

Open Source Research Group
Department of Computer Sciences
Faculty of Engineering
Friedrich–Alexander University of Erlangen–Nuremberg



Why and How to Forge Open Source Alliances

Master Thesis

Submitted by: Cecilia La Fuente
Matriculation No.: 21509327
Supervisor: Prof. Dr. Dirk Riehle
Time Frame: 01.06.2012 – 28.01.2013

Abstract

It is well known that Open Source software brought changes to the software development domain. It changed the paradigm from payed developers working in offices to people developing geographically separated, including basements, garages, etc. Even the innovation factor of organizations changed to open innovation, intensively including customers or consumers. This work pretends to further analyze the changes Open Source brought along concerning to alliances. It intends to determine the ways or channels through which Single-Vendor Commercial Open Source firms meet partners in order to set a formal cooperation. To distinguish channels, the researcher first examined a set of interview transcripts. Based on the transcripts analysis, certain channels are identified. To validate the use of those channels, interviews with Open-Source based companies are accomplished. During the interview process, some other channels are identified. Results obtained from this study contribute to the understanding how Open Source modified the approaches companies use to get to know partners.

Contents

Abstract	ii
List of Figures	v
List of Tables	vi
1 Introduction	1
2 Background	6
2.1 Open Source	6
2.1.1 Open Source Inception	6
2.1.2 Open Source Software	7
2.1.3 Open Source Community	11
2.1.4 Single–Vendor Commercial Open Source	15
2.2 Strategic Alliances	18
2.2.1 What an alliance is	19
2.2.2 Rationales for Alliances	23
2.3 Alliances on Open Source Software	26
3 Methodology and Approach	28
4 Results	32
5 Conclusions	38
5.1 Discussion	38

5.2	Research limitations	40
5.3	Recommendations for future research	41
5.4	Summary	42
	Bibliography	43
	A Summary of transcripts review	49
	B Survey	54
	Author's résumé	63
	Eidesstattliche Erklärung	64

List of Figures

1.1	Sales distribution channels	3
2.1	Simple Open Source project classification.	9
2.2	Elements of community-managed governance	14
2.3	Single-Vendor Commercial Open Source revenue sources	16
2.4	General alliance type classification accord duration dimension	20
2.5	Types of technology alliances	21
2.6	Basic phases of an alliance	23
3.1	Research methods employed during elaboration of research work . . .	29
5.1	Means used to meet partners	40

List of Tables

4.1	Distribution of company type.	32
4.2	Percentage of customers and non-paying users from participating firms	33
4.3	Rationale for alliances forging	34
4.4	Descriptive data of partnerships channels	35
4.5	Results categorized according to levels of interactions	36
4.6	Descriptive data of alliances	36

Chapter 1

Introduction

Open Source software is software made available as a form of binary executable files, along with a source code form which is modifiable by its users and redistributed without generating revenues for the original owners [45]. Desphande & Riehle [14] suggested the growing relevance of Open Source software. In 2009 Gartner, the information technology research and advisory firm, predicted the high-speed growing significance of Open Source industry as well [46]. As one consequence of this phenomenon, many studies underlining the Open Source organizational aspects have been realized. Studies such as the one from Capra & Wasserman [7] where the authors present a characterization of different managerial styles required for Open Source projects due to the impact on how organizations and individuals think up, distribute and employ software. In another study, the analysis of Open Source business models from Krishnamurty [29] introduces the specifics of how to build and grow businesses around the Open Source strategy. Another work from Riehle [45] presents the core properties of the business model and the characteristics across the different business functions (i.e., sales, marketing, product management, engineering, support). In his work, he focuses on the so called Single-Vendor Commercial Open Source companies or firms that are the sole owner of an Open Source software product or project they mostly develop and generate revenue from.

The business model of commercial Open Source software or strategies supporting

the revenue generation, at glance, are categorized as:

- Core product
- Whole product
- Operational comfort
- Consulting services

The core product refers to a customer paying for the software mostly due to legal reasons, in order to receive either a certification, indemnification or to be able to embed the software into own products. A customer or a commercial user of the Open Source software pays for the whole product as a consequence of the utility generated from it. A free Open Source product mostly provides restricted features or capabilities. Extending these features requires an upgrade to a non-free version of it and therefore fulfilling requirements of commercial users. Commercial users are typically willing to additionally provide monetary returns in favor of technical support, real-time systems monitoring, non-functional requirements or customizations, bug-fixing subscriptions, encompassing the operational comfort category. Finally, commercial users also pay in favor of training, documentation and implementation or consulting services [43].

Single-Vendor Commercial Open Source firms are strongly supported by a community of users. These firms profit considerable from open innovation, a practical paradigm proposing that companies take advantage of both external and internal ideas for innovation purposes benefiting their product to get faster into the market, increase sales and continually innovate. The open innovation model promotes companies to become active members of an ecosystem, benefiting from each other and enabling them to deliver real business solutions. However, several firms opt to constitute formal and continuous alliances with specific partners [16, 34, 46].

In her work, Jarrat [27] talks about creating an alliance or partnership thereby pooling specific resources and skills in order to achieve a common goal, along with specific goals of the stakeholders. An alliance reflects a joint use of resources and information flows to facilitate alliance partners achieving a desired strategic position. Moreover, forming partnerships is considered as a mechanism of change or an attempt towards influencing their environment through exchange and interfirm relationships. At last, organizations need to form alliances with external entities with the purpose to acquire and/or access assets outside of their own boundaries [5].

In general, there is a bunch of information regarding why and how organizations forge alliances [10, 13, 17, 22, 23, 49]. However, for Single-Vendor Commercial Open Source firms there is no clear evidence by what means or *channels* they find partners since such organizations are built upon a particular business model and are strongly supported by a community of users. Similar to the sales business function, which in order to better reach customers and effectively sell products, differentiate some basic sales distribution channels:

- Wholesales
- Retail
- eCommerce



Figure 1.1: Sales distribution channels

Source: Original illustration adapted from [53, 54]

Wholesale is selling the goods in large quantities to distributors or resellers. Retail sells the products to the final consumers in smaller quantities. eCommerce sales are made through the Internet: a company has a website and their product or services are offered in there [53, 54].

The present research work pursues to analyze another organizational aspect inside the Open Source domain. It examines the strategic partnership forging of Single-Vendor Commercial Open Source firms (SVCOS). Precisely, the hypothesis to validate is if a software company, functioning as a Single-Vendor Commercial Open Source firm, is able to find the right partners, better and/or faster. In order to determine the validity or invalidity of this supposition, the main research inquiry is subdivided into sub research questions. The first sub research question is why a Single-Vendor Commercial Open Source firm forms a partnership. Second sub research question is how a Single-Vendor Commercial Open Source firm gets to know a partner in order to forge an alliance. Third, how effective is the *channel* used to meet the partner for an alliance? Herein, the researcher connotes *channel* as the method or means used to meet a partner, similar as how the term *channel* is used in the sales area to differentiate the approaches to contact customers (i.e., wholesales, eCommerce, etc).

Concretely, the first research sub-objective seeks to determine the rationale behind forging alliances. The second research sub-objective seeks to answer the channels or methods how partners become acquainted toward forming an alliance. Making an analogy with sales channels briefly mentioned before, in Open Source is necessary to differentiate how a Single-Vendor Commercial Open Source firm operates, who its users and customers are, what type of events or activities is involved in, how would be best to get to know new partners in order to increase strategic alliances potential. Furthermore, the last research sub-objective intends to analyze how effective a specific channel is, the relevance of a partner or partners found through a certain channel in the sense if they achieve the expectancies or expected common objectives.

Chapter 2 gives an overview of Open Source foundations and concepts, a brief review of ordinary strategic alliances from non-based Open Source firms. The existing literature related to the topic under investigation in this thesis is also examined. To this chapter, follows Chapter 3 covering a brief description of methods and approaches employed during the elaboration of this thesis. Chapter 4 exhibits the findings suggesting that Single-Vendor Commercial Open Source firms are mainly driven to forge alliances due to potential collaborations, complementary product portfolio among others. Finally, Chapter 5 summarizes the findings and makes conclusions around evidence of channels adopted for making enterprise alliances of Single-Vendor Commercial Open Source firms. This chapter suggests, in addition, future research directions linked to the topic of the present work.

Chapter 2

Background

This chapter presents a review on literature related to Open Source software. First, it presents the Open Source inception, how this software development style was born in a well-known telecommunication company and eventually moved to be developed in basements or garages. Following it, a review on the open source software industry is presented, including its main actors: Open Source community, and Single-Vendor Commercial Open Source firms. Second, this chapter presents an overview on strategic alliance notions from typical organization perspectives (whose business models do not include giving their products for free). Finally, the chapter presents a short overview on existing literature regarding Open Source alliances.

2.1 Open Source

2.1.1 Open Source Inception

The genesis of Open Source began when AT&T, inside its not-for-profit AT&T Bell Laboratories, developed the earliest operating system known as UNIX. AT&T was inquired by the Department of Justice to not generate revenues of it since their main activity was in the field of telecommunications. As a response to the limitation of revenue generation from the new operating system, AT&T began to offer its operating system licenses with a cost of US \$1, mostly to universities and related

academic institutes throughout the world. Among the scientist community, shortly, it became a regular practice to share improvements to the new operating system [32].

At some point, while UNIX was being widely used especially by academics on research projects, the Department of Justice inquiry became invalid and AT&T began to make sales with commercial licenses for their operating system under the Original Equipment Manufacturer (OEM) license type. After this event, UNIX users stop sharing their modifications anymore. As a consequence to the restriction on UNIX, among computer scientists the notion that operating systems are required to be free and their source code should be also made freely available started gaining popularity. Thereby, the idea of “*free software*” originates [32].

Around a decade after, in the 1980s UNIX loses its high popularity and quite a number of people were looking for alternatives to it. Among these people, Linus Torvalds reused the ideas behind Minix, the tiny UNIX-like operating system. He became the most well-known developer due to his first release of Linux in the year 1991 [32, 40]. Similarly to Torvalds, the GNU project¹ developed a UNIX alternative as well. However, GNU intended to completely develop an operating system while Torvalds only a kernel, a merely component of any operating system. The kernel development within the GNU became a nightmare for them and afterwards, Torvalds arrived to relieve them of the struggling situation and jointly develop the GNU/Linux operating system. The GNU/Linux operating system is commonly known as Linux and distributed under the GNU General Public License developed by Richard Stallman who is considered as a pioneer of the GNU Project [32].

2.1.2 Open Source Software

Meeker [32] denominates Open Source to a sort of commercial licensing paradigm as well as a method of software development. Open source and Open Source soft-

¹The GNU project was launched in 1984 to develop the GNU system. “GNU” is an acronym for “GNU’s not UNIX!” [21].

ware are terms, mostly, used interchangeably. In accordance with Perens & Sroka [39], Open Source software or non–proprietary software is an approach of software development which does not only mean obtain access to the source code but the diffusion of Open Source programs must comply with the Open Source Definition (OSD) specified by the Open Source Initiative (OSI). The OSI is a non–profit global organization which aims to indoctrinate and promote the benefits of Open Source, as well as to set up communication paths between Open Source community members. OSI is the guardian of the Open Source Definition. The OSD comprises principles concerning the source code distribution and the licenses for this purpose. The OSD claims that a program should be freely distributed. This distribution must include a compiled version of the software as well as its source code. An Open source license must permit alteration and stem derived programs plus allowing its distribution ensuing the same terms of the original software license. However, given the case, a license may also not allow explicit distribution of software produced from modified source code. But, allow the modified version of the program to be passed on with a different name or a different version number. A license must be technology–neutral, not restricting other software and must not be specific to a product. The rights attributed to a license must apply to everyone to whom a program is redistributed. Additionally, regarding licenses, the OSD states that a license must not restrict anyone from using an Open source program in a specific field of endeavor and to not discriminate any individual or group of persons [38].

According to Riehle [41], in the Open Source software domain a distinction exists between community Open Source and commercial Open Source (see Figure 2.1).

Community Open Source refers to software which is developed and owned by a community of volunteers, usually represented by a legal entity or a foundation (e.g., PostgreSQL, Ubuntu). On the contrary, commercial Open Source software is owned and developed by a for–profit entity (e.g., Oracle is the owner of MySQL). In the first case, the community is in charge to decide the features and contributions the

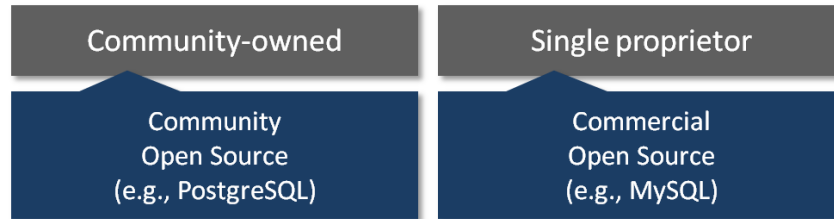


Figure 2.1: Simple Open Source project classification.

Source: Original illustration adapted from [42]

product will carry on in future versions. In the latter case, the firm is the one who decides what is adopted and inserted into the software code base alike what features to implement next.

Corporate users switch to Open Source products due to product performance, low product risk. Since Open Source products are mostly available online, they offer more attractive cost of ownership² such is the case of the US healthcare system that increasingly seeks to reduce its technology infrastructure costs by deploying Open Source solutions [4]. Furthermore, Open Source products are qualified as robust in the sense of security, reliability, availability and system survivability [29].

Wheeler [57] presents in his work quantitative data about the different advantageous characteristics of Open Source industry: a faster growing market share, more reliable, scalable and secure products. Moreover, part of the success from Open Source business is attributed to Open Source service companies which (1) provide first-level support and implementation services and (2) provide second-level support, training and development services [41].

The Open Source software revolution has modified the panorama of system integrators or solutions providers. System integrators offer solutions which may consist of

²Total cost of ownership (TCO) refers to the cost of purchasing, installing and maintaining a software product. TCO is a financial estimate which helps enterprise managers to determine direct and indirect costs of a product or system [25].

hardware, software and services. Therefore their revenues are split up into hardware and software providers. Assuming system integrators use Open Source software, licensing prices sink, hence more profit for them since low prices extend the reach of price-sensitive customer as well as savings on licenses that usually turn into services expenses [41].

Perspectives of software developers as employees are also adjusted as a result of Open Source software. A developer strives to acquire more project specific knowledge rather than firm-specific one so that his\her income increases along with the flexibility or facility to change of employer anytime [41].

Solution providers have a great interest in turning closed-source software markets into software markets with at least one community Open Source product and in consequence, grow their user communities, reduce the cost burden on the company, and build a software ecosystem³ around their products and services. An example of this is the Ximian (nowadays Xamarin) and Microsoft collaboration on the development of project Mono, a standard compatible .NET framework [3, 29, 41].

All that glitters is not gold... This old saying also applies to Open Source. This has some disadvantageous aspects either. Open source firms have an increased risk of getting prosecuted due to patent violations of leaking important intellectual property. Another circumstance that may jeopardize an Open Source firm is that it ends up competing against its own free Open Source project, such is the case of office application suites OpenOffice⁴ and LibreOffice⁵. LibreOffice is forked from OpenOffice and claimed to provide better multi-lingual support among some other advantages. Both applications are available for free but OpenOffice offers a paid premium version for support rather than LibreOffice only relies on the community

³A software ecosystem is a form of platform extension, consisting of a set of software products that together enable, support and automate the business activities and transactions of its users [6].

⁴<http://www.openoffice.org>

⁵<http://www.libreoffice.org>

for it. Additionally, providing a public infrastructure for the community in order to enable communication, provide documentation, wikis, etc., significantly increases firm's expenditures [43].

2.1.3 Open Source Community

The key actors in community Open Source software are individual contributors and not-for-profit organizations [30, 42]. On the first case, those individual contributors participate on the development of Open Source project from different locations, providing freely and constant share of information, assistance and ideas for innovation. To this phenomenon, West and Lakhani [56] denominated a community as a voluntary association of actors, typically lacking of a common organizational affiliation but united by a shared goal in circumstances such as creating, adapting, adopting or disseminating innovations outside a firm's boundaries [56]. The second participant in community of Open Source software is the not-for-profit organization or foundation. Foundations are groups of people or companies that usually started as volunteers on the projects and thereafter envision the economic potential of them. In some cases, customers of Open Source software may also decide to bring efforts together and form a foundation. Foundations manage the Open Source project and provide economical legal support. Some examples of Open Source foundations include the Apache Software Foundation, the Gnome Foundation among some others [44].

The most common motives from individuals to contribute to Open Source software development are the personal ego gratification or peer recognition (e.g., by solving technical problems), altruism and career concern incentives referring to future job offers, shares in commercial Open Source companies. The reasons of foundations to join the Open Source development are mostly economically driven. A first motivation is to reduce development costs by distributing code writing tasks into the participating community members. A second motivation is the ability to increase economical revenues by supplying sales on complementary products. A successful

implementation of an Open Source platform is considered another motivation, as this will lead to compete more effectively across technology stacks and therefore extend their market extent [30, 36].

Within the Open Source community it is possible to differentiate three roles of participants: the user, the contributor and the committer or maintainers as introduced by Riemer [47] and Riehle [41]. This distinction relies on the level of participation and commitment to the Open Source project along with social and technical abilities of the developer. A user is characterized by performing simple downloads, installing and using the software. A contributor is a participant who not just downloads and uses the Open Source program but may develop some improvements and contribute to the project somehow. A committer is a kind of Open Source user who owns more rights over decisions concerning which improvements and contributions to include on a specific program version. He or she is also responsible to perform quality assurance checks of new code to be included on a module. A committer is responsible for one part or one sub-system of an Open Source project [41, 47].

There are various advantages for Open Source software products being supported by a community, the more familiar are:

- The facility to become global products, since communities do not differentiate users across countries [29]. Products involving the community tend to be widely accepted as well as to have more quality [26, 40]. Community members make sales faster and easier since its members are sources of believable testimonials, which might be spread like viruses through social networks, therefore yielding into a cheaper and more effective marketing [40].
- Adoption of “user driven innovation” or the open innovation, meaning that sophisticated users can play in accelerating technological progress due to the fact of source code distribution and therefore driving Open Source products to be highly innovative [30]. The community helps to find ideas for new features [42].

- Product support costs are significantly reduced as a consequence of a self-supporting user community which owns extended product documentation available online any time. Plus, customer support provided by communities, such as Apache and Linux, have won excellence awards [29, 42].

The phenomenon of community supporting firms is not only present in the Open Source software field but also in information goods, fashion, video games, motorcycle equipment and even in food industry with the famous “*my Nutella*”, the web community site of *Nutella*⁶ [11, 56].

A community, in order to reach common goals and accomplish significant outcomes, somehow should coordinate their activities of its multiple members; therefore the idea of Open Source governance is present within communities. Open source governance refers to the means employed to follow a direction, control and coordination of activities of partially or entirely independent individuals and organizations on behalf of Open Source software project to which they contribute [36]. Along the years, Open Source community way of governance experienced different phases of what the shared basis of authority was. At early stages, the leadership of the community is in charge of the project founder and when he or she decides to leave the project, the leadership role is passed on to a trusted member who maintains an active role within the project. In a following phase, the community leadership is an authority democratically selected and this authority’s actions are limited to democratic votes. At a later stage, the authority basis consists of a leader who poses superior skills or his contributions are widely used in the project [37].

Governance models may yield affordances or constraints for contributions of community members, depending if certain principles are present. O’Mahony [36] identifies five common elements of community-managed governance depicted on Figure 2.2. An independent community does not rely on resources from any organization. The

⁶Worldwide renowned hazelnut spread food.

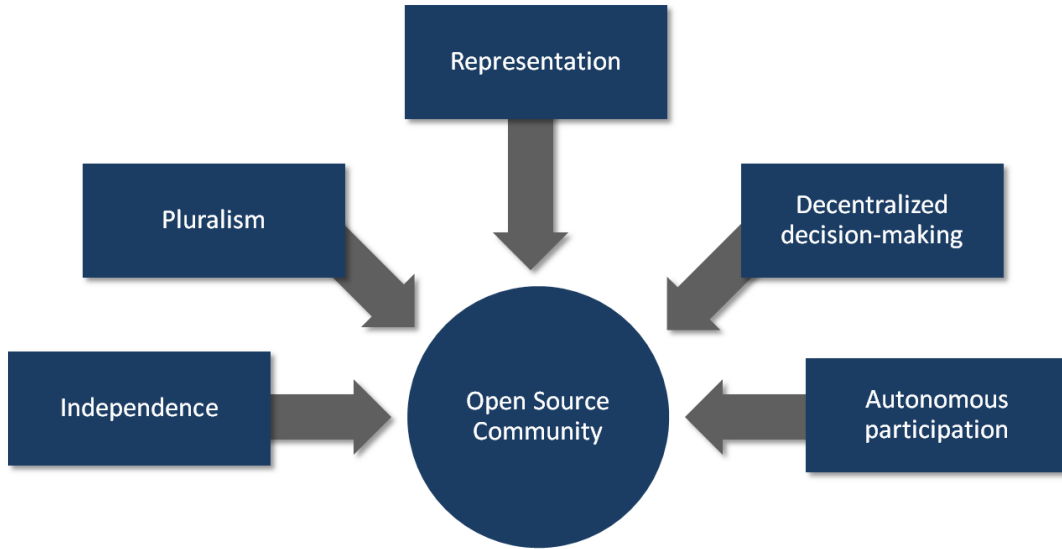


Figure 2.2: Elements of community–managed governance

Source: Original illustration adapted from [36]

independence of a community is determined by the material support or supporting members’ work, the decision–making structure and the independence of employment relations, even though the existence of reward structures from a firm where the community contributor might be employed [36].

Pluralism is a community characteristic which considers several methods, approaches, theories, plans of actions, points of view as reasonable for pursuing a course of action. A community with pluralism avoids to a control group to emerge and foster a multilateral participant base [36]. When representatives are used within a community, their authority is mostly limited to decision making on behalf of the project and not to earn authority over community members or code level decisions. For a better functioning of representatives in a community, it is necessary to delineate the membership base. So that it is possible to differentiate responsibilities and representation of firms, individuals and not–for–profit foundations among the community [36].

A decentralized decision–making in a community means that certain decisions rights are attributed to community members. Community decision rights are distinguished at three levels: the first level concerns to source code decisions, the second level to sub–project decisions and the third to community wide decisions. In order to main-

tain a decentralized decision-making a clear development process is indispensable, so that as many people as possible participate in decisions concerning the Open Source project [36].

Communities consider anonymous participation as a critical factor in order to sustain itself. Contributors to a software project should have the freedom to make contributions under their own terms, as well as increase the socialization opportunities for *newbies* or new members of the community. Hence, the probability of serendipity to happen is higher and the rate of idea generation increases [36].

2.1.4 Single-Vendor Commercial Open Source

Besides the Open Source community as a main actor of Open Source software, there are Single-Vendor commercial Open Source firms. They are ventures which build their business grounded on an Open Source software project or product. Opposite to communities and foundations, these are for-profit organizations with the authority and command of an Open Source software application. Some examples of this kind of software firms are Actuate⁷, Jaspersoft⁸, MySQL⁹, Talend¹⁰.

A Single-Vendor commercial Open Source firm fully owns the control and copyright of an Open Source project and its related intellectual property. It is discussed that to maintain the full control over the Open Source project is considered as crucial for the operations of a Single-Vendor Commercial Open Source firm. In the other hand, a full ownership is not mandatory but acquiring relicensing rights is sufficient.

Single-Vendor Commercial Open Source firms differ from closed-source software vendors for the reason of how the software is built and sold. They provide the software for free and the product in its code form as well. Normally, non-profit users have access to commercial Open Source software at no price. Sometimes the core

⁷<http://www.actuate.com>

⁸<http://www.jaspersoft.com>

⁹<http://www.mysql.com>

¹⁰<http://www.talend.com>

or the whole software product is also offered for free and Open Source companies rely on making money by providing specialized consulting services or proprietary software enhancements (see Figure 2.3). Deliverables conferred to Open Source software customers are binary installation files and software source code [41, 43].

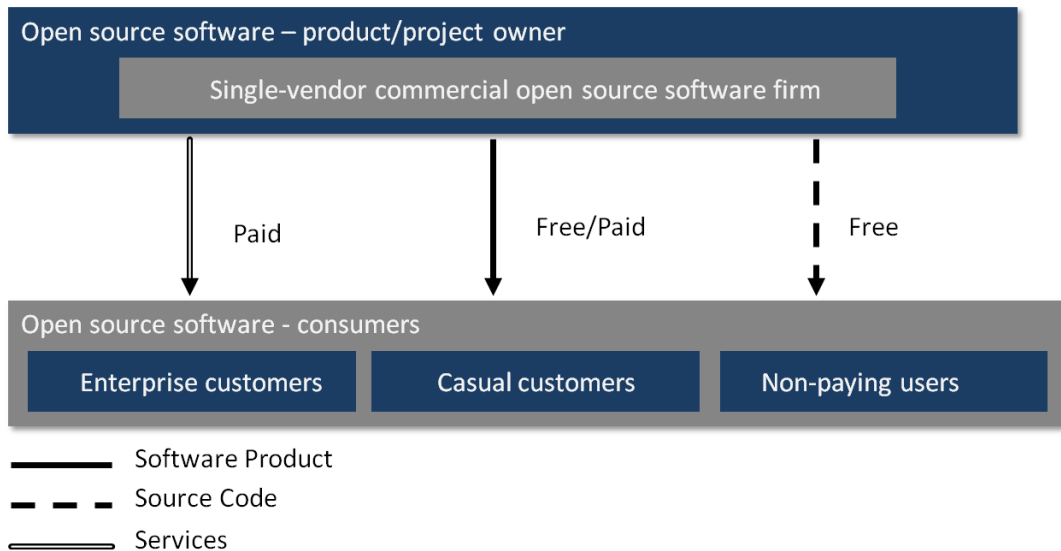


Figure 2.3: Single-Vendor Commercial Open Source revenue sources

Source: Original illustration adapted from [29, 42].

Daffara [12] defines the business model of an organization as the structured model that represents the business and money earning logic of a firm. Literature presents several business models of commercial Open Source software [12, 29, 43]. Krishnamurthy [29] states that for Open Source companies the sale of software alone is insufficient and that potential profit of its products also depend on the type of products they offer and the value they bring for end-users.

Generally, the revenue sources of Single-Vendor commercial Open Source firms are categorized by Riehle [43] as:

- Core product
- Whole product

- Operational comfort
- Consulting services

Krishnamurthy [29] categorizes the revenue sources of Open Source according to different perspectives and licensing model used. The distinct perspectives are of software producers, foundations, communities and third-party service providers. Single-Vendor commercial Open Source firms fall into the software producer category. There exist a bunch of licenses for Open Source; the most well-known is the General Public License (GPL) whose fundamental characteristic is to restrict the distribution terms. Software producers working under a non-GPL licensing model mainly generate revenues by creating a new software product which results from incorporating existing source code into a larger code base. Secondly, by taking an Open Source product and bundling it to their existing products, so to say, by building an ecosystem. In this case, software producers may operate under what is denominated *dual licensing*. This licensing model allows Open Source customers freely use the software, but if they want an extended version of it or do not wish to release further modifications of the software under an OSS license, they must pay for commercial license. For software producers operating under GPL model, the software products derived from existing ones, should be also available as source code form to the end user. Consequently, they expect to generate revenues from enterprise services, support installation, training, etc [15, 55].

Meeker [32] refers Open Source software as a risky approach due to the licensing model it uses. However, he also mentions that is a primarily powerful technological opportunity. GPL is the resulting shortened term of the GNU General Public License and, as earlier mentioned, this is the most widespread license on Open Source. GPL intends to guarantee the freedom of sharing and changing free software. Under the GPL terms, any software company that introduces a product under GPL license must make the source code available, reducing the risk of competing with a proprietary modified version of the same OS product. In other terms, GPL license

reduces the competitors risk but at the same time reduces the potential profit of software firms [20].

Single–Vendor Commercial Open Source firms still inherit the benefits of Open Source communities: a faster product adoption, gratis and quick user feedback, product loyalty and possible code contributions with the appropriate licensing agreements beforehand. In most of the cases, Single–Vendors do not receive external contributions to their program code base or if they agree to do receive them, they previously obtain the copyright of the contributing author. For the contributor, to provide the copyrights means to grant a software company with unlimited use and give the right to relicense the code as well as the authority to represent the project. In return, the contributor receives attribution, reputation and protection by the grantee [29, 44].

Commercial Open Source firms may opt to hire some of their project committers from the Open Source community. The reasoning behind such a determination is a marketing strategy: the visibility of a committer is used with the goal of converting non–paying users into paying customers. Furthermore, hiring a committer yields to a faster and enhanced problem fixing, and a better alignment between company’s strategy and Open Source project [41, 43].

2.2 Strategic Alliances

Despite the fact that the alliances “phenomenon” exists already for more than 25 years, it still has a relevant role among companies’ strategies and their continuous means of adding value to firms [8]. This, either due to the continuous movement toward globalization [17], as the main force which side by side goes with the upraising of technology breakthroughs, or due to the current economy based on ideas rather than material objects: an economy focusing on thoughts, design and organization [8]. In 2001, Dyer *et al.* [18] stated that the top 500 global enterprises possess an average of 60 strategic alliances. Even though this number dates back to about 11

years; the current knowledge-based economy¹¹ favors the strategic alliance forging, providing flexibility, rapid response, customization and deconstruction of the value chain where joint efforts deliver superior value in comparison to single-company competitors [8]. This section introduces what is meant by an alliance, its life cycle and rationales of enterprises to opt for one.

2.2.1 What an alliance is

On the literature, is possible to find several definitions of what an alliance is. Definitions by Das & Teng [13] and Holtbrügge [23] affirm that an alliance is any voluntary inter-firm cooperation agreement, aiming to achieve a common goal of the at least two partners, which is not achievable by only one of them. Alternatively, as Das & Teng [13] mention in their work, forming strategic alliances is about producing the most value out of a firm's existing resources by combining these with other firm's resources resulting in optimal profits.

The nature of alliances may vary according to the cooperation type, herein implying as well how long an alliance may last. Types of alliances may range from short-term to long-term as illustrated in Figure 2.4.

Relational contracts are short-term alliances. They are considered as one time contracts (e.g., training, coaching). Medium-term contractual relationship corresponds to mid-term alliances, for instance, this refers to licensing which implies a greater amount of knowledge in addition to technology transfer and are extended, in average, for five years. The subsequent mid-term alliance type, medium- to long-term supply chain relationship comprises a full-time supplier firm which coordinates the supply chain in addition to taking part in research and development activities. At last, the equity joint venture demands the higher commitment level, larger invest-

¹¹“Knowledge-based economy is an expression made up to describe trends in advanced economies towards greater dependence on knowledge, information and high skill levels, and the increasing need for ready access to all of these by the business and public sectors” [35].

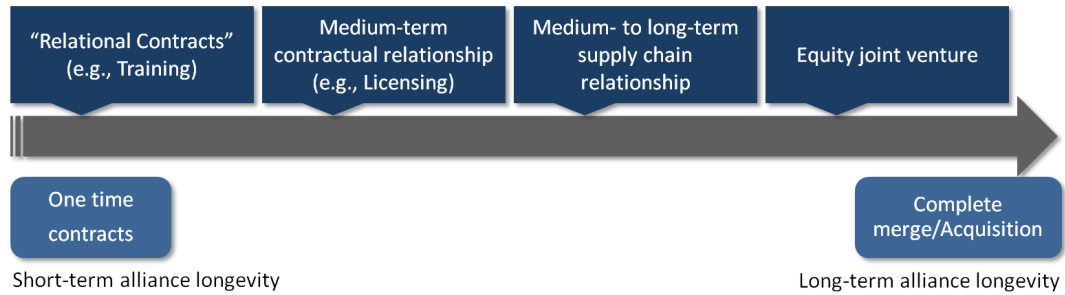


Figure 2.4: General alliance type classification accord duration dimension

Source: Original illustration adapted from [9].

ment and is contemplated to perpetually last [9].

Strategic alliances between organizations remain to increase at an accelerating rate. This fact includes the increasing number of alliances in direction to acquire technology capabilities. Technology alliances are considered as a subset of strategic alliances. An alliance forged in order to acquire technological competences include software licensing agreements, access to technological infrastructure and technological know-how. Knowledge is pondered a crucial resource exchanged, managed and integrated in technology alliances [5, 48].

Technology alliances may vary or be differentiated depending on their scope and their complexity: starting from simpler forms of alliances and moving on to more elaborated settlements (see Figure 2.5).

Some of the technology alliances categories identified by Awazu [5] are equivalent as the category classification identified according to its lasting period. Technology alliances are grouped as:

- Licensing agreements
- Marketing and distribution agreements
- Production and development agreements

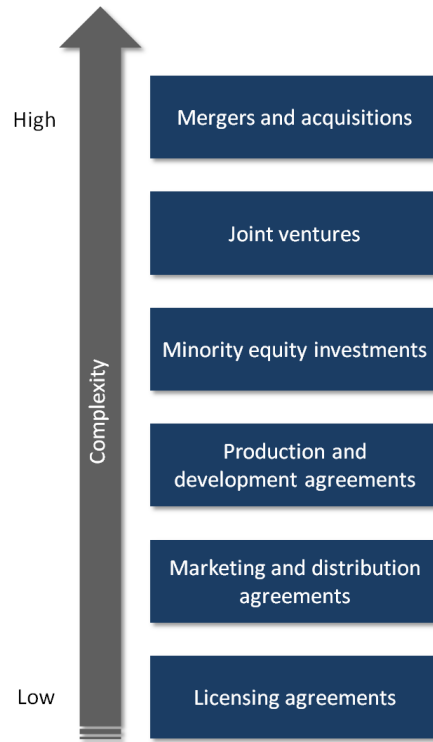


Figure 2.5: Types of technology alliances

Source: Original illustration adapted from [5].

- Minority equity investments
- Joint ventures
- Mergers and acquisitions

Licensing agreements are written contracts under which an intellectual property owner allows a license to use copies of the original property. Licensing agreements are suitable to fulfill simple intellectual property or knowledge access but no control over it needed. Licensing agreements are mainly adopted in commercialization of technology, for instance, Microsoft Office products are available under a proprietary license to use the software but not to modify it. [5]. Marketing and distribution agreements are more involving agreements than licensing complying. Enterprises specialized in different areas are interested in marketing and distribution agreements because they could complement their knowledge. An organization might not directly use other organization's knowledge but enter into an agreement where the

organization uses other's specialized know-how so that its goals are met. An example of this type of technological alliance was in 1998, when Chip Application Technology Ltd. and a division of Sun Microsystems, Inc. engaged in the development, manufacturing, sale and distribution of computer hardware and software produced by Chip [5, 51]. Production and development agreements imply a greater complexity degree than marketing and distribution agreements. They demand more effort in terms of integration, coordination, communication along extended time periods. Members of this type of agreement engage in a cooperation dialogue in order to share their know-how [5]. Minority equity investments are considered the simplest form of equity-based or equally shared values alliances. This type of alliances is conceived to gain access to new and emerging know-how. A minority equity investment has limited access and control over the knowledge source however it is able to monitor its development. A couple of companies, well-known for investing in upcoming and promising startups are Microsoft and Intel [5]. Joint ventures are settlements whereby two or more parties fund resources so as to create a new organism with own corporate entity, structure and resources. This kind of settlements are well situated when a firm intends to enter into a foreign country, a new market, or when pursuing a high risky R&D project. In 2001 Sony, the Japanese consumer electronic company, and the Swedish telecommunication company Ericsson formed a joint venture with the purpose to combine Sony's expertise with Ericsson's technological leadership [5, 50]. Mergers and acquisition is a more complex type of equity-based commitments; an organization combines itself with another or incorporates itself into another. The objective of this kind of alliance is to increase mergers' chances to compete in the marketplace. After around 10 years of joint venture, Sony completely acquired Ericsson by buying the remaining 50% of the joint venture value they held previously [5, 50].

Spekman *et al.*[49], Gonzalez [22] and Dyer *et al.*[18], on their work, identify basic phases an alliance goes through (see Figure 2.6):

- Strategic analysis
- Partner assessment and selection
- Relationship configuration
- Alliance implementation
- Alliance re-evaluation



Figure 2.6: Basic phases of an alliance

Source: Original illustration adapted from [18, 22, 49]

In the first phase, the strategic analysis implies an examination to the business strategy from which the alliance strategy stems. The objective of this phase is to examine and identify industry dynamics and value chain areas where collaboration is reasonable. Based on an alliance strategy definition, different potential partner companies are evaluated. This assessment considers congruence of objectives, success expectations from both parties and their level of companies' culture compatibility. Within the relationship configuration phase, partners establish contractually financial and legal aspects. The alliance implementation stage comprises basis activities that permit the alliance to function. At last, the alliance re-evaluation phase measures the cooperation performance determining if the partnership accomplished its objectives [18, 22, 49].

2.2.2 Rationales for Alliances

Whereas the risk of a partnership failure usually is higher than fifty percent, firms decide to embrace an alliance considering the huge opportunities it could bring.

Opportunities similar to the ones from companies like Coca-Cola and Procter & Gamble obtained from an US-\$4.2 billion joint venture to share distribution system in order to increase reach and reduce time to market. Similarly, the Star Alliance partnership with 27 airline partners granting 1356 destination airports on 193 countries [2, 22].

Contractor *et al.* [8] identify some rationales of enterprises bearing to conform a cooperation. Commonly, these reasons are:

- Risk reduction
- Economies of scale and/or production rationalization
- Technology synergy
- Co-opt or block competition
- Investment barriers
- International market access

Risk reduction for companies usually implies a product portfolio diversification, a lower capital investment, and reduction of fixed costs. For example, a manufacturing company, which wants to develop a new car, spread the production of its parts among the different partners. The huge risk and higher expenses of producing a whole new car by its self would be reduced through sharing risk among all cooperation partners. This was the case of the General Motors\Toyota partnership for their project of creating a small car for the U.S. market. Risk reduction rationale is of special relevance in research-intensive industries (e.g., computers, cars, aircrafts) where rapid and successive creation of new technologies tend to reduce time for production cost amortizations [8].

The second partnership reason, economies of scale and production rationalization, refers to two different concepts, however, at the same time related. Production ra-

tionalization means when the local production of a component is significantly higher when the same component is produced in a different location but with less costs. Therefore, the production of this item is transferred to a lower-cost production location. Added to this fact, there might be the advantage of volume production in the lower-cost production location due to economies of scale. As an example, General Motors has this component's production interchange between its partners Isuzu and Suzuki [8].

Technology synergy enables a technological interaction or exchange between partnerships members which produce a combined effect over the products developed greater than the sum of each. Under this rationale, it is expected that partnership members supply complementary technologies, talents and skills covering aspects of state-of-the-art knowledge required for high technology industries (e.g., pharmaceutical industry, hardware industry) [8].

Co-opt or block competition refers to form an alliance in order to reduce competition or to place pressure on profits and market share with other competitors. For example, Caterpillar Tractor made an alliance with Mitsubishi in order to pressure the shared Japanese market with Komatsu [8].

Investment barriers signify that cooperation is done with the main purpose to obtain a "permit" so as to operate as a "local" entity fulfilling local market requirements. This alliance reason is considered to be the most common. For example, the government of China has the policy to form alliances with the most convenient way for the country to open its market to a foreign company [8].

International market access connotes that a partnership's main objective is to reduce expenses and time while accessing a new international market. Building an organization with global characteristics usually requires considerable costs and plenty of time so as to get market's position [8]. In 1998, Siemens formed a joint venture

with Corning, an US company. The purpose of this cooperation was for Siemens to get the share of the U.S. market and together develop fiber–optic cable technology [19]. Additionally, the reason behind this rationale, mostly in R&D departments, is to acquire country–specific knowledge embedded in the partner [8].

By means of forming alliances, there is also the provision of a platform for organizational learning¹², giving access to the knowledge of the partnership members. The formation of an alliance reduces the risk of knowledge to quickly dissipate. Therefore, it provides an ideal platform to acquire partners’ know–how, which is the critical resource exchanged, managed and integrated, specially, in technological co–operations. Additionally, knowledge management is essential to harvest the benefits of a partnership [5, 8, 48].

2.3 Alliances on Open Source Software

Literature regarding strategic alliances and partnerships is vast and permits to appreciate a certain maturity level concerning the topic. Literature specific to alliance forging among software companies is scarce. As the author presented in the section before, this type of alliances are categorized as technological partnership. However, there is no literature digging into the specifics of technological alliance formation, meaning the phases of alliances presented in Figure 2.6 are determined for alliances in general and not specific for software co–operations.

Open Source software development originated a revolution in the software industry. It drastically changed the software development process compared to the approach of closed–source software. Furthermore, the way Single–Vendor commercial Open Source can approach its users or customers who usually are part of the self–supporting community. Finally, from the software developer perspective, as em–

¹²Inkpen [24] states organizational learning as a function of access to new knowledge and the capabilities of building new knowledge based on previous one.

ployee of a Single–Vendor commercial Open Source company, challenges are diverse as well as continual forms to improve the résumé. Morgan *et al.* [34] state that employees of Open Source companies are a principal source of acquiring collaboration networks, partnerships. This remark is based on the fact that employees of Open Source companies often are involved on projects with research institutes, companies, universities.

Chapter 3

Methodology and Approach

In the present work, the researcher divides the investigation approach in two main phases. In each of the phases, the author uses different research investigation methodologies. For the first phase, along with performing the evident literature review, the author employs qualitative content analysis in order to analyze available material provided by the *Open Source Research Group* of the *University Erlangen–Nuremberg*. The material subject of analysis was transcripts of interviews held with notable people from renowned commercial Open Source companies. For the second phase, in order to validate the data acquired on the first phase, and possibly validate the proposed theory, the researcher intends to employ descriptive statistics so that to describe basic features or a simple summary of the data in this study and therein simple inferences [52]. Therefore, together with the chair’s Professor, the researcher prepares an online survey addressed to responsible persons in charge of strategic alliances on commercial Open Source firms. Supplementary, at this step, the researcher identified the population or to whom the survey is intended for. Hence, a list of companies based on Open Source and their contact information on–line available is collected. A depiction of the process the author carried out is presented below (see Figure 3.1).

The content analysis methodology, employed on the first phase, is used for subjective interpretation of the content of text data through a systematic classifica-

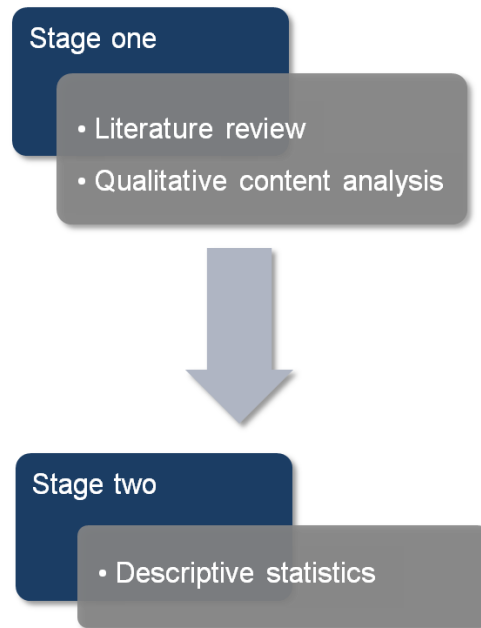


Figure 3.1: Research methods employed during elaboration of research work

Source: Original illustration

tion process of coding labels or patterns [31, 58]. The supporting tool used on this phase was the qualitative data analysis software NVivo 8¹. Following the ideas of King & Horrocks [28], for coding the interviews, the author carried out the following steps:

1. Identify those parts of the transcripts data that are likely to help while addressing the research question of the study. This step included the following tasks:
 - Familiarize with interviews content without making attempts to code it. At this stage, highlight references or make notes regarding the topic under investigation
 - Using references and notes from previous task, define the descriptive codes
 - Perform a second review with the aim to reduce or merge possible overlapped codes together

¹http://www.qsrinternational.com/products_nvivo.aspx

In this step, the coding procedure suggested by Miles & Huberman [33] is utilized.

2. As a second step, go beyond relevant participants' story and rather focus on the researcher's own interpretation of descriptive codes meaning. Thereafter, group descriptive codes that seem they share a common meaning. The descriptive codes resulting from this step were: integration, collaboration, partner, strategy, community, alliances, and product development.

Besides the descriptive codes resulting from previous steps, a summary of how did the interviewed Open Source companies found partnerships is prepared, additionally to some findings from authors Morgan and Feller [34]. This extract is presented in Appendix A: Summary of how and how to forge Open Source alliances. Therein, the name of companies and their products' names are changed to unrealistic names due to privacy constraints.

At stage two, using the material and data gathered on stage one, the online survey is prepared (see Appendix B). The tool employed for the survey's layout and online deployment was LimeSurvey². The survey is organized on four sections each with different type and number of questions. The content of survey's first section is mainly based on the summary presented in Appendix A. This section consists of inquiries concerning how alliances were forged (by which means or channels). The channels listed on this section are additionally sub categorized according to their level of interaction or relationship among the elements interacting (i.e., intercommunication between customer and software enterprise, interactions at enterprise's management level, interactions at enterprise's engineering level). Furthermore, each channel is evaluated or qualified with two additional sub questions that are structured as Likert scale, balanced with five points. As a last question on this section, an open-ended question is included in case any channel was overlooked during the analysis activities.

²<http://www.limesurvey.org>

Section B, in order to make some statistical inference, aims to gather demographic data, hence, this contains multiple choice questions, in most of the cases, and numeric values as input data.

Section C comprises a multiple choice question with regard to the typical rationale behind partnership forging.

Section D, consists of an open ended-question regarding the efforts invested in forging alliances when functioning as an Open Source firm.

Chapter 4

Results

The collected list on phase two resulted on 117 identified Open Source-based companies. Out of the 117, the researcher contacted 71 via email as a first contact approach. 3 firms answered they are smaller companies with no partnerships and also not interested in forming them. 7 companies successfully answered the online survey and out of the 7, 3 of them additionally agreed to an interview via phone. The resulting response rate is 9,86%.

Out of the companies which contributed to this study, 43% are uniquely software vendors and the remaining 57% are software vendors and provide consultancy services as well (see Table 4.1). This data, exhibit that the majority of Open Source based firms rely on other revenue sources as seen on the literature review section 2.1.4.

Type of Company	Percentage
Software vendor	43%
Consulting firm	0%
Software vendor & Consulting firm	57%

Table 4.1: Distribution of company type.

Source: Original illustration

Table 4.2 summarizes the demographic data of participating firms. These data display, as expected, that Open Source-based firms mostly have non-paying users: 71,4% of the companies have more than 100.000 free users. On the contrary, companies with large amount of paying customers is rather low: only 14,3% have between 10.000 and 99.999 paying customers and none of them have more than 100.000 paying customers.

	Number of customers	Number of non-paying users
<100	28,6%	0,0%
100 – 999	28,6%	14,3%
1.000 – 9.999	28,6%	14,3%
10.000 – 99.999	14,3%	0,0%
100.000 +	0,0%	71,4%

Table 4.2: Percentage of customers and non-paying users from participating firms
Source: Original illustration

Table 4.3 presents the reasons of Open Source-based firms in order to forge alliances. Out of eleven identified rationale on the analysis phase, the three considered as more important are access to potential new customers (86%), benefit from collaboration and innovation (71%) and complement product portfolio (57%).

Table 4.4 presents the channels identified on the analysis phase along with their effectiveness evaluations. Despite the effectiveness of the channel was thought to be measured with a balanced five point Likert scale, Table 4.4 omits points “disagree” and “strongly disagree” as they presented no occurrence. The data expose that the channel mostly used is interactions with customers (a) at the level of external interactions. Companies agreed that this channel is effective and 75% of them agreed the partners with whom they forged an alliance were relevant. At the second level of in-

Rationale	Percentage
Access potencial new customers	86%
Benefit from collaboration and innovation	71%
Complement product portfolio	57%
Share expenses with partners	29%
Get free market research	29%
Savings through community contributions	14%
Re-use of engineering efforts	14%
Get relevant technology access	14%
Access to new valuable know-how	14%
Increase savings through shared copyrights	0%
Current hot topics\trends	0%

Table 4.3: Rationale for alliances forging

Source: Original illustration

teractions, recommendations of senior executives (c) and members of the marketing department (e) are means frequently used for partnership conformation. Interviewees agreed 100% on the effectiveness of both channels however, the relevance of the partnership was considered to be 67% for marketing recommendations. At the last interaction level, three channels are more often used: customers who asked for a product feature pointing to a future partner (h), doing an analysis of the community over own company software forge (i), and through events of current software trends (n). Interviewees agreed on channels (h) and (i) to be effective, however, they agreed partners found only through channel (h) are 100% relevant. This fact suggests that finding partners through customers is a relevant and effective channel in order to forge alliances.

Channels	No. answers	Effectiveness of the channel			Relevance of partners found by this channel		
		Strongly agree	Agree	Neutral	Strongly agree	Agree	Neutral
Level: external interactions							
(a) Through customers recommendations	4	0%	100%	0%	25%	75%	0%
(b) Through non-paying users recommendations	1	100%	0%	0%	1%	0%	0%
Level: inner communications							
(c) Through senior company's executive recommendations	3	0%	100%	0%	0%	100%	0%
(d) Through sales person recommendations	1	0%	100%	0%	0%	100%	0%
(e) Through marketing person recommendations	3	0%	100%	0%	0%	67%	33%
(f) Through product manager recommendations	2	50%	50%	0%	50%	50%	0%
(g) Through recommendations of engineering department	1	0%	100%	0%	0%	100%	0%
Level: own's team work							
(h) Through a customer who asked for a product feature pointing to future partner	2	0%	100%	0%	0%	100%	0%
(i) Through analysis of the Open Souce community on the own company's software forge	2	0%	100%	0%	50%	50%	0%
(j) Through analysis of the Open Souce community on a public Open Source forge (e.i., SourceForge.net)	1	0%	100%	0%	0%	100%	0%
(k) Through analysis of existing partners network	1	100%	0%	0%	0%	100%	0%
(l) Through participation in international research collaborations	1	0%	0%	1%	0%	0%	1%
(m) By trade publications	1	0%	100%	0%	0%	100%	0%
(n) Through events of current software trends	2	50%	50%	0%	50%	50%	0%

Table 4.4: Descriptive data of partnerships channels

Source: Original illustration

Table 4.5 exhibits a summary of the channel categorization. This table presents that a combination of certain channels of the three different categories result more effective and let find relevance partners. For instance, 57% of the firms selected channels from the different categories.

Interaction level	Answers
External interactions (enterprise–customer)	14,3%
Inner communications (within enterprise)	14,3%
External interactions & inner communications (enterprise–customer)	14,3%
External interactions, inner communications & own’s team work	57,1%

Table 4.5: Results categorized according to levels of interactions

Source: Original illustration

Table 4.6 presents simple descriptive data regarding characteristics of a partnership itself. There, the reader can appreciate that commercial Open Source companies hold partnerships mainly with companies with the same credo or business model rather than with closed–source enterprises. Additional information suggests that the time period a partnership was hold was not even for one year, the longest for eight and an average and median of 4 years and a half.

	min	max	mean	median
Number of closed–source partners	0	2	1,40	2,00
Number of commercial Open Source partners	0	10	3,20	1,00
Years the longest partnership lasted	1	8	4,50	4,50
Years the longest partnership lasted	0	3	1,17	1,00

Table 4.6: Descriptive data of alliances

Source: Original illustration

Answers to the open–ended question on Section D of the survey, suggest that for the majority of commercial Open Source software companies is simpler to form alliances with companies with whom they share a common background. This fact

makes the setting business process smoother. Since the technology point of view, interviewees also stated that Open Source tools generally tend to connect very well and even better if software products are used in similar areas. However, is not easier to forge an alliance with a closed-source company. Closed-source firms have different niche, stream revenues, and policies, therefore more difficult to set a partnership.

Data obtained during the research work disclose that Single-Vendor Commercial Open Source firms form alliances in order to access to potential new customers, benefit from collaboration and innovation, and complement product portfolio. Data also suggest that finding partners through customers, in addition to senior executives' suggestions, are relevant and effective channels in order to set up alliances.

Chapter 5

Conclusions

5.1 Discussion

Data gathered during the realization of this work revealed that Open Source based companies frequently go after alliances in order to access new customers, benefit from collaboration, innovation and a complement product portfolio. These results slightly differ from what the literature review in Section 2.2.2 presented. According to Contractor *et al.* [8], the common reasons for alliances are risk reduction, economies of scale, block competition, reduction of investment barriers, access international market and technology synergy. Among these reasons, it could be said that Open-Source based companies are interested only in a subset of the rationale for partnerships of typical enterprises. For this type of companies, it seems it is more relevant to seek in a partnership technology synergy. This fact may be ascribed to the firms nature. Technology companies are more interested to gain technological competences rather than accessing international markets or benefiting from economies of scale and production rationalization. Specially, Open Source-based companies might not worry about accessing international markets because since they are founded, they are open to everyone, so anyone throughout the world might know what they offer. Or they might either not worry about economies of scale because they are strongly supported by a committed community and therein

several developers willing to contribute with their work at no cost.

The ways how Single-Vendor commercial Open Source firms meet partners, in the best case, combines different interacting elements as presented in Table 4.6, Figure 5.1. For Open Source-based firms the most important and effective way to meet new partners is through customers. Customers who, asking for a new product feature, indirectly pointed to a potential partner. Or customers who just directly recommended another company as a partner. Within the organization, the recommendation of senior executives and marketing staff is still qualified as relevant and effective, even if the company is Open Source-based. Furthermore, an analysis of what the community is up to, along with events of trends are important ways to meet partners. It is well known that in order to set up alliances companies, at first line, hear what their senior executives have to say in order to select a partner. Senior executives are in charge of performing strategic analysis, partner assessments and finally partner selection. For Single-Vendor commercial Open Source firms this way of selecting partners is still valid. Though, other atypical forms of finding partners come into play. For instance, finding partners as a result of analysis from the company's software forge. This event could be attributed to participating entities on Open Source. The community and customers of Open Source-based companies are highly involved in innovation processes. Hence, customers easily might be included on product roadmaps, letting them suggest how products can be better interconnected to other firm's products, hence propose partners.

Aspects related to partnerships within Single-Vendor Commercial Open Source organizations slightly differ from a common company. Open Source-based firms, just like any other firm, seek in a partnership mainly to get new customers and deriving in more revenues. Other reasons that organizations have for forming alliances, might be considered less or more relevant than others. This may depend on the focus the organization has. For instance, the results of this study revealed that for organizations more technology related, in an alliance, is more considerable to benefit



Figure 5.1: Means used to meet partners

Source: Original illustration.

from collaboration and innovation.

5.2 Research limitations

One of the methodologies employed during the development of this thesis presents some limitations. Qualitative content analysis is greatly dependent on the researcher personal inclinations or biases. Therefore, transcripts interpretation might be subjectively managed.

The findings of this research are restricted to a general denomination of the concept partner. Partners are organizations that voluntarily cooperate. As Section 2.2.1 presents, there are different types of technology alliances and different types of co-operation. Furthermore, Open Source-based firms presented another categorization of partners. Such as, original embedded manufacturer partner, service partners, technology partners, sales partners, hardware partners and several more.

Another limitation of the present work is the sample. The results of this study are limited to the sample size. The response rate of the survey carried out, in order to validate or invalidate the proposed argumentation, is rather low. Consequently, the results might be highly biased. During the interviews, two out of three interviewees repeatedly mention how important the community for them is. Both interviewees mentioned that regardless the type of the partner, the community, most of the time,

was responsible for the successful and effective alliances. This speculation, at first view, seemed to be very well connected with what has been already written about a community supporting a company. Unfortunately, this speculation was not possible to prove due to the lack of data. Therefore, findings of these research need to be re-validated with a significant sample size.

5.3 Recommendations for future research

A future research work comprising how Open Source-based companies meet partners might include a thorough classification of types of partners. This could result in additional channels for making alliances. Moreover, IT consultancy companies could also contribute with different channels as well as other motivations for partnerships.

For further research, it might be also interesting to identify if different type of channels are employed according to the nature of a given alliance. Classification of technology alliances in Section 2.2.1 presented diverse types of technology alliances. These vary on duration and complexity. It could be suggested that, for instance, licensing agreements, the simplest form of a technology alliance, uses the community channel. In the other hand, more complex types of alliances might use the channel of senior executives recommendations. It is also interesting to further consider if there is a combination of channels used in order to set up a partnership. Another interesting research objective might be to determine how long a partnership forged through a certain channel lasts, resembling the classification of alliance types presented on Figure 2.5

Furthermore, for characterization of partnership channels, it could be useful to determine if an alliance resulting from the use of a given channel, necessarily goes through the basic phases of an alliance as presented on Section 2.2.1, Figure 2.6.

5.4 Summary

Open Source software development, since its inception, began to confront imposed paradigms. To mention one example, software development used to be carefully and jealously performed so that renowned software companies can make substantial revenues. On the other hand, Open Source started to do totally the opposite: software development with no secrecy at all. As one consequence of this event, several paradigms faced the change in order to adapt to this school of thought. For instance, managerial styles need to be adjusted, product innovation must to consider customers and therefore be open innovation.

The purpose of this study was to extend the research on the open source phenomenon. Therefore, the research work revisited the rationale for alliance formation in the specific case of Open Source-based firms. Then, it intended to determine the channels or means Single-Vendor Commercial Open Source firms utilize in order to get to know partners and consequently forge formal alliances. The investigation also sought to qualify how effective those partnering channels are.

The results of this research determined that Single-Vendor Commercial Open Source firms seek partners in order to gain more customers, profit from collaboration and innovation, and to complement product portfolio. The channels or ways Open Source-based firms typically meet effective and relevant partners are through interactions with customers, through recommendations from senior executives, and through events of trends.

Bibliography

- [1] <http://www.eweek.com/c/a/Linux-and-Open-Source/UnitedLinux-RIP/>. Accessed: 30/06/2012.
- [2] Member airline – Star Alliance. http://www.staralliance.com/en/about/member_airlines/. Accessed: 26/08/12.
- [3] Ximian and Microsoft [from the Mono FAQ]. <http://www.linuxtoday.com/developer/2001071700620NwGNMS>, 2001. Accessed: 29/12/2012.
- [4] Transforming the future of healthcare with an IT foundation based on Open Source Software. Technical report, Redhat, 2012.
- [5] Y. Awazu. Managing technology alliances: The case for knowledge management. *International Journal of Information Management*, 26(6):484–493, Dec. 2006.
- [6] J. Bosch. From Software Product Lines to Software Ecosystems. In *13th International Software Product Line Conference*, number Splc, pages 1–10, San Francisco, CA, 2009.
- [7] E. Capra and A. I. Wasserman. A Framework for Evaluating Managerial Styles in Open Source Projects. *Open Source Development, Communities and Quality*, 275:1–14, 2008.
- [8] F. Contractor and P. Lorange. The growth of alliances in the knowledge-based economy. *International Business Review*, 11(4):485–502, Aug. 2002.

- [9] F. J. Contractor. *Cooperative Strategies and Alliances (Google eBook)*. 2002.
- [10] F. J. Contractor and P. Lorange. Why Should Firms Cooperate? The Strategy and Economics Basis for Cooperative Ventures. In P. Contractor, Farok J.; Lorange, editor, *Cooperative Strategies in International Business: Joint Ventures and Technology Partnerships Between Firms*, chapter 1, page 513. Emerald Group Publishing, Kidlington, Oxford OX5 1GB, UK, 2002.
- [11] B. Cova and S. Pace. Brand community of convenience products: new forms of customer empowerment the case my Nutella The Community. *European Journal of Marketing*, 40(9/10):1087–1105, 2006.
- [12] C. Daffara. Business models in FLOSS-based companies 1. In *Workshop presentation at the 3rd Conference on Open Source Systems (OSS 2007)*, pages 1–8, 2007.
- [13] T. K. Das and B.-S. Teng. A Resource-Based Theory of Strategic Alliances. *Journal of Management*, 26(1):31–61, Feb. 2000.
- [14] A. Deshpande and D. Riehle. The Total Growth of Open Source. In *Fourth Conference on Open Source Systems*, number December 2006, page 13, Palo Alto, CA, 2008.
- [15] C. DiBona, D. Cooper, and M. Stone. *Open Sources 2.0: The Continuing Evolution(Google eBook)*. 2005.
- [16] M. Dortch. The Open Source Ecosystem Matures : Observations and Recommendations. Technical Report August, Robert Frances Group, Westport, Aug. 2007.
- [17] Y. L. Doz and G. Hamel. The New Alliance Game. In *Alliance Advantage: The Art of Creating Value Through Partnering*, chapter 1, page 316. Harvard Business Press, United States of America, 1998.

- [18] J. H. Dyer, P. Kale, and H. Singh. How To Make Strategic Alliances Work, 2001.
- [19] Forbes.com. Telecommunications/Transportation & Logistics. <http://www.forbes.com/best/2001/0521/106print.html>, 2001. Accessed: 26/08/2012.
- [20] Free Software Foundation, Inc. Frequently Asked Questions about the GNU Licenses. <http://www.gnu.org/licenses/gpl-faq.html>WhatDoesGPLStandFor, 2011. Accessed: 18/08/2012.
- [21] Gnu.org. gnu.org. <http://www.gnu.org/home.en.html>. Accessed: 19/06/12.
- [22] M. Gonzalez. Strategic Alliances: the Right way to compete in the 21st Century. *IVEY Business Journal*, (September/October):47–51, 2001.
- [23] D. Holtbrügge. Management Internationaler strategischer Allianzen. In J. Zentes, B. Swoboda, and D. Morschett, editors, *Kooperationen, Allianzen und Netzwerke Grundlagen-Ansätze-Perspektiven*, chapter 5: Gestalt, pages 1181–1201. Gabler, 2008.
- [24] A. C. Inkpen. Learning, Knowledge Management, and Strategic Alliances: So Many Studies, so Many Unanswered Questions. In F. J. Contractor and P. Lorange, editors, *Cooperative Strategies and Alliances*, chapter 12, page 925. Emerald Group Publishing, 2002.
- [25] Investopedia. total cost of ownership. <http://www.investopedia.com/terms/t/totalcostofownership.asp#axzz2GvfMFXGW>. Accessed: 12/12/2012.
- [26] B. Ives and M. H. Olson. User Involvement and MIS Success : A Review of Research. *Management Science*, 30(5):586–603, 1984.

- [27] D. G. Jarratt. A strategic classification of business alliances: a qualitative perspective built from a study of small and medium-sized enterprises. *Qualitative Market Research: An International Journal*, 1(1):39–49, 1998.
- [28] N. King and C. Horrocks. *Interviews in Qualitative Research (Google eBook)*. SAGE Publications Ltd, London, UK, 2010.
- [29] S. Krishnamurthy. An Analysis of Open Source Business Models. In *Making Sense of the Bazaar: Perspectives on Free and Open Source Software*, pages 279–296. Bothel, WA, February 2003.
- [30] J. Lerner and J. Tirole. Some Simple Economics of Open Source. *The Journal of Industrial Economics*, 50(2):197–234, Mar. 2003.
- [31] P. Mayring. Qualitative Content Analysis. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, 1(2), 2002.
- [32] H. J. Meeker. *The Open Source Alternative: Understanding Risks and Leveraging Opportunities(Google eBook)*. John Wiley & Sons, 2008.
- [33] M. B. Miles and A. M. Huberman. *Qualitative Data Analysis: A Sourcebook of New Methods*. Sage Publications, 1984.
- [34] L. Morgan, J. Feller, and P. Finnegan. Open Source Innovation Networks : Exploring High and Low-density Models. page 13, 2012.
- [35] OECD. Knowledge-based economy Definition. Glossary of Statistical Terms. <http://stats.oecd.org/glossary/detail.asp?ID=6864>, 2005. Access: 20/08/12.
- [36] S. O’Mahony. The governance of open source initiatives: what does it mean to be community managed? *Journal of Management & Governance*, 11(2):139–150, June 2007.
- [37] S. O’Mahony and F. Ferraro. The Emergence of Governance. April 2007.

- [38] Open Source Initiative. The Open Source Definition. <http://opensource.org/docs/osd>, 2012. Access: 08/06/2012.
- [39] B. Perens and M. Sroka. The Open Source Definition. Technical report, 2008.
- [40] E. Raymond. The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3):23–49, Sept. 1999.
- [41] D. Riehle. The Economic Motivation of Open Source Software: Stakeholder Perspectives. *IEEE*, (April):25–32, 2007.
- [42] D. Riehle. The Business Model of Commercial Open Source Software. Technical report, SAP Research, SAP Labs LLC, 2009.
- [43] D. Riehle. The Single-Vendor Commercial Open Source Business Model. Technical report, Friedrich-Alexander-University of Erlangen-Nürnberg, Erlangen, Germany, 2009.
- [44] D. Riehle. The Economic Case for Open Source Foundations. *IEEE Computer*, 43(1):86 – 90, 2010.
- [45] D. Riehle. The Single-Vendor Commercial Open Source business model. *Information Systems and e-Business Management*, 10(1):5–17, Nov. 2010.
- [46] D. Riehle. Controlling and Steering Open Source Projects. *IEEE Computer*, 44(7):93–96, 2011.
- [47] P. Riemer. An Analysis of Work Rhythms in Open Source, Thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2012.
- [48] F. T. Rothaermel and D. L. Deeds. Alliance type, alliance experience and alliance management capability in high-technology ventures. *Journal of Business Venturing*, 21(4):429–460, July 2006.
- [49] R. E. Spekman, T. M. I. Forbes, L. A. Isabella, and T. C. MacAvoy. Alliance management: a view from the past and a look to the future. *Journal of Management Studies*, 35(6):747–772, 1998.

- [50] K. Suleman. Sony completes takeover of Sony Ericsson joint venture. <http://www.v3.co.uk/v3-uk/news/2152944/sony-completes-takeover-sony-ericsson-joint-venture>, 2012. Accessed: 12/12/2012.
- [51] Sun Microsystems. Sun Microsystems:Computer Company Joint Marketing Agreement. <http://contracts.onecle.com/cativity/sun.mkt.1998.08.01.shtml>, 1998. Accessed: 12/12/2012.
- [52] W. M. Torchim. What is the Research Methods Knowledge Base ?, 2002.
- [53] D. Traxler. Choosing Sales Channels to Reach Target Consumers. <http://www.practicalecommerce.com/articles/3526-Choosing-Sales-Channels-to-Reach-Target-Consumers>, 2012. Accessed: 21/12/2012.
- [54] Virtual Sales. Direct sales vs Channel sales – What’s the difference? <http://www.virtual-sales.com/Direct-sales-vs-Channel-sales.html>, 2009. Accessed: 20/12/2012.
- [55] R. T. Watson, M.-C. Boudreau, P. T. York, M. E. Greiner, and D. Wynn. The business of open source. *Communications of the ACM*, 51(4):41–46, Apr. 2008.
- [56] J. West and K. R. Lakhani. Getting Clear About Communities in Open Innovation. *Industry & Innovation*, 15(2):223–231, Apr. 2008.
- [57] D. A. Wheeler. Why Open Source Software/Free Software (OSS/FS)? Look at the Numbers ! 2004.
- [58] Y. Zhang and B. M. Wildemuth. Qualitative Analysis of Content. pages 1–12, 2005.

Appendix A

Summary of transcripts review

How?

- **Exploring the market before any time or engineering investment**

- JalapeñosSoft continuously monitors what is occurring in their Sourceforge as well as in other forges so that they find “winning” chances like the one from Mandarinaleaf with their wizard (module developed by a customer and afterwards, included as part of JalapeñosSoft core product). As a second example, JalapeñosSoft finds the German project SweetReport which has an optimized version of LocotosReports with Sweet Sale Rep. Concluding, JalapeñosSoft observes and tests these modules/connectors first and if they are successful, an inclusion into the core product of the new modules/connectors is highly probable therefore a partnership with project’s owner/representatives is seen in the upcoming future (Interview_1, ln.290)

- **Customers contact**

- SICMom, JalapeñosSoft’s customer who uses their software, contributes back some modules after establishing contact with them (Interview_1, ln.119)
- Partnerships through customers (Interview_1) On the case of Mandarinaleaf with JalapeñosSoft for wReports wizard (Interview_3, ln.1066)

- **Top–down driven decision**

- JalapeñosSoft finds out the necessity to include ETL capabilities on their product portfoliosuite. ETL is considered to be quite a unique domain, therefore they decide not to write code for it but go for an strategic alliance with Talentosos, with whom JalapeñosSoft holds strong relationships (Interview_1, ln.751)

- **Community contact**

- Community provides supporting data for product roadmap and distribution strategy roadmap (Interview_2, ln.1345)

- **Sales**

- Sales department has the ability to open up partnerships, distribution channels, system integrators (Interview_1, ln.1391)
- OEM customers care more about how well and easily can the code be customized, how well the code would fit in, how is the code organized so to easily make use of it. It's more tech partner–ish embeddable sale (Interview_3, ln.636)
- Among a high–density collaboration network¹ of open source firms, firms are more responsive to external ideas and knowledge from various networks partners. This openness manifests recurring exchange relationships and agreements forming later strategic alliances or joint ventures [34]
- As a result of employee representation of open source software companies in collaborative international research funded open source software projects which involve research institutes, companies, universities [34]

¹High–density networks of strong ties represent those firms and partnersthat are closely related [34].

Why?

- **Avoid duplication of engineering efforts**

- A customer of JalapeñosSoft decides to contribute their further development of the LocotosReport instead of waiting for them to develop what they need. Many opportunities like these are open from the fact of giving the customer access to the source code (Interview_1, ln.77). Similar case, another customer contacts JalapeñosSoft for contributing modules of the product which are not planned on their official product road-map for future development. The customer contributes with a backward-looking module but it's included in the core product anyway (Interview_1, ln.119)
- Example given by John Doe regarding LocotosReports and wReport (JalapeñosSoft's reporting and analytical tools) which are built upon open standards and available through permissive licenses, yielding to the community not to extend energy and time making their applications again (Interview_3, ln.185)
- As literally said by Dan Kusnetzky, International Data Corp. vice president for system software, says about United Linux alliance [1]

- **Form an effective competitor for Red Hat Inc.** Especially, in the sense of making Linux more attractive to enterprises and independent software vendors [1, 29]

- **Save money or share expenses**

- Literally said by Dan Kusnetzky while making declarations regarding United Linux Topic [1]
- Saving money on licensees, legal aspects (Interview_1, ln.25)
- About relicensing (Interview_1, ln.158:). Getting the community contributions which afterwards would be valuable for the company especially for revenue generation (Interview_1, ln.213). Committers of the 5 core JalapeñosSoft projects

must have signed an agreement with the company. The agreement grants the code to JalapeñosSoft or agreed on a shared copyright (Interview_1, 236)

- Example of giving product's translation to the community, getting a fair amount of contributions from community members and spending US \$2 instead of US \$20.000 (Interview_1, ln.344)
- Example of Prismtech², one unfamiliar company developed an advanced UML modeling tool for one Prismtech's product without any financial investment from Prismtech side [34]

- **(Millions of) opportunities from the fact of giving the source code to customer**

- SICMom, JalapeñosSoft's customer, contacts JalapeñosSoft for contributing modules of the product which are not in their product road-map planned for future development. The customer contributes with a backward-looking module but it's included in the core product anyway (Interview_1, ln.119)

- **Free market research**

- In the case of Mandarinaleaf (the company which developed a wizard for wReport use), JalapeñosSoft analyzes the adaption and popularity of it, especially among the SourceForge. If it is widely popular, they might integrate it into the core projects (Interview_1, ln.265; Interview_2, ln.1107)
- On JalapeñosSoft's case, for their product LocotosReport, JalapeñosSoft finds a German specialized version for AzucarCRM (SweetReport). This project is found due to a continuous exploration of what is occurring on SugarForge and on SourceForge: a free market research using world's largest open source software development site³ (Interview_1, ln.275; ln.290)

²<http://www.prismtech.com>

- Asked by JalapeñosSoft, Asiasoft (JalapeñosSoft’s distribution partner) analyzes the increasing volume of community activity on the Forge. Sub sequentially, thanks to this information, JalapeñosSoft is easily able to define product roadmap and distribution strategy roadmap (Interview_2, ln.1332)
- JalapeñosSoft, as an open source vendor, does not need to hire focus groups and do expensive market research in order to determine which widgets might be integrated on the core products. The data required for this analysis is available on the Forge (Interview_2, ln.1406)
- **In order to benefit from open collaboration and innovation** [34]
 - Level of commitment, potential to work with others towards a common goal/product
 - that helps to complete their product/service portfolio Knowledge exchange, ability to work and exchange useful information, innovation, capabilities, ideas
 - Alignment of objectives and governance
 - Access to technologies, standards, further development of technologies
 - More effective access to potential customers all over the world due to open innovation model
- Provide capabilities, experience, resources and services that compliment firms’ product or services [34]

Appendix B

Survey

This survey investigates how commercial open source companies find strategic business partners. Filling out the survey should take less than 10 min of your time.

Thank you for your help!

Section A

We have a list concerning channels or ways of learning about a future partner we would like to ask you about. In particular:

How did you get to know about a future strategic partner?

Please choose all options that apply:

Through external interactions like:

- **A customer who recommended the future partner**

Evaluation of this channel:

This channel is effective

- Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

- Strongly disagree Disagree Neutral Agree Strongly agree

- **A non-paying user who recommend the future partner**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

- **A non-paying user who recommend the future partner**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

Through inner communications like:

- **A senior companys executive recommending the future partner**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

- **A sales person recommending the future partner**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

- **A marketing person recommending the future partner**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

- **A product manager recommending the future partner**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

- **Someone from engineering department recommending the future partner**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

Through your or your team's own work:

- **A customer who asked for a product feature that pointed to the future partner (e.g., software integration solutions)**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

- **By analyzing the open source community on the own company's software forge**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

- **By analyzing the open source community on a public open source forge like SourceForge.net**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

- **By analyzing the existing partners network**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

- **Participating in international research collaborations**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

- **By trade publications (e.g., magazines, blogs, etc.)**

Evaluation of this channel:

This channel is effective

Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

Strongly disagree Disagree Neutral Agree Strongly agree

Any other channel we overlooked?

Evaluation of this channel:

This channel is effective

- Strongly disagree Disagree Neutral Agree Strongly agree

Partners found through this channel are relevant

- Strongly disagree Disagree Neutral Agree Strongly agree

Section B

1. What type of company are you?

Choose only one of the following:

- Software vendor
- Consulting firm
- Both

2. How many employees does your company have?

Choose only one of the following:

- 1 – 9
- 10 – 49
- 50 – 249
- 500 – 999
- 1000 – 2499
- 2500 – 4999
- 5000 – 9999
- 10000+

3. How many customers does your company hold?

Choose only one of the following:

- <100
- 100 – 999
- 1000 – 9999
- 10000 – 99999
- 100000+

4. How many non-paying users does your company hold?

Choose only one of the following:

- <100
- 100 – 999
- 1000 – 9999
- 10000 – 99999
- 100000+

5. Approximately, how many closed-source software partners does your company have?

.....

6. Approximately, how many commercial open source software partners does your company have?

.....

7. Approximately, how many years does/did the longest partnership last?

.....

8. Approximately, how many years does/did the shortest partnership last?

.....

Section C

What is/are the reason(s) for striking strategic partnerships?

Check any that apply:

- o To benefit from collaboration and innovation
- o To get free market research
- o To access to (new) valuable know – how
- o To complement product portfolio
- o To access potential (new) customers
- o To re – use engineering efforts
- o To increase savings through shared copyrights
- o To saving through community contributions
- o To share expenses with partners
- o To get relevant technology access
- o Current hot topics/trends (e.g., Cloud Computing, SaaS, etc.)

Section D

In your opinion, do you think it is easier to form alliances between Single-Vendor open source companies?

Please, briefly explain your answer.

.....
.....
.....
.....

Thanks!

Author's résumé

Cecilia La Fuente holds a Bachelor of Science (B.Sc.) degree in Systems Engineering from the Universidad Católica Boliviana San Pablo, Cochabamba–Bolivia. After some years of working experience, on the Software Engineering domain, decides to pursue postgraduate education. Since 2010, studies at the University Friedrich–Alexander–University Erlangen–Nuremberg for a Master's degree (M.Sc.) on International Information Systems.

Eidesstattliche Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Nürnberg, den 28. Januar 2013

Cecilia La Fuente