statement that *OSS businesses rely on investments because they offer free software.* This has also to do with the value exchanges between the different stakeholders in a OSS business model. As Lindman & Rajala (2012, p.6) state; "[In OSS businesses] resources are accessed through collaborative relationships between two or more parties". Reviewing the e3-value model based on these relations in Figure 2.5 we can state that *different value streams are needed to create a successful OSS company.* Moreover, the interview transcripts and the e3-value model shows us that *OSS enables a growing customer base* because the customers are not the only users. The developers from the OSS community are often partially working on the product for their own benefit. When the community grows, the customer base is also grows. Additionally it is known that OSS can be offered for a lower price since the software is provided by an outside party (Lindman & Rajala, 2012) and in turn low software prices attract new customers. The OSS contributes in another way, where we can state that *the strategic advantage of OSS is the already existing network bounds offered by the communities.* Interviewees agree with the idea that being Open Source accommodates the establishment of a business and its OSS product. We can conclude that no matter which OSS business model is chosen, a business can choose to make the source code freely available and still serve its business interests as a for-profit organization (Hecker, 1999, p.46).

# Chapter 5

## Discussion

The research findings give us new perceptions on the building blocks and success factors of OSS business models. The literature study has shown us that business models are widely deployed as a conceptual tool to reflect on realized strategies and offered values of businesses. The use of the Software Business Model Framework of Schief (2013) has given us the opportunity to classify business models of OSS businesses. Nevertheless, this framework has so far only been applied to the same type of (non-open source) software businesses. Therefore we can question the applicability of this specific framework on OSS business models. Additionally, we suggest adjustments to the framework to make it up-to-date and applicable for more specific types of business models.

- Sales channel; The interviewees agreed on having sales agents who accommodate the sales and marketing department, but additionally mention their use of online advertising to market their products. Details on the type of the (online) advertisements could give an indication of the revenue creation initiated by advertising.
- Support model; details around the support model could be specified by looking at the specific price modules, payment flow structure, if it is based on a stand-alone subscription model or part of a large set of services in- or excluding the product. This is also interesting for the construction of their revenue stream, where we could review how much of the revenue is coming from services versus from the software product itself.
- Support model; most companies from the sample do not have only one but plural software products applying different SLAs and licenses. This confused the interviewees during the process of completing the framework. Defining sub-components in the support model section could make it possible for the practitioners to fill in the different options for each of their products.

Nevertheless, we are able to define a blueprint applicable in the OSS field. The blueprint shows which OSS business model has worked best for most companies of the sample. Businesses usually follow a hybrid business model according to the market they are in and the demand of current or potential customers. Therefore it has to be mentioned that the applicability of the blueprint is not a guarantee for success due to external factors. This brings us to the threats to validity of this work. Wohlin et al. (2012) mention threats to conclusion internal, external and construct validity that can be recognized in studies. In this case study we find a threat to conclusion validity in the way the interviews are conducted because random irrelevancies in the experimental setting can disturb the results. The interviews were held through phone so random events in the environment of the interviewee could disturb the completion of the framework by not paying adequate attention. Additionally, we can find a threat to validity in the construction of the sample and its size. The chosen sample criteria reduces the heterogeneity since the interviewees are in the same market, offering a similar product. At the same time, it reduces the external validity of the study since the subject is not selected from a general enough population. Time constraints causes the sample to cover 10 B2B open source businesses. Conclusions can be drawn from this sample but a larger sample size could add knowledge and solid statements, therewith building stronger validity. The size of the sample is also due to the gathering

of the sample which is a known problem in Case Study research. The expert employees are all fulfilling time-consuming roles within the company which leads to the lack of time to participate in research like this one. Future research can build on the blueprint for OSS start-ups by interviewing a larger sample size. To make the gathering of the data less time-consuming, a questionnaire could replace or be added to the interviews. When this method is used, the data outcomes will be based on quantitative instead of qualitative measurements which eliminates threats to validity.

Threats to internal validity concern how the subjects are treated during the experiment and if special events occur during the experiment. In this case study, the use of the SBMF could threaten the internal validity because of the interviewees' lack of knowledge in the use of the framework. Threats to the construct validity can be found in the extent to which the experiment setting, in this case B2B OSS businesses, reflects the construct under study. The sample could be expanded to B2C companies by applying additional sample criteria. External validation of the blueprint can be expanded by applying it on the business models of OSS start-ups. To eliminate the threat on external validity, the blueprint can be put into use on businesses that do not have a business model yet, or on businesses that want to change their current business model and strategy. Future research could possibly focus on the influence of the community on the business model and the development of the product. Ultimately, more research on OSS business models for start-ups is needed. The field of OSS, start-up strategies and investments should be further explored to add to the current body of knowledge.

## Chapter 6

# Conclusion

We have seen that the OSS phenomenon has metamorphosed into a more mainstream and commercially viable product (Fitzgerald, 2006). We recognized the lack of a comprehensive theoretical framework to analyze how businesses use this product to create revenue. Additionally, we found a gap in literature for prospective practitioners that try to establish an OSS business. In this study we examined and compared the business models of 10 B2B OSS companies. Business models have proven to be useful as a conceptual tool to analyze the revenue logic. The comparison was established by using the Software Business Framework of Schief (2013) for the interview protocol and analysis of the business models. The Case Study Protocol of Pervan & Maimbo (2005) was followed to gather and analyze the data. In-depth interviews with experts in the field gave us directly insider information on OSS business models and experiences in the Open Source ecosystem. Knowledge gained from the interview transcripts and completed frameworks showed some unexpected outcomes. The statements from findings in Chapter 4 show what we have learned from the current business models of OSS businesses. For example the very restrictive open source license GPL was used by most companies. Additionally, the opinions of the interviewees differ when it comes to the advantages and disadvantages of establishing a start-up around an open source product. The information provided by the interviewees has showed us that it is not harder to establish a business around OSS than around a proprietary product. We have seen that different value exchanges exist in the product development process of an OSS product and that it enables a growing customer base. When it comes to their business models, we can conclude that the sample companies lean toward a hybrid model where the combination of a diversity of SBMF components are applied which together form a blueprint for OSS start-ups. We can state that the OSS companies have a b2b focus. Compared to literature, the Direct-sale value models of E. Raymond (1999) cover the category "commercial open source business models" of Riehle (2009). Overall, the interviewees call their business models "subscription models" but this is not a complete representation of the exact building blocks.

We can place the applicability of this research in the field of OSS software, entrepreneurship, OSS businesses and strategy. The blueprint and guideline together are useful for entrepreneurs who want to start a business around an existing open source project, or for who want to change their business model when growth is not part of the long-term picture anymore. Furthermore, investors can compare the business model of prospective start-ups they want to invest. Overall, this research provides an overview of the current used business models in the OSS market and mentioned in literature. This overview is offered in the shape of a blueprint for current OSS practitioners, start-ups and researchers. This work is aiming to establish a new field in literature focusing on the inside information on OSS business models and the applicability for start-ups.

# References

- Androutsellis-Theotokis, S., Spinellis, D., Kechagia, M., Gousios, G., et al. (2011). Open source software: A survey from 10,000 feet. *Foundations and Trends® in Technology, Information and Operations Management*, 4(3–4), 187–347.
- Baxter, P., & Jack, S. (2008). Qualitative case study methodology: Study design and implementation for novice researchers. *The qualitative report*, 13(4), 544–559.
- Beaver, G. (2002). *Small business, entrepreneurship and enterprise development*. Pearson Education.
- Benjamin, R., & Wigand, R. (1995). Electronic markets and virtual value chains on the information superhighway. *Sloan Management Review*, 36(2), 62.
- Blank, S. (2013). Why the lean start-up changes everything. *Harvard business review*, 91(5), 63–72.
- Bonaccorsi, A., Giannangeli, S., & Rossi, C. (2006). Entry strategies under competing standards: Hybrid business models in the open source software industry. *Management Science*, 52(7), 1085–1098.
- Bonaccorsi, A., & Rossi, C. (2003). Why open source software can succeed. *Research policy*, 32(7), 1243–1258.
- Bonaccorsi, A., & Rossi, C. (2006). Comparing motivations of individual programmers and firms to take part in the open source movement: From community to business. *Knowledge, Technology & Policy*, 18(4), 40–64.
- Boote, D. N., & Beile, P. (2005). Scholars before researchers: On the centrality of the dissertation literature review in research preparation. *Educational researcher*, 34(6), 3–15.
- Bosch, J., Olsson, H. H., Björk, J., & Ljungblad, J. (2013). The early stage software startup development model: a framework for operationalizing lean principles in software startups. In *Lean enterprise software and systems* (pp. 1–15). Springer.
- Burkhart, T., Krumeich, J., Werth, D., & Loos, P. (2011). *Analyzing the business model concept—a comprehensive classification of literature* (Vol. 12). ICIS 2011 Proceedings.
- Buxmann, P., Diefenbach, H., & Hess, T. (2013). *The software industry: Economic principles, strategies, perspectives*. Springer Science & Business Media.
- Casadesus-Masanell, R., & Ricart, J. E. (2010). From strategy to business models and onto tactics. *Long range planning*, 43(2), 195–215.
- Chang, V., Mills, H., & Newhouse, S. (2007). From open source to long-term sustainability: Review of business models and case studies. In *Proceedings of the uk e-science all hands meeting 2007*.
- Chesbrough, H. W., & Appleyard, M. M. (2007). Open innovation and strategy. *California management review*, 50(1), 57–76.
- Comino, S., Manenti, F. M., & Parisi, M. L. (2007). From planning to mature: On the success of open source projects. *Research Policy*, 36(10), 1575–1586.

- Cooper, D. R., Schindler, P. S., & Sun, J. (2006). *Business research methods* (Vol. 9). McGraw-Hill Irwin New York.
- Cotterman, W. W., & Kumar, K. (1989). User cube: a taxonomy of end users. *Communications of the ACM*, 32(11), 1313–1320.
- Crowston, K., Annabi, H., & Howison, J. (2003). Defining open source software project success. *Former Departments, Centers, Institutes and Projects*, 4.
- Crowston, K., Howison, J., & Annabi, H. (2006). Information systems success in free and open source software development: Theory and measures. *Software Process: Improvement and Practice*, 11(2), 123–148.
- Daffara, C. (2007). Business models in floss-based companies. In *Workshop presentation at the 3rd conference on open source systems (oss 2007)* (pp. 1–8).
- Dahlander, L. (2005). Appropriation and appropriability in open source software. *International Journal of Innovation Management*, 9(03), 259–285.
- Dubey, A., & Wagle, D. (2007). Delivering software as a service. *The McKinsey Quarterly*, 6(2007), 2007.
- Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of management review*, 14(4), 532–550.
- Endo, P. T., Gonçalves, G. E., Kelner, J., & Sadok, D. (2010). A survey on open-source cloud computing solutions. In *Brazilian symposium on computer networks and distributed systems* (Vol. 71).
- English, R., & Schweik, C. M. (2007). Identifying success and tragedy of floss commons: A preliminary classification of sourceforge. net projects. In *Emerging trends in floss research and development, 2007. floss'07. first international workshop on* (p. 11-11).
- Feller, J., Fitzgerald, B., et al. (2002). *Understanding open source software development*. Addison-Wesley London.
- Fernhaber, S. A., McDougall, P. P., & Oviatt, B. M. (2007). Exploring the role of industry structure in new venture internationalization. *Entrepreneurship Theory and Practice*, 31(4), 517–542.
- Fitzgerald, B. (2006). The transformation of open source software. *Mis Quarterly*, 587–598.
- Fogel, K. (2005). *Producing open source software: How to run a successful free software project*. " O'Reilly Media, Inc."
- Gagliardi, A. R., Brouwers, M. C., Palda, V. A., Lemieux-Charles, L., & Grimshaw, J. M. (2011). How can we improve guideline use? a conceptual framework of implementability. *Implementation Science*, 6(1), 26.
- Giardino, C., Unterkalmsteiner, M., Paternoster, N., Gorschek, T., & Abrahamsson, P. (2014). What do we know about software development in startups? *IEEE software*, 31(5), 28–32.
- Goldman, R., & Gabriel, R. P. (2005). *Innovation happens elsewhere: Open source as business strategy*. Morgan Kaufmann.
- Gordijn, J., & Akkermans, J. (2003). Value-based requirements engineering: exploring innovative e-commerce ideas. *Requirements engineering*, 8(2), 114–134.
- Grimshaw, J., Thomas, R., MacLennan, G., Fraser, C., Ramsay, C., Vale, L. e. a., ... others (2004). Effectiveness and efficiency of guideline dissemination and implementation strategies. , 1–72.
- Hawkins, R. E. (2004). The economics of open source software for a competitive firm. *NETNOMICS: Economic Research and Electronic Networking*, 6(2), 103–117.

- Headd, B. (2003). Redefining business success: Distinguishing between closure and failure. *Small business economics*, 21(1), 51–61.
- Hecker, F. (1999). Setting up shop: The business of open-source software. *IEEE software*, 16(1), 45–51.
- Hellmann, T., & Puri, M. (2002). Venture capital and the professionalization of start-up firms: Empirical evidence. *The journal of finance*, 57(1), 169–197.
- Hippel, E. v., & Krogh, G. v. (2003). Open source software and the “private-collective” innovation model: Issues for organization science. *Organization science*, 14(2), 209–223.
- Hoch, D. J., Roeding, C., Lindner, S. K., & Purkert, G. (2000). *Secrets of software success*. Harvard Business School Press Boston.
- Hope, J. (2007). Open source licensing. In A. Krattiger, R. Mahoney, & e. a. Nelsen L. (Eds.), *Intellectual property management in health and agricultural innovation: A handbook of best practices* (Vol. 1). Oxford.
- Huemer, C., Schmidt, A., Werthner, H., & Zapletal, M. (2008). A uml profile for the e3-value e-business model ontology. *Proceedings of the Third International Workshop on Business/IT Alignment and Interoperability (BUSITAL'08) held in conjunction with CAiSE'08 Conference*, 336(1).
- Iansiti, M., & Levien, R. (2004). Strategy as ecology. *Harvard business review*, 82(3), 68–81.
- Jansen, S., Popp, K. M., & Buxmann, P. (2011). The sun also sets: ending the life of a software product. In *International conference of software business* (pp. 154–167).
- Jennings, P., & Beaver, G. (1997). The performance and competitive advantage of small firms: a management perspective. *International small business journal*, 15(2), 63–75.
- Johnson, B., & Shneiderman, B. (1991). Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd conference on visualization'91* (pp. 284–291).
- Kennedy, D. M. (2001). A primer on open source licensing legal issues: copyright, copyleft and copyfuture. *Louis U. Pub. L. Rev.*, 20, 345–380.
- Koenig, J. (2004). Seven open source business strategies for competitive advantage. *IT Manager's Journal*, 14.
- Krishnamurthy, S. (2005). An analysis of open source business models. *Making sense of the Bazaar: Perspectives on Free and Open Source Software*, 279–296.
- Krumeich, J., Burkhart, T., Werth, D., & Loos, P. (2012). Towards a component-based description of business models: a state-of-the-art analysis.
- Lakhani, K. R., & Wolf, R. G. (2003). Why hackers do what they do: Understanding motivation and effort in free/open source software projects.
- Lambert, S. C., & Davidson, R. A. (2013). Applications of the business model in studies of enterprise success, innovation and classification: An analysis of empirical research from 1996 to 2010. *European Management Journal*, 31(6), 668–681.
- Lee, S.-Y. T., Kim, H.-W., & Gupta, S. (2009). Measuring open source software success. *Omega*, 37(2), 426–438.
- Lerner, J., & Tirole, J. (2001). The open source movement: Key research questions. *European economic review*, 45(4), 819–826.
- Lerner, J., & Tirole, J. (2002). The simple economics of open source. *The journal of industrial economics*, 50(2), 197–234.

- Lerner, J., & Tirole, J. (2005). The scope of open source licensing. *Journal of Law, Economics, and Organization*, 21(1), 20–56.
- Letellier, F. (2008). Open source software: the role of nonprofits in federating business and innovation ecosystems. (a submission for AFME 2008).
- Li, J., & Liu, W. (2013). Selecting a target segment: market structure and new venture entry strategies. *Management Decision*, 51(7), 1402–1421.
- Lichstam, M., & Johansson, B. (2016). Requirements engineering in startups with open source software related business strategies. *LU-CS-EX 2016-38*.
- Lindman, J., & Rajala, R. (2012). How open source has changed the software industry: Perspectives from open source entrepreneurs. *Technology Innovation Management Review*, 2(1).
- Madill, J. J., Haines, G. H., Jr, & Rlding, A. L. (2005). The role of angels in technology smes: A link to venture capital. *Venture Capital*, 7(2), 107–129.
- Midha, V., & Palvia, P. (2012). Factors affecting the success of open source software. *Journal of Systems and Software*, 85(4), 895–905.
- The mit license*. (n.d.). Retrieved 2018-05-01, from <https://opensource.org/licenses/MIT>
- Moen, Ø., Gavlen, M., & Endresen, I. (2004). Internationalization of small, computer software firms: Entry forms and market selection. *European Journal of Marketing*, 38(9/10), 1236–1251.
- Morris, M., Schindehutte, M., & Allen, J. (2005). The entrepreneur's business model: toward a unified perspective. *Journal of business research*, 58(6), 726–735.
- Nobel, C. (2011). Teaching a 'lean startup' strategy. *Harvard Business School Working Knowledge*, 1–2.
- Onetti, A., & Capobianco, F. (2005). Open source and business model innovation. the funambol case. In *Proceedings of the first international conference on open source systems* (pp. 224–227).
- Onetti, A., & Verma, S. (2008). *Licensing and business models* (Tech. Rep.). Department of Economics, University of Insubria.
- The open source definition*. (1999). Retrieved 2017-10-19, from <https://opensource.org/docs/definition.php>
- Oreg, S., & Nov, O. (2008). Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values. *Computers in human behavior*, 24(5), 2055–2073.
- Osterwalder, A. (2004). The business model ontology: A proposition in a design science approach.
- O'Mahony, S. (2005). 20 nonprofit foundations and their role in community-firm software collaboration.
- Pal, N., & Madanmohan, T. (2002). Competing on open source: Strategies and practise. *Bangalore: Indian Institute of Management Bangalore*.
- Paternoster, N., Giardino, C., Unterkalmsteiner, M., Gorschek, T., & Abrahamsson, P. (2014). Software development in startup companies: A systematic mapping study. *Information and Software Technology*, 56(10), 1200–1218.
- Perens, B. (2005). The emerging economic paradigm of open source. *First Monday*.
- Perr, J., Appleyard, M. M., & Patrick, P. (2010). Open for business: emerging business models in open source software. *International Journal of Technology Management*, 52(3/4), 432–456.
- Pervan, G., & Maimbo, M. (2005). Designing a case study protocol for application in is research. In *Proceedings of the ninth pacific asia conference on information systems* (pp. 1281–1292).



- Popp, K. M. (2012). Leveraging open source licenses and open source communities in hybrid commercial open source business models. In *Iwseco@ icsob* (pp. 33–40).
- Raymond, E. (1999). *The magic cauldron*. Thyrus Enterprises.
- Raymond, E. (2001). *The cathedral & the bazaar: Musings on linux and open source by an accidental revolutionary*. " O'Reilly Media, Inc."
- Raymond, E. S., et al. (2000). Open sources: Voices from the open source revolution.
- Riehle, D. (2009). The commercial open source business model. In *Value creation in e-business management* (pp. 18–30). Springer.
- Ries, E. (2008). *The lean startup*. Retrieved 2017-11-23, from <http://www.startuplessonslearned.com/2008/09/lean-startup.html>
- Robinson, O. C. (2014). Sampling in interview-based qualitative research: A theoretical and practical guide. *Qualitative Research in Psychology*, 11(1), 25–41.
- Rönnbäck, A. Ö., Rosén, T., Olsson, C.-O., & Goldkuhl, G. (2010). Innovative business models enhancing open source software in the public sector. *innovation*, 9, 10.
- Rosen, L. (2004). *Open source licensing: Software freedom and intellectual property law*. Prentice Hall PTR.
- Rosenfall, T. (2012). *Open source vendors' business models* (Unpublished doctoral dissertation). Linköping University Electronic Press.
- Rothaermel, F. T., Hitt, M. A., & Jobe, L. A. (2006). Balancing vertical integration and strategic outsourcing: effects on product portfolio, product success, and firm performance. *Strategic management journal*, 27(11), 1033–1056.
- Rubenstein, D. (2017). *4 successful open source business models to consider*. Retrieved 2018-01-09, from <https://opensource.com/article/17/12/open-source-business-models>
- Sanchez, P., & Ricart, J. E. (2010). Business model innovation and sources of value creation in low-income markets. *European management review*, 7(3), 138–154.
- Schief, M. (2013). *Business models in the software industry: the impact on firm and m&a performance*. Springer Science & Business Media.
- Schief, M., & Buxmann, P. (2012). Business models in the software industry. In *System science (hicss), 2012 45th hawaii international conference on* (pp. 3328–3337).
- SEC. (2013). *Securities and exchange commission: Form 10-k*. Retrieved 2018-04-13, from <http://www.catb.org/~esr/open-source.html>
- Shafer, S. M., Smith, H. J., & Linder, J. C. (2005). The power of business models. *Business horizons*, 48(3), 199–207.
- Simpson, M., Tuck, N., & Bellamy, S. (2004). Small business success factors: the role of education and training. *Education+ Training*, 46(8/9), 481–491.
- Singh, P. V., Tan, Y., & Mookerjee, V. (2008). Network effects: The influence of structural social capital on open source project success.
- Spiller, D., & Wichmann, T. (2002). Floss final report–part 3: Basics of open source software markets and business models. *Berlecon Research GmbH*, 1–58.
- Stallman, R. (2002). *Free software, free society: Selected essays of richard m. stallman*. GNU Press.

- Stewart, K. J., Ammeter, A. P., & Maruping, L. M. (2005). A preliminary analysis of the influences of licensing and organizational sponsorship on success in open source projects. In *System sciences, 2005. hicc's'05. proceedings of the 38th annual hawaii international conference on* (pp. 197c–197c).
- Stewart, K. J., Ammeter, A. P., & Maruping, L. M. (2006). Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects. *Information Systems Research, 17*(2), 126–144.
- Thomas, & Fernández, W. (2008). Success in it projects: A matter of definition? *International journal of project management, 26*(7), 733–742.
- Thomas, & Magilvy, J. K. (2011). Qualitative rigor or research validity in qualitative research. *Journal for specialists in pediatric nursing, 16*(2), 151–155.
- Vilkinas, T., Cartan, G., Saebel, J., et al. (2011). Business success factors.
- Von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: a case study. *Research Policy, 32*(7), 1217–1241.
- Walker, E., & Brown, A. (2004). What success factors are important to small business owners? *International small business journal, 22*(6), 577–594.
- Watson, K., Hogarth-Scott, S., & Wilson, N. (1998). Small business start-ups: success factors and support implications. *International Journal of Entrepreneurial Behavior & Research, 4*(3), 217–238.
- West, J. (2003). How open is open enough?: Melding proprietary and open source platform strategies. *Research policy, 32*(7), 1259–1285.
- West, J., & Gallagher, S. (2006). Challenges of open innovation: the paradox of firm investment in open-source software. *R&d Management, 36*(3), 319–331.
- What is nvivo?* (n.d.).
- Whichmann, T. (2002). Floss final report: Firms' open source activities: motivations and policy implications.
- Wieringa, R. J., & Gordijn, J. (2005). Value-oriented design of service coordination processes: correctness and trust. In *Proceedings of the 2005 acm symposium on applied computing* (pp. 1320–1327).
- Wirtz, B. W. (2011). Business model management: Design—instruments—success factors. *Wiesbaden. Gabler*.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.
- Yin, R. K. (2009). Case study research: Design and methods (applied social research methods). *London and Singapore: Sage*.
- Young, R. (1999). Giving it away: How red hat software stumbled across a new economic model and helped improve an industry. *Journal of Electronic Publishing, 4*(3).
- Zhang, J., & Seidmann, A. (2010). Perpetual versus subscription licensing under quality uncertainty and network externality effects. *Journal of management information systems, 27*(1), 39–68.
- Zott, C., Amit, R., & Massa, L. (2011). The business model: recent developments and future research. *Journal of management, 37*(4), 1019–1042.

# Appendices

## **Appendix A**

# **Interview protocol**

## Interview Protocol

Date:

Interviewer:

Company:

Interviewee:

Job title:

E-mail:

Length in min.:

---

## Questions

1. What year was the company founded?
2. Can you tell me in short the history of the company?
3. Why did you choose to create OSS (and not proprietary software)?
4. Did you create your business model according to research or a specific framework or research?
5. Have you received any funding ? (e.g. angel investors, capital venture..)
6. How do you perceive success, and according to this, is your business successful
7. Are there any tools used to measure the business' succes?
8. What have been the key factors to accomplish success?
9. Did you have any trouble starting your OSS business?
10. What would you advise OSS entrepreneurs?

## **Appendix B**

# **Software Business Model Framework**

<b>Strategy</b>												
<b>Value Proposition</b>	Image	Quality	Functionality		Innovation Leadership		Intimate Customer Relationship		Design & Usability	One Stop Shop	Price	
<b>Investment Horizon</b>	Subsistence Model		Income Model		Growth Model		Speculative Model		Social Model	Cross Finance Model		
<b>Value Chain</b>	Research	Development	Production	Marketing	Implementation	Operations	Maintenance	Support	Education	Replacement		
<b>Degree of Vertical Integration</b>	Low				Medium				High			
<b># of Cooperation Partners</b>	None			One			Few			Many		
<b>Revenue</b>												
<b>Sales Volume</b>	Low				Medium				High			
<b>Revenue Source</b>	Direct				Advertising				Commission			
<b>Pricing Assessment Base</b>	Usage-based				Hybrid Combination				Usage-independent			
<b>Payment Flow Structure</b>	Upfront				Hybrid Combination				Recurring			
<b>Revenue Distribution Model</b>	Low				Medium				High			
<b>Upstream</b>												
<b>Software Stack Layer</b>	Application Software			Systems Software			Hardware Control & Embedded Software			(Web) Content		
<b>Platform</b>	Desktop Computers & Notebooks		Servers		Mobile		Cloud Computing		Embedded Systems	Social Media	Game Consoles	
<b>License Model</b>	Proprietary: Sell Usage Rights			Proprietary: Sell all Rights to Customers			Open Source: Copyleft Licenses (e.g. GPL)			Open Source: Permissive Licenses (e.g. BSD)		
<b>Degree of Standardization</b>	Individual Production				Batch Production				Bulk Production			
<b>Key Cost Driver</b>	Research & Development		Marketing & Sales		Services		Third Party Software Licenses		Hardware	Subcontracting		
<b>Downstream</b>												
<b>Localization</b>	All		Local		EMEA (Europe, Middle East, Africa)		AMERICAS (North-, Central-, and South America)			APJ (Asia, Pacific, Japan)		
<b>Target Customer</b>	Small Organizations			Medium Organizations			Large Organizations			Private Individuals		
<b>Target Industry</b>	All	Consumer	ICT	Manufacturing	Finance & Insurance	Wholesale & Retail	Services (e.g. Health)	Pharma & Chemicals	Const. & Utilities	Transport & Storage	Public Sector	Others
<b>Target User</b>	Business - Broad Workforce		Business - Dedicated Specialists		Business - Managers		Consumers			Software Developers		
<b>Channel</b>	Sales Agents		Events		Telesales		Online Shops			Retail Stores		
<b>Usage</b>												
<b>Implementation Effort</b>	Low				Medium				High			
<b>Operating Model</b>	On Premise				Hybrid Combination				On Demand			
<b>Maintenance Model</b>	Daily		Weekly		Monthly		Quarterly		Biyearly		Yearly	
<b>Support Model</b>	Standard Support				Hybrid Combination				Customer Specific Support			
<b>Replacement Strategy</b>	One Release				Few Releases				Many Releases			

Figure B.1: Software Business Model Framework (Schief, 2013).

## Appendix C

# Software Business Model Framework components



<b>Strategy</b>	
Value proposition	Competitive advantage
Investment Horizon	Business models' strategic time horizon
Value Chain	Summarises the main software value chain activities a firm may cover
Degree of Vertical Integration	Each value chain activity can be either performed internally or externally. The degree of vertical integration measures how many value chain activities are performed within the boundaries of a single firm compared to the overall set of required value chain activities.
# of Cooperation Partners	This component deals with the number of partners that a firm cooperates with.
<b>Revenue</b>	
Sales Volume	Number of sold solutions, usually, this number refers to the number of installations as one customer may have more than one installation.
Revenue Source	Who finally pays for the solution.
Pricing Assessment Base	explores how prices are defined.
Payment Flow Structure	Point in time when customers pay for a solution.
Revenue Distribution Model	The percentage of revenues that is shared with stakeholders.
<b>Upstream</b>	
Software Stack Layer	Classifies software based on a software stack concept.
Platform	The technical platform of a solution.
License Model	The legal regulations associated with the software code.
Degree of standardisation	Analyzes if a firm sells a highly standardized solution or a highly customer-specific one.
Key Cost Driver	This component asks for the dominating cost drivers.
<b>Downstream</b>	
Localization	The geographic areas that a firm addresses.
Target Customer	The customer size .
Target Industry	The target industry that a software firm may address.
Target User	The type of users a solution is designed for.
Channel	The sales channel types used to address customers.
<b>Usage</b>	
Implementation effort	The effort for product installation and configuration. The choice options Low, Medium, and High represent qualitative judgments.
Operating Model	This component analyzes how a software solution is deployed.
Maintenance Model	The release frequency of a software solution. In contrast to patches and hot fixes, a release stands for a major up-date of a product's software version.
Support Model	What kind of support is needed by customers. It mainly depends on the type of support contract.
Replacement Strategy	The number of available product releases at a time

Figure C.1: Explanation of the Software Business Model Framework cited from Schief (2013)

## **Appendix D**

# **Open Source Software Licenses**

License name	Restrictive?	Highly restrictive?	Observations in sample	Observations with activity data
OSI approved licenses				
Apache Software L	N	N	301	121
Apple Public Source L 1.2	Y	N	15	3
Artistic L	N	N	736	223
BSD L	N	N	1,708	618
Common PL	Y	N	34	18
Eiffel Forum L	Y	N	5	3
General PL	Y	Y	18,133	5801
IBM PL 1.0	Y	N	33	7
Intel OSL	N	N	10	6
Jabber OSL	Y	N	20	7
Lesser General PL	Y	N	2,501	1047
MIT L	N	N	395	151
MITRE Collaborative Virtual Workspace L <sup>a</sup>	Y	Y/N	5	1
Motosoto L	Y	N	0	0
Mozilla PL 1.0	Y	N	229	76
Mozilla PL 1.1	Y	N	134	62
Nethack PL	Y	N	16	6
Nokia OSL	Y	N	5	2
Open Group Test Suite L	N	N	1	0
Python (CNRI) L	N	N	162	53
Python Software Foundation L	N	N	0	0
Qt PL	Y	N	136	39
Ricoh Source Code L	Y	N	5	3
Sleepycat L	Y	N	5	2
Sun Industry Standards Source L <sup>b</sup>	N	N	26	9
Sun PL	Y	N	0	0
University of Illinois/NCSA OSL	N	N	1	1
Vovida Software L 1.0	N	N	1	0
W3C L	N	N	0	0
X.Net L	N	N	0	0
Zope PL 2.0	N	N	125	47
zlib/libpng L	N	N	0	0
Other/proprietary	?	?	531	220
Public domain	N	N	820	244

The table summarizes all open source initiative-approved licenses, as well as selected others. The final two columns indicate the number of observations of each license type in the SourceForge database.

*Definitions:* Restrictive: Y implies that the source code from modifications to the program must be made available. Highly restrictive: Y implies that the program cannot be compiled with proprietary programs.

L, license; OS, open Source; PL, public license.

<sup>a</sup>Licensees can choose between two possible options.

<sup>b</sup>Deviations from certain industry standards, however, must be documented.

Figure D.1: Open Source Software Licenses (Lerner & Tirole, 2005)