Drivers and Barriers to the Application of Free and Open Software in SMEs fostering Security, Safety and Digital Sovereignty

Master's Thesis

Maria Susana Soriano Ramírez

Thesis Advisor: Dr. Peter Heinrich, ZHAW Winterthur
Thesis Co-Advisor: Dr. Christian Weber, ZHAW Winterthur

Master's Thesis submitted to the School of Management and Law, Zurich University of Applied Sciences (ZHAW), for the degree of Master of Science in Business Information Technology.

Winterthur, 26th May 2022

Management Summary

In principle, many arguments make Free and Open-Source Software (FOSS) an attractive proposition for all companies. However, in practice there are human, organizational, environmental, and technological barriers that prevent its adoption. Much of society's critical infrastructure depends on the robustness of FOSS. The adoption and innovation of this technology is even embedded in the business model of some companies to gain as many benefits from the technology as possible.

This thesis has the main objective of answering the following research question:

> *What are the barriers and drivers that a Swiss SME can have for adopting FOSS?*

It focuses on the development of the FOSSA (Free and Open-Source Software Adoption) reference model, that permits systematic analysis of this phenomenon in both individual and organizational contexts. FOSSA encompasses the behavior of individuals that belong to an SME towards the adoption of open software, as well as business related factors that influence the performance of a company in the marketplace. The FOSSA model offers a structured approach to evaluate individual and organizational reasons and concerns for OSS adoption.

Ultimately, decision makers must reach the best conclusions for their organizations and create digital strategies, so that their security, safety, and digital sovereignty.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Peter Heinrich, for his professional guidance, and his time throughout the entire process of this Master's Thesis.

Many thanks also go to Dr. Mike Krey, for his quick answers to all my questions related to the administrative master's thesis process as well as for his support during the years of the master's degree program.

Furthermore, I would like to thank my co-supervisor Dr. Christian Weber for his time and support. To all the anonymous respondents to the interviews that shared their valuable experience that helped enrich this thesis work, please accept my sincere gratitude as well because this thesis work would not be possible without all your treasured contributions.

Special thanks to my husband for his love and patience, to my brother and two sisters for their unconditional love, and last but not least to my two daughters for their immense support, love and for brightening my days since they were born.

Table of Contents

List of Figures

List of Tables

1 Introduction

Free and Open-source software (FOSS) is being used across the IT industry, particularly in larger enterprises such as GitHub, Salesforce, Google, Dropbox, and Microsoft Azure (The Linux Foundation 2021). Most companies recognize that open-source in an enterprise can lead to successful digital transformation, act as an innovation driver, provide security benefits, and that vendor participation in open-source communities is important to them (Haff 2021). This is because there are a number of behaviors in the open-source community that drive continuous improvement and foster innovation (OECD Publishing 2019; Petrinja et al. 2013).

Traditionally, IT systems such as Enterprise Resource Planning (ERP), Supply Chain Management (SCM), Customer Relationship Management (CRM), etc. have been implemented and deployed on the company's own hardware. Nowadays, these IT systems are extended via cloud computing environments, some of which are based on FOSS ecosystems. Even private cloud providers such as AWS, Microsoft Azure and Google Cloud support open-source communities such as Node.js and the OpenJS Foundation (Balla et al. 2020).

It is important for any enterprise to see a return on its IT investment through efficiency and effectiveness (Peppard and Ward 2004). To design and develop information systems (IS) that fit an organization is the main tasks of many (IS) practitioners and business managers. This design aids companies in purposeful organizing resources to accomplish their goals (Hevner et al. 2004). Furthermore, how organizations can achieve their strategic information systems (SIS) goals while seeking opportunities for competitive advantage through IT resources is a critical issue and the top main concern of business executives (Galliers et al. 2020) .

On the other hand, the phenomenon of open-source software has been researched for nearly two decades, and yet it is the least understood of the revolutionary inventions of our time (Fitzgerald et al. 2019). Looking at it from an economic perspective, it seems quite incredible that people participate and contribute to create, change, or fix code

without receiving any financial reward (Wen et al. 2019). However, much of society's critical infrastructure today depends on the robustness of open-source software. A current example of this dependency can be seen in the recently discovered log4j vulnerability (Nagle et al. 2021; MITRE 2021), which has forced an industry wide effort to upgrade or fix vulnerable components.

It can be argued that this is one of the reasons why profitable companies fund and even participate in the development of open-source (Detwiler 2021). The adoption and innovation of this technology is even embedded in the business model of some companies to gain as many benefits from the technology as possible (Karels 2003).

FOSS offers companies the opportunity to adapt the software to their needs, to avoid vendor lock-ins and the ability to scale without licensing issues. Despite all these advantages, the adoption of pure open-source software in the infrastructure of Small and Medium Enterprises (SMEs) is very rare.

According to The Linux Foundation 2018 survey (OpenExpo Europe 2021), the majority of the participants whose SME companies are planning to adopt a FOSS ecosystem commented that the major barriers that hinder the adoption of a FOSS system are:

- the lack of knowledge of implementation of open-source policies,
- not creating a powerful enough guiding coalition (Kotter, John, P. 1995),
- the scarcity of legal experts in FOSS,
- the difficulty to find the required resources to be in accordance with the licensing requirements.

The same survey indicates that one of the reasons that participants have not implemented an open-source program in their companies refers to the lack of information about the possibility of using an open-source system. A significant percentage of the participants that work in SMEs stated the lack of time as one of the two reasons for not having plans to implement an open-source program. The second

reason is that they cannot identify the business value due to the small size of the company.

Adoption on a FOSS ecosystem in Switzerland is not common but according to the SME Swiss Government Portal for small and medium enterprises “the number of companies expressing an interest in free operating systems, like Linux, is steadily rising” (SME Portal for small and medium-sized enterprises 2021). This statement can also be found in the Open-Source Study Schweiz 2021, where it is stated that the importance of open-source software in the last three years have grown significatively in Swiss organizations. Indeed, users of open-source in Switzerland recognize the importance of reduced dependency and the resulting of better negotiating opportunities when purchasing software (Open Source Studie Schweiz 2021).

In everyday life, each Swiss SME must comply to very specific Swiss laws and regulations. A good example is the Swiss payroll administrative process or *einheitliches Lohnmeldeverfahren (EML): swissdec* (SME Portal for small and medium-sized enterprises and KMU 2021). The *EML:swissdec* association is sponsored by the Swiss government to help Swiss SMEs to align with the payroll administrative processes, and it is composed of the major organizations that need to process payroll data in Switzerland. This association certifies accounting software manufacturers according to their compliance to the *EML:swissdec* regulations.

The terms of use from most of the *EML:swissdec* certified software manufacturers do not provide access to the application source code, and forbid their users from decompiling it. However, it is possible to use open-source software for this specific task. For example, the Swiss certified *odoo* accounting program opens the possibility for an SME to start working on an open-source software as a base for production deployment. However, comparing the community and enterprise commercial edition of the software in the finance and accounting section, the commercial edition makes the major tasks already available for the company including the certification from the *EML:swissdec*. On the other hand in the community edition, the SME needs to start

from zero, building every accounting task plus applying for and passing the Swiss payroll administrative process certification (Odoo Community 2021).

Therefore, if an SME wants to adopt a pure FOSS ecosystem, one barrier that it encounters is complying with the Swiss regulations of the payroll administrative process and having to invest time working on building different features of the payroll administrative process on the open-source software. Also, there is no open source community that can contribute in building open source software certified by *EML:swissdec*; thus the SME might opt for the benefit of the readiness of the commercial software.

Once the SME chooses a commercial software, it is distributed in binary form which simplifies installation (Karels 2003), but also creates a situation where these organizations are essentially forced to use the software and comply with the manufacturer because the consequences of changing to another software supplier might result in interruptions to business operations or financial loses. This means that the SME experiences "lock-in", i.e., is not allowed to end or change its arrangement with the vendor; thus, losing its part of its freedom.

The idea of free software, according to Stallman, from a philosophical point of view, is that "it respects the users' essential freedoms: the freedom to run it, to study and change it, and to redistribute copies with or without changes" (Stallman 2009, p. 31). In the same article, Stallman states the difference between Free software and Open-Source software. Free software promotes social solidarity in the sense of sharing and cooperation. Open-Source software allows users to change and redistribute the software to make it more powerful and reliable. These two terms appear together in the acronym FOSS.

The ability for an individual in a company to act and make decisions in a self-determined manner in the digital space is what free and open-source software aims for. This last argument is the definition of "Digital Sovereignty" and is attracting attention in Switzerland. For instance, the Open-Source Swiss Study (2021) concluded that the reason that digital sovereignty has increased its popularity in this country is because

large tech companies have progressively made the most of their vendor lock-in advantage and as a result they have irritated costumers.

However, irritated costumers are not the only ones turning their heads to an open-source solution, but also worried business managers. In an interview with Chris Fearon, head of Black Duck Software's in, he states "if organizations don't take Open Source Software for Security seriously, they could be making the task of an attacker easier" (Mansfield-Devine 2017, p.7). Certainly, the beneficial effect of open source on both transparency of the source code and security makes FOSS attractive.

Hoepman et al. (2007) argued that open-source software for security is a requisite to assemble systems thar are more secure. Transparency is also an important advantage because the user can corroborate that no backdoor functions or security risks exist. This might avert unauthorized access to sensitive information (Hon et al. 2010).

Nowadays, the mechanisms behind FOSS projects and how they are controlled are better understood. This has been achieved by large companies involving themselves more into the FOSS projects and participating in the FOSS community, so they know who the developers are and which projects they can trust. By participating so actively, they can reduce the risk associated with using that system (Lara Torralbo et al. 2021; Hoepman and Jacobs 2007).

1.1 Research Question

All the above arguments make FOSS an attractive proposition for all companies in principle; however, in practice there are also organizational and technological barriers that prevent its adoption. Much of society's critical infrastructure depends on the robustness of FOSS. The adoption and innovation of this technology is even embedded in the business model of some companies to gain as many benefits from the technology as possible. Therefore, this thesis has a main objective to answer the research question:

> *What are the barriers and drivers that a Swiss SME can have for adopting FOSS?*

This thesis explores the recurring phenomenon of Swiss SME companies relying on commercial software despite adequate FOSS alternatives being readily available. Furthermore, it focuses on the development of a reference model that permits systematic analysis of this phenomenon and could provide an answer to the research question.

1.2 Thesis Contribution

The contribution of this study is a reference model called Free and Open-Source Software Adoption (FOSSA) for analyzing and understanding the drivers and barriers for adoption of FOSS in SMEs, and for further theoretical, experimental, or applied science and engineering research. The FOSSA model is validated empirically through interviews with industry and research practitioners, to further understand the adoption of FOSS in the Swiss industry. The knowledge acquired from the interviews is used to further refine the model and make it applicable in practice as guidance for practitioners interested in questioning their own FOSS strategy.

1.3 Methodology

The methodology followed in this thesis was based on the Information Science Research Framework according to Hevner (2004). The IS research framework adapted to the OSS adoption in a Swiss SME is described in Figure 1 and it can be divided in three phases.

The first phase (Chapter 3) consists of a literature review structured towards a deductive approach. Through literature research, the theoretical basics of the topic are explored, which serve as a basis for the rest of the work and explain its context. For this step, literature on topics such as Open-Source Software, Free/Libre Open-Source ecosystems, as well as Open Innovation will be collected and analyzed.

The preceding literature research forms the basis for a qualitative analysis towards the investigation of the research questions. In a first step, the limits of the technology are researched in the context of Swiss SME companies, with the goal of selecting the most promising areas for an in-depth analysis.

Additionally, to gain more knowledge, I review Swiss Publications including journals, conference proceedings and other periodicals that cover the topics of interests. The advantages and disadvantages of open-source technology in Swiss SMEs are evaluated and opportunities and risks are considered.

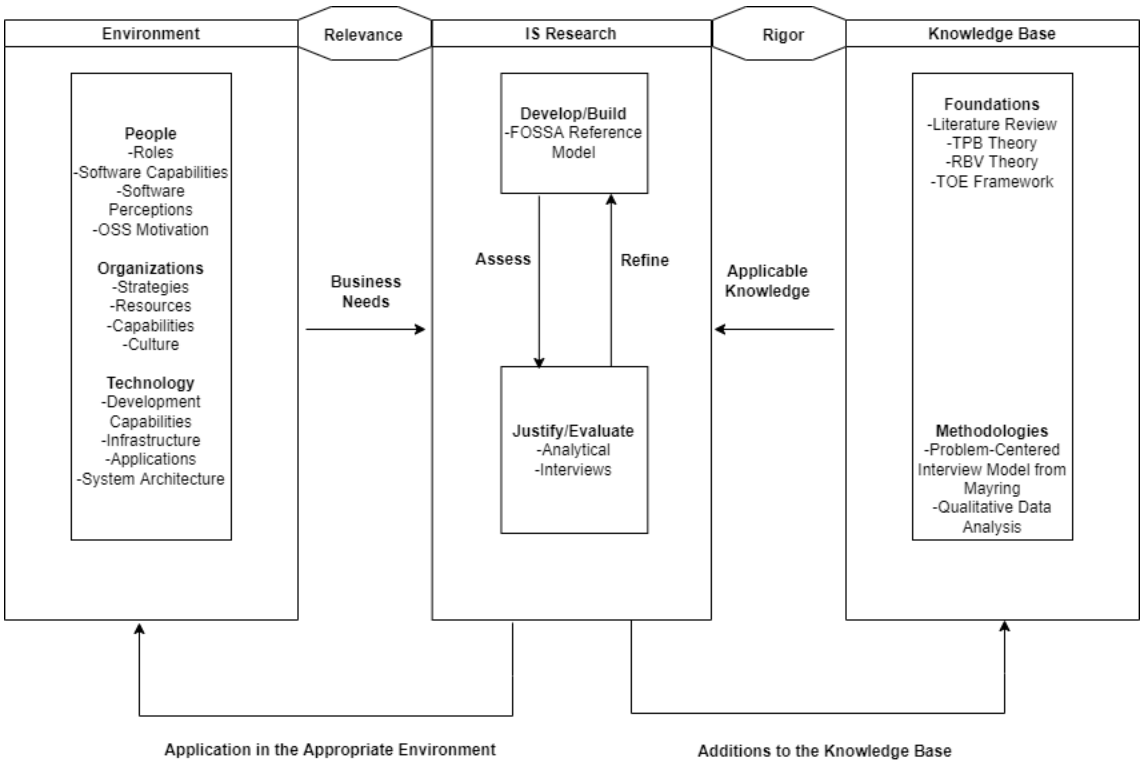


Figure 1: Information Science Research Framework based on Hevner et al. (2004)

The second phase (Chapter 4) is the creation and development of a reference model, FOSSA, that allows analyzing the behavior of individuals in companies with respect to open-source and how that behavior leads to the adoption or rejection of FOSS by the company based on its business needs, strategies, and goals, and considering the influence of environmental factors.

The third phase (Chapter 5) consists of using the findings from the first and second phases as a starting point and expanding them through a series of semi-structured interviews with Service Providers, a Cloud Expert, a Research Expert, an IT-Consultant and three additional interviews with CEOs. The interviews expand and refine the

findings by complementing them with information obtained from practitioners. To this effect, I have chosen the problem-centered interview model from Mayring (2016).

The interviews consist of open exploratory questions (see Appendix A) to give the respondents freedom to provide the full context of their experiences in the use of open-source technologies in their companies, or research work which might help to discover additional factors that were not foreseen before the interview.

The goal of the interviews is to enrich the findings from the two previous phases and modify the FOSSA reference model to include them.

2 Terminology

Terminology related to open source, open innovation and SMEs in innovation can have various interpretations, therefore this section describes the terms as they are used in this work.

2.1 Proprietary Software

The term proprietary software (also known as non-free software or closed software) is used when computer software is subject to some restrictions on its use, access to its binaries or source files. This type of software is usually regulated by licenses regarding its use, modification, sharing or redistribution of the source software. Proprietary software sometimes contains patented source code.

The first commercially available scientific computer was manufactured by IBM in 1953 (IBM Archives 2021). All available software was supplied by this manufacturer, and its customers could access the software but could not make it public. The first programming language, Fortran (Formula Translator), was also developed by IBM in 1957, but was only available to the company's customers. In 1969, IBM began charging separately for mainframe software.

The first personal computers with a graphical user interface (GUI) were launched by Apple in January 1984 (Behring Center 2021). In November of 1985, Microsoft Windows 1 was released (Gibbs 2014). The popularity of these two personal systems drove the growth of the proprietary software industry.

Proprietary software is the property of a particular company or legal entity. Usually, this company or person is also the producer of the software. Proprietary software is also referred to as commercial software.

Normally, the user does not become the owner of the software, but merely acquires a right of use. In the case of commercial software, certain rights of use and prohibitions are granted or imposed in the license agreement.

All software is written by programmers in a programming language. This text is known as source code. So that this source code is executable by a computer, it must be interpreted or compiled.

Commercial software products are delivered as compiled program code, the source code is usually no longer accessible to the user. The error removal and advancement of the program lie exclusively in the hands of the manufacturer. The source code is then kept secret and may only be available in the form of executable binary files or may only allow access to its functions over a network without the executable files being accessible at all.

The professional creation of software is very labor-intensive and requires high investments. Revenues are to be expected among other things from the sales of the license. From the manufacturer's point of view, the source code as a "raw product" represents an asset that is very much worth protecting and is usually not accessible to the later user who purchases a license. As it is stated in the KMU Portal (2021), the new term PropTech is proprietary software.

A program package is delivered, which is implemented on the target system by means of installation routines. Since software can be easily copied, there is an increased investment risk for the manufacturer due to unauthorized copying (pirated copies). Piracy is widespread, although unlawful copying of software is a criminal offense. The manufacturers of software try to prevent the non-legal distribution of their products by copy protection measures.

2.2 Free and Libre Open-Source Software (FOSS)

Even though Open Source and Free Software are two different approaches, the two concepts appear on the FOSS acronym. Free Software can be considered as an ideological approach while Open Source is more realistic (Costa 2013).

The term “free” represents the freedom that every user of the software should have to use, distribute, and modify the code. Users should be authorized to control the software that they are using. Moreover, giving this control to the user is the main goal of Free Software (Kuhn and Stallman 2001, p.1).

> *“Discussions of rights and rules for software have often concentrated on the interests of programmers alone. Few people in the world program regularly, and fewer still are owners of proprietary software businesses. But the entire developed world now needs and uses software, so software developers now control the way it lives, does business, communicates, and is entertained. The ethical and political issues are not addressed by the slogan of freedom of choice (for developers only).”* (Kuhn and Stallman 2001, p.1)

In the context of this thesis, the term “Free Open Source Software (FOSS)” is used to refer to software whose license is classified as a free software license by either the Free Software Foundation (FSF) or the Open Source Initiative (OSI) (The Open Source Initiative 2007). Alternatively, the terms “Open-Source Software (OSS)”, “Open Source (OS)”, “Free and Libre Open Source (FLOSS)” or “Free Software” are also interchangeably used.

The essence of FOSS is that it may be distributed free of charge and with open-source code. Thus, every user has the possibility to read the programming code of the software and the right to change it. The best-known example of open-source programs is the Linux operating system.

The definition of open-source software does not include programs that are referred to as public domain software, freeware, or shareware. With public domain software, the creator permanently cedes the rights to the program to the public but does not necessarily make the source code available. Freeware summarizes the software that may be used free of charge, but further rights or the source code are not passed on. Shareware may be copied arbitrarily. The use is however functionally, personally, or

temporally limited. The user receives the full range of functions, an unlimited right of use and commercial use only against payment of a license fee.

2.3 Types of software development

Software serves the purpose of making processes more efficient. It is therefore regularly only an aid but has often become indispensable. Today, even business-critical processes or decisions can often no longer be carried out without the use of software. Software can therefore be described as part of the digital transformation of an organization (González-Blázquez et al. 2021).

If there is a need for software, there are two main ways to obtain it. First, one develops and programs the software (self-development or in-house development) or second, one uses finished software. Ready-made software is also called standard software. If finished software is acquired as a license, one speaks also of commercial standard software. These are the two extreme positions. Between self-developed and standard software, another possibility exists. Frequently a combination of finished software and specific adjustments is used. One speaks in this case also of "Customizing".

Depending on the area of application, open-source software represents an alternative to standard software in that the open-source product can be used productively in the company unchanged or with only minor adaptations. Examples include the Linux operating system, Open Office or the Apache web server (OECD Publishing 2019).

However, open-source software can also be the basis for in-house developments. Since the source code is open and changes are not prohibited, it is entirely possible for an open-source program to give rise to complex software adapted to the company in question. The programming effort can be so considerable that one can speak of an in-house development (based on another product) (González-Blázquez et al. 2021).

Commercial open-source software is defined as open-source software that belongs to a for-profit organization (Riehle 2007). The company owns the copyright legal rights,

approves the specifications of the software, and its deployment processes. The MySQL database software[1] is an example of this kind of software.

Community open-source software is the software created by a group of volunteers that determine the specifications of the software, its deployment processes and how the software is approaching the solution of a problem or situation (Riehle 2007). For example, the Apache Web server[2].

Economic studies focused on community open-source labor can be widely found. However, a common result among them is that a reason for the contribution of a developer is due to personal gratification. The personal gratification of every person that contributes to the development of open source software solution is directly proportional to the increasing reputation among peers (Riehle 2007).

Developers who start or join an open-source project and can exhibit their social and technical abilities devoting to an open-source project become *committers*. Having the status of a committer increases the influence of a developer within the project; thus increases the possibilities of obtaining financial benefits and opportunities to get hired by companies that might let the committer work full-time on an open-source project. The latest will only be possible, if the open-source project is part of the business strategy of the company (Riehle 2007).

Several large companies have benefited from the use of open source. They support several open-source projects to see how the community reacts to their results (OECD Publishing 2019). If this reaction is positive, then there is an innovative product. This is how many projects have been created that now have proprietary components (e.g., CFEngine, RedHat).

[1] www.mysql.com

[2] httpd.apache.org

The scenario could look like this: One looks for an OSS that fits the business and system requirements as best as possible and modifies the program code to fit one's own business. There are some examples of SMEs using this approach (Startwerk 2010).

At first glance, this seems to be an advantageous approach, since the new complex system does not have to start from scratch, but already has finished program code that can be reused. In operational practice, however, it is hardly ever followed. The reasons for this are:

- The time required to familiarize oneself with a program that has been developed externally is considerable.
- The danger of side effects is great.
- The customized OSS program causes a high dependency on the application developers due to its specialty over the life cycle.

A further possibility, and one which is commonly used, is the in-house software development, but making use of available open-source libraries. The use of open-source in the form of libraries avoids the problem of having to familiarize oneself with the open-source code, since the library can be used through its documented interfaces. For example, in the area of Machine Learning there is a large number of available libraries such as Panda, Numpy and others, which are often used by companies in developing their custom machine learning or data science applications (Wen et al. 2019).

It is also important to note that among all existing open-source software licenses, one of the most popular is the GNU General Public License (Petrinja et al. 2013). The GPL is an example of a so-called restrictive open-source license. Restrictive licenses require that any developer who makes changes to GPL-licensed software must publish their changes under the same terms of the GPL.

This means that a company that wants to use GPL-licensed software in its software architecture must release it. This regulation, of course, becomes a problem for any

company that wants to use GPL-licensed code. Large companies know this and make sure that they do not use these components in their main software architectures (Synopsys Editorial Team 2019).

2.4 Open Innovation

Open innovation is defined as "a distributed innovation process that relies on purposively managed knowledge flows across organizational boundaries, using pecuniary and non-pecuniary mechanisms in line with the organization's business model to guide and motivate knowledge sharing" (Chesbrough and Bogers 2014, p.1). From the time that Chesbrough created the term Open Innovation in 2003, it has transcended from academic research to the industry practice (Chesbrough and Bogers 2014).

Open innovation mechanisms have been described as tools that are "purposive inflows and outflows of knowledge" (Chesbrough and Bogers 2014, p.3), i.e., tools that companies use to gain external ideas and technologies to implement in their business and to let unused internal ideas and technologies to be used by others in their business. Some examples of open innovation mechanisms are network, communities, and innovation contest, and crowdsourcing.

The concept of open innovation with respect to a company focuses primarily in using external knowledge to enhance the internal innovation processes and to grow the organization's performance (Leckel et al. 2020). On the other hand, open source has also helped companies by designing, developing, and collaborating on tools that are affordable and accessible. Both concepts help companies of all sizes to achieve innovation, especially for small-medium business.

2.5 The role and characteristics of SMEs in innovation

The role of SMEs in innovation has been encouraged and supported by the SME Policy and the Regional Dimension of Innovation (SMEPOL) project. This project was managed by seven European research teams, and funded by the European Commission under the Targeted Socio-Economic Research program (Asheim et al. 2003).

A distinctive characteristic of SMEs related to innovation is the fact that they struggle to overcome barriers to achieve their potential contribution to the innovative capabilities of their local, regional, and national economy.

According to the SMEPOL project, the definition of an SME is: "very small enterprises, with less than 10 employees; small enterprises with 10-49 employees and a turnover of less than €7m; and a medium enterprises with 50-249 employees and an annual turnover of less than €50m" (Asheim et al. 2003, p.3; Eidgenössisches Departement des Innern EDI Bundesamt für Statistik BFS 2021). In Switzerland, this definition of SME was also adopted; however the criteria of the turnover from the company is not considered on the formal definition of an SME according to the Swiss Federal Department of Home Affairs (FDHA) (2021).

The first SMEs were born in the UK in 1970 as a small business activity. Since this year, SMEs are considered a source of employment on several countries. Even though, recessionary periods occur that cause job losses, studies have shown that SMEs' performances in terms of employment, during these trying times in market economies, have remained stable in comparison to large companies (Asheim et al. 2003).

An increase in the business services had led to the growth of the importance and creation of small and medium companies. These companies have used the opportunity to fulfill the needs of the consumers to create, maintain or distribute a diversity of services. In Switzerland, the number of very small companies have grown especially in urban areas and there is a trend where SMEs' main business area is the service sector. The integration of these companies in the global value chains play an important role (Eidgenössisches Departement des Innern EDI Bundesamt für Statistik BFS 2021).

3 Related Work

3.1 Free and Open-Source Software and Commercial Software as competitors

Free and Open-Source Software (FOSS) has been extensively explored in the literature. Articles about its transformation, understanding the concept, understanding the motivation of its collaborators and developers, licenses, etc., have been available for decades. For instance, the article by Miller et al. (2010), focuses on the history of FOSS. The main statement written in this article is that FOSS and commercial software have a symbiotic relationship. The rivalry between them makes it possible to keep the software marketplace under control (Miller et al. 2010).

Sometimes the competition between proprietary software and FOSS has led companies to issue provocative statements about the characteristics of FOSS. Thus, some misconceptions have been created. Costa (2013) wrote a journal paper about FOSS prejudices. According to Costa (2013), there are some perceptions that FOSS can be considered piracy or that it cannot be use by government or companies due to quality or lack of support.

The author defined four freedoms that involves what "free" in the FOSS acronym means. He states that free means:

- freedom to execute the program
- freedom to examine how the program works and modify it to adjust someone needs
- freedom to redistribute copies of it
- freedom to distribute copies of the modified versions

Therefore, the prejudice that OSS is piracy is a misconception because most of OSS works with copy-left practice that grants the rights to work freely with the software with the requisite that whoever distributes it, must do it under the same conditions.

Costa (2013) promoted that nowadays, there are many companies that profit distributing open-source and their earnings are based on their services. These companies create premium versions while distributing OSS.

Other OSS adopters who profit by using OSS are national and local government agencies, primarily on the fields of health informatics or computer forensics. Moreover, Cloud solutions, Organizational Systems and Big Data, Data Mining and Business Intelligence are also fields where government organizations and companies implement OSS (Costa 2013).

OSS is also embracing new technologies in Cloud computing such as Function as a Service (FaaS) or Serverless. FaaS is coming out as a novel technology where not only public cloud service providers such as AWS, Microsoft Azure or Google Cloud are creating their own infrastructure, but also the OSS community has been implementing initiatives (Balla et al. 2020).

Platforms such as OpenFaaS, Kubeless, Fission and Knative are some examples of the new FaaS solutions where OSS has an active role. The most used languages of open-source projects are Node.js, Python, Java and Golang (Balla et al. 2020).

3.1.1 OSS projects vs Commercial software products

Studies about competition between proprietary and free open-source software have been written focusing on several aspects. The focus of the study written by Lee et al. (2008) is the competition between commercial software products and free and open-source projects. They explained how a first-mover advantage by the commercial firm helps them to implement a “divide and conquer” strategy to charm early adopters and capitalize on late adopters.

The commercial software company can have a first-mover advantage only if the technologies are incompatible. Incompatibility works as an advantage to the commercial software company, when the network effect of the open-source software declines and the FOSS developers do not obtain sufficient benefits to compensate them for their efforts, i.e., they do not get enough personal satisfaction from developing the software. In this case, FOSS developers would allow the commercial software company to win the total market. Thus, to exploit its first-mover advantage, the organization must grow its development investment to refine its product features (Lee and Mendelson 2008).

In case that the open source software enters the market as a first-mover, “the commercial firm is always better off mitigating the network effects by following the open source lead and designing a compatible product” (Lee and Mendelson 2008, p.26).

Lee et al. (2008) stated that “because software is often characterized by network effects, compatibility and sequence of entry become key issues, especially in a heterogeneous market” (Lee and Mendelson 2008, p.13). A key result in this study is that open-source software can be considered as a threat to pressure the commercial software vendors to reduce their prices. Hence, the importance of the open-source software in the landscape of the software industry.

3.1.2 OSS vs Proprietary software as a digital strategy

Competition between proprietary and free open-source software focused on digital strategy was written by Gupta (2018). Gupta states that it is always a challenge for a company to choose between proprietary and free open-source software. However, the choice should not always be “black and white, as there are various parts of the platform that can be kept open or proprietary” (Gupta 2018, p.74).

The kind of software that a company uses should depend on its objectives and strategies according to Gupta (2018). He further states that open-source system software creates a larger market (e.g., the Linux operating system), and closed system software creates a better customer experience (e.g., the macOS operating system). Moreover, large companies such as Google, Apple and Samsung are using open-source mobile-payment systems to build a large market and to foster greater innovation. Whereas Starbucks created a successful closed software system for mobile payment to enhance customer experience that benefit mainly Starbucks frequent visitors (Gupta 2018).

Benzerga (2016) created a pattern map that shows the importance of software in a digital transformation strategy. Extending from these ideas, I argue that using OSS can have a direct impact on hardware and connectivity, which in turn has an impact on cost. However, this does not mean that the quality would decrease.

3.1.3 OSS vs Proprietary software on Risk Management for Information Security

A study, about competition between proprietary and free open-source software focused on risk management for information security, was written in the technical report by Rees et al. (2006), where it is stated that the selection of software makes a difference in this field. The authors' goal was to provide empirical results to prove if the common belief that OSS is more secure than proprietary software had any scientific foundation.

Their results were inconclusive because their study was restricted to "purely technical concerns, and to just the vulnerabilities found in operating system applications" (Rees et al. 2006, p.26). The authors could show that proprietary and open-source software had about the same vulnerabilities, and they could not find any patterns that indicated an advantage from one software over the other from a security point of view.

On the contrary, Hoepmann et al. (2007) discussed the impact of open source on security and transparency on a software system (Hoepman and Jacobs 2007). The authors mentioned that what determines a system "secure enough" is the risk associated with the use of the system. For this purpose, they defined risk as "a combination of the likelihood of a successful attack on a system together with the damage to assets resulting from it" (Hoepman and Jacobs 2007, p.80).

The authors argue that when the source code of a software is exposed, the security of the system does not increase, mainly because it does not change the code, but it makes any existing vulnerability more visible. They go further stating that exposing the source code of the software makes the process to find the solution more agile and forces developers to increase the quality of their code. Thus, open-source software permits independent assessment of the exposure of a system and of the risk associated with using the system.

Open source offers the benefit of providing information to its users to make a more educated decision about the security of the system, either based on their own or by an independent judgment (Hoepman and Jacobs 2007).

On the other hand, Silic et al. (2016) argue that past research has identified that OSS has security risks, lack of expertise, compatibility, ownership and training issues. The authors stated that the reasons of the failures in an OSS project "lay in the origins of the OSS model, which was conceived on the foundations of freedom" (Silic and Back 2016, p.152). They go further announcing that "freedom and choice introduce risks" (Silic and Back 2016, p.152), and the risks might lead to expensive consequences.

According to Silic et al. (2016), the lack of good risk management is one of the crucial elements that an enterprise should take into consideration if they want to implement an OSS solution (Silic and Back 2016). The authors argue that OSS enterprise adoption is popular with selected software segments such as the web server market, server software, and smart phone market.

Silic et al. (2016) go further suggesting that hidden costs such as hiring expert support for end-user training and the lack of cost-benefit analysis might be two of the reasons that companies ended not adopting OSS, and enterprise adoption of OSS is still minimal. Nevertheless, the authors found through literature review that the results of the study of potential risks caused by an OSS adoption are inconclusive and more clarification on the factors that motivate the proper stakeholders to adopt OSS need to be done.

In this study, the authors proposed a theoretical model based on the Theory of Reasoning Action (TRA). One of the goals for the construction of this model was to clarify the perceived IT security risk (PISR) in relation with adoption intention (AI) in the OSS context. The results attested that there is a positive correlation between PISR and AI, which means that the risk factors that could result from adopting OSS have a strong influence on the adoption intention.

Among other results from this study is that "loss of privacy" is the major distress from companies that are considering adopting OSS. The authors sustain "opportunistic OSS behavior" (Silic and Back 2016, p. 157) as one of the worries from companies to adopt

a software because it has been modified by unknown people without any credentials. This worry directly affects the motivation of decision-makers to use OSS because the protection of information is paramount to companies.

3.2 Open-Source Adoption

3.2.1 Open-Source Software Adoption in an Individual

How exactly a person adopts a proprietary software product, an OSS solution or both as a hybrid ecosystem can be considered, according to Li et al. (2010), by examining the adoption intentions of the non-adopters of FOSS solutions and the extent of adoption for the adopters. By using the Self-Determination Theory (SDT), the authors created a theoretical model whose goal is to capture the motivations and their influence on the individuals to adopt OSS.

The study differentiates three types of motivational elements: intrinsic motivation, extrinsic motivation and amotivation; noting that intrinsic and extrinsic motivation are not the opposite to amotivation. In the journal article, amotivation is defined as “a state in which people lack the intention to behave”, and “external locus of control” (Li et al. 2010, p.78). The main statement in this section of the study is that individuals with lower levels of motivation usually exhibit external locus of control in their behavior.

The authors proposed two theoretical models. One of them would capture the intrinsic and extrinsic motivations and their influence on the individual to adopt OSS. The second model would size the core nature of the amotivation factors and their influence on the intentions of a person for the adoption.

There was a bias on the results on the study since the authors handled the same list of questions to the adopters and non-adopters of OSS. This “operational problem” as they named it (Li et al. 2010, p.87), caused confusion among the subjects of their study leading to the necessity of future research to build a better design to capture the intrinsic and extrinsic motivations for non-adopters.

3.2.2 Open-Source Software Adoption in a Company

The benefits of free and open-source software are attracting the attention of production environments as well as research and software development (Karels 2003). Prior research gives deep understanding of the FOSS adoption phenomenon. For instance, Macredie et al. (2011) presented a theory-grounded framework based on the Decomposed Theory of Planned Behavior (DTPB) to explain the factors influencing OSS adoption, the context of study being small and medium-sized (SME) information technology (IT) businesses in the United Kingdom. It focuses on the eight belief structures of the DTPB theory. The constructs of this model are defined in a mixed individual and organizational level perspective.

The authors conducted case studies on UK-based businesses and stated that OSS is useful to SMEs because of its financial and human resource benefits. Their research led to several results, including that license cost saving is seen as a motivating factor and an economic benefit in the adoption of OSS. They explained further, these economic benefits are valuable to SMEs as they usually have limited IT budgets and a scarce supply of IT resources.

A second important result from the study of Macredie et al. (2011) is that the support community plays an important role on the normative beliefs of the involved stakeholders that ultimately make final decisions on the adoption of OSS in their SMEs. This result coincides with the Swiss Open Source study (2021), where it was stated that the benefit of having a helpful community in many open source projects is one of the central reasons for the Swiss companies to use open source software. This benefit involves the exchange of knowledge and experience among a diversity of individuals and creates the basis for professional support and eventually the opportunity to acquire experienced open-source experts. It also influences normative beliefs of decision makers to the use of FOSS. In addition, Macredie et al. (2011) argue that this influence contributes positively to the intention of use.

Capital investment is another factor for the adoption of OSS mentioned also by Macredie et al. (2011). The authors believe that capital investment is important because

it provides “computing and network infrastructure to support communication and other business activities” (Macredie and Mijinyawa 2011, p.246). They quoted a manager stating that if the SME does not have human resources with open-source skills and expertise, it is mandatory to hire or outsource these resources. This factor is relevant because it supports the statement from the 2018 survey conducted by The New Stack and The Linux Foundation (OpenExpo Europe 2021), where it was stated that the lack of information about the possibility of using an open-source system as well as the lack of time to develop the skill prevents SMEs from adopting FOSS ecosystems.

Macredie et al. (2011) also found that time was a key factor for the adoption of FOSS, because time is needed to develop expertise and lead the adaptation to the new OSS ecosystem. Capital investment is an essential facilitating condition that contributes positively to the intention to use.

Remarkable is the fact that Macredie et al. (2011) mentioned the internet infrastructure as a factor that facilitates the adoption of FOSS in SMEs. After eleven years, this factor is becoming obsolete, as an SME can avoid having to manage its own computing and connectivity infrastructure, for example by using cloud services.

3.2.3 Open-Source Software Adoption in Government

Huysmans et al. (2008) conducted a case study on the Belgian Federal Public Service (FPS) Economy, and identified different barriers that hinder the adoption of OpenOffice in similar backgrounds. In their study, the authors stated that “the absence of any drivers towards the use of open-source desktop software is not the only possible reason for non-adoption. Instead, other factors may limit the tendency of the organization to adopt” (Huysmans et al. 2008). They based their study on the Technology-Organization-Environment (TOE) framework.

Among the drivers that triggered the use of OpenOffice was that the ICT manager had a personal belief that open-source applications such as OpenOffice could be implemented with success in an organization. Thus, user attitude regarding the open-source software was positive. Cost reductions and government guidelines were also mentioned as

drivers. Moreover, external influences activated the interest of FPS Economy in the adoption of OpenOffice.

One result of this study is that the main factor in the decision to not adopt open-source application software for their case study was change management, because there was a short-time need for the changes to be implemented and assimilated for the organization. The adoption of the open-source application software would have not only been time-consuming but also work-flow unstable and non-compatible with the existing desktop environment configuration of end users.

Having already stated that the issue of change management in the organization context was the main barrier, the authors categorized additional barriers that hindered the adoption of open-source software. Among them in the technology context is the integration with third party software, in the organizational context is the lack of internal knowledge and in the environment-context is the decision of similar organizations as the FPSs.

The authors go further with the following findings:

- the characteristics of the technology should not be the only consideration towards the adoption of open source; the organizational and environmental contexts have the same level of importance as the technological context.
- the existence of network effects can also play a role in the adoption decision and even if a positive attitude towards open-source software is in place, the adoption decision might be overruled.

At the end of this study, it is stated that the FPS Economy opted for a hybrid approach to provide its users the ability to open most of the files without problems.

3.2.4 Open-Source Software Adoption in Education

Concerning organizational context, the adoption of open source is not limited to the industry, or government organizations. Educational organizations have also made use of the advantages of open-source and there are several articles written about this

phenomenon. For example, Pfaffman (2020) wrote a journal article about the benefits that FOSS applications offer to high school science classrooms.

The network effects are also mentioned as a factor in the study of Pfaffman (2020). The author argues that if there is a community willing to join forces towards unified goals in FOSS, this software will benefit because a crowd of programmers will be willing to collaborate. One result is that the perceived ease of use of open-source applications in K-12 institutions is greater than the perceived ease of use of commercial software. This study lists different science related open-source applications that show how the perceived expectations of open-source software efficacy and efficiency are relevant for education purposes.

Pfaffman (2020) clarified further about the phenomenon of FOSS adoption in K-12 schools. The author explains that digital sovereignty is the major advantage that FOSS offers to the learning community. By teaching ICT lessons based on commercial software, the educators are giving value to that specific software, making a good business for the software vendor, which in return offers major discounts to the schools. However, the students might need in the future to buy it or even steal it to carry on with the skills learned in school. On the other hand, when students learn and use FOSS, they also learn that they can freely evaluate and later choose the software tools needed for their daily routine.

3.2.5 Open-Source Software Adoption Advantages

The study written by Hedgebeth (2007) focused on gaining competitive advantage by using open source software. This study stated how the use of open-source software can reduce the total cost of ownership (TCO) in an enterprise; thus, the enterprise can benefit by meeting its organization goals in a more efficient and effective manner.

In addition, the author wrote that open-source software is usually chosen by European organizations due to the free access to its source code which can be inspected by security specialists to identify vulnerabilities that can be openly announced without any legal repercussions. According to the author, the Chinese government disclosed a

SWOT analysis that revealed that in the information security area, transparency of open source played a major role while deciding its adoption because it shows all its source code, reducing any government concerns.

He mentioned, how private software companies such as Microsoft Corporation have perceived the open-source movement as a threat, while other companies such as IBM, Oracle and Apple computer have created alliances with the OSS community, since they can profit on their business operations as well as their client's needs (Hedgebeth 2007).

Microsoft has since 2016 built some bridges between its organization and the open-source community. This commercial software company has introduced open-source software to its products, such as the Windows Subsystem for Linux (WSL) (Lewis et al. 2018).

Additional studies have been written about the advantages of OSS and how these are perceived by its users. Paulson et al. (2004) was able to support, through an empirical study, that the perception that errors are fixed quickly in open source projects is true. Also, this study had as a goal to investigate other common perceptions about open-source projects such as the belief that open source grows more quickly. The results turned out to be inconclusive as there was no evidence to prove this hypothesis.

On the other hand, the authors could find evidence that the perception of creativity is more common on open-source projects than in closed source projects. In addition, a result from this study validates the hypothesis that external global factors and feedback mechanisms have a major impact on the evolution of the software, instead of the economic model of the production of the software itself.

The authors conclude that before a company adopts either an open source or a closed system, it should define its goals, strategies, and the appropriate metrics that the company will use to monitor that these goals are achieved.

FOSS offers software companies the opportunity to attain economic gain through the implementation in a systematic way of its components and products. By reusing FOSS, these companies can reach software development productivity and product quality (Ajila and Wu 2006).

Ajila et al. (2006) attested in their study, that FOSS components are of prime quality and that the FOSS community is setting a good example in terms of good practices. FOSS has made a shift from being successful in the low-level infrastructure to the middle layer and component-based development and reuse. Johansson et al. (2008) go further arguing that FOSS systems are now focusing on business application such as Enterprise Resource Planning (ERP) systems.

The arguments of Johansson et al. (2008) are based on literature review. In their results they mentioned that one might think that the cost factor is crucial for the adoption of open-source ERP systems by an SME; however, this conception is false as the cost factor has a secondary role. The authors found that FOSS-developed ERP systems are an alternative since they offer quality and reliability. A second reason for an SME to adopt ERP systems based on FOSS technology is that they are highly flexible and adaptable to change.

Even though OSS in ERP systems offers all these advantages plus zero cost of software, what might hinder the adoption of OSS in ERP systems is the consultancy service of an expert. The payment to the OSS expert might be twice as much as a payment to a Microsoft software specialist. Therefore, if the SME does not have an in-house OSS expert that can customize and maintain an OSS-ERP system, the saving cost for the OSS-ERP system is not an advantage anymore.

The authors make a point stating that if proprietary ERP software does not meet the needs and specified standards of the company, they can legally prosecute the information technology provider or software manufacturer. Whereas in an OSS-ERP system, there is no defined entity to assume legal responsibility (Johansson and Sudzina 2008).

3.3 Service Providers

Information technology service providers offer solutions that address the uneasiness that a company might have with IT problems, liberating the company to focus on its business (Riehle 2007). This statement coincides with the Swiss Open Source Study (2021) where respondents stated that the perceived availability of open-source support providers is an important factor in the adoption of OSS. Support for the introduction of open-source software, its maintenance and future development is a prerequisite for the use of open-source solutions in a business-critical environment.

Riehle (2007) explains how a service provider benefits while selling open-source software. The author argues that normally IT service providers profit depending on what part of the IT service components they own. These components can be either hardware, software, or services. Usually, an IT service provider owns its service components and must buy hardware and software; therefore, its profits increases are inversely proportional to the cost of hardware and software. If these two components can be acquired as inexpensively as possible, the economic benefit for the IT service provider will grow.

Riehle (2007) further argues that IT service providers would rather use community open source over commercial open source software. He states, “only community open-source software prevents vendor lock-in” (Riehle 2007, p.27), because community open-source software makes certain that the cost of software support be dependent of the free market instead of only one corporation.

There are major similarities between being part of the team of a start-up and being part of the community of developers of an open-source project according to Riehle (2007). In both cases, the earlier a person joints the team or the community, the greater the probability of the upgrade to committer status, and the greater is the risk of failure of the project (Riehle 2007).

In other words, depending on the kind of software that a company uses, its platform develops a business ecosystem of partners that will work together to support new products, satisfy customers' needs and hopefully will create the next wave of innovation. A business ecosystem of partners should not only be represented as IT service providers, but also entrepreneurs, research institutions, public sector, role companies' models and even many independent players. Concerning to this business ecosystem of partners is open innovation.

3.4 Open innovation

Leckel et al. (2020) explored the phenomenon of how public policy could effectively increase collaboration for innovation among business ecosystem of partners "in a way that is easily accessible and beneficial for SMEs" (Leckel et al. 2020, p 1). Among their results through literature review, the authors found that resource limitations and the low ability at which an SME can learn and use external knowledge are the barriers that hinder small-medium companies to benefit of any open innovation advantages.

The authors searched for pragmatic instruments to encourage regional innovation systems such as "hackathon" events. They were also active in finding the fundamental barriers preventing SMEs from participating in open innovation. For example, they compared how a large company can afford to hire "knowledge gatekeepers" (Leckel et al. 2020, p. 2) to search for opportunities and win knowledge transfer. An SME does not have this opportunity due to its budget limits.

A second constraint that prevents SMEs from adopting open innovation according to Leckel et al. (2020) is the "Not-Invented-Here Syndrome" or the challenge for research and development (R&D) professionals. The authors go further stating that SMEs have mostly the problem of "professional identity from problem-solver to solution-seeker" (Leckel et al. 2020, p.3).

The fear that competitors will gain access to their knowledge when revealing their technical problems is another constraints that SMEs usually have (Leckel et al. 2020). This fear can be overcome if there is an intermediate network approach. Leckel et al.

(2020) called them intermediaries that help to support the collaboration among the parties during the innovation process.

According to Leckel et al. (2020), to win knowledge transfer for an SME it is important to not only study open innovation on the organizational level, but also on the systematic level of the economy. In this sense, it is necessary to implement regional innovation policies where a business ecosystem of partners can be involved in the process. The authors proposed two types of intermediaries to support SMEs through open innovation process. The first type of group of intermediaries is a regional institution and the second is an open innovation expert (Leckel et al. 2020).

Expert knowledge, experience and creative ideas are constructs that are part of a conceptual model of a local open innovation (Leckel et al. 2020). According to the Swiss study of Strategy Development in the Digital Age (2021), there are several Swiss SMEs that have started developing their strategy with elements of open innovation. These companies defined their business goals by using open innovation mechanisms to focus on their customer's needs.

According to the Strategy Development in the Digital Age Study (2021), entrepreneurs that think in terms of customer-centric business model innovations will be successful in the digital age.

However, not all Swiss SMEs are ready to adopt open innovation (FHNW-Strategylab-Strategieentwicklung-Digitales-Zeitalter 2021). In his review of two books, Evans (2003) states that every SME has different problems with respect to its regional environment; nevertheless, "elements of good practice in entrepreneurship and innovation policy can be imported" (Evans 2003, p.424,425).

Evans (2003) mentioned that key policy responses should be created to support a knowledge-based economic strategy to stimulate a more innovative SME sector at a regional, national, and European level.

3.5 Open-Source Research

Aksulu and Wade (2010) created a holistic tree-dimensional framework of open source research. This framework is entirely theoretical, representing the normative view of open-source research. It states that linkages between pure proprietary and open systems exist, creating hybrid types of systems.

Furthermore, Aksulu and Wade (2010) argue that most open-source research has centered its attention on software development, and the ecosystem that encompasses it, and there is a scarcity in organizational research of the adoption, acceptance, and use continuance or discontinuance behavior of open source products or services (Aksulu and Wade 2010).

Therefore, this thesis work focuses on the creation and development of a reference model, FOSSA, that can provide a better understanding of the behaviors of individuals in companies with respect to the adoption of FOSS and how these behaviors lead to the adoption or rejection of FOSS by their companies based on their business needs, strategies, and goals. In addition, the influence of environmental factors is also considered.

4 Theoretical Model

This study follows the conceptualization of growing the open source software adoption process rather than building it in a single step (March and Smith 1995). The difference is that when the adoption process of FOSS is built, one foresees the specification of the entire plan and then constructs the adoption of FOSS according to this plan. When the process of adoption of FOSS is grown, it is developed incrementally (Brooks 1987). This means that the development of the adoption of open-source software is conceptualized in this study as dynamic, i.e., it is constantly evolving rather than static which is terminated in a lapse of time. This approach more closely reflects reality, since in practice organizations cannot implement an entire adoption concept at once, but rather start with the minimum requirements and incrementally adapt their systems according to their needs.

Until now, the studies of OSS adoption have been focused either individual or in organizations. The FOSSA reference model, proposed by this thesis, is designed to achieve a comprehensive understanding of both the individual and the organizational factors that determine the adoption of open-source software in an SME.

Open-Source software adoption is divided in two different areas of implementation. First, there is the open-source system software (also called back-office software) adoption and second is the application software adoption (also called front-office software).

The adoption of open-source system software in an SME can generally be driven by the preferences and skills of a small group of individuals, whereas the adoption of open-source application software is driven by the needs of the whole organization. For example, the adoption of Linux for running the company servers can be driven by the skills of the IT organization. As a counterexample, the adoption of Open Office for use throughout the company needs to be driven by a competitive advantage for the whole company and requires training and time from the entire workforce. These resources are particularly scarce in an SME because time and training need to be focused on the core business rather than on a supporting function.

The intention to adopt an Open-Source system software in an SME, starts when at least one person's attitude towards the open-source system software is positive, i.e., the individual has a positive experience with the proposed open-source system software; therefore, he favors the implementation of this OSS solution in the company. This statement is based on the definition of attitude according to Ajzen et al.:

> *"we use the term attitude to refer to the evaluation of an object, concept, or behavior along a dimension of favor or disfavor, good or bad, like or dislike"* (Ajzen and Fishbein 2010, p. 78).

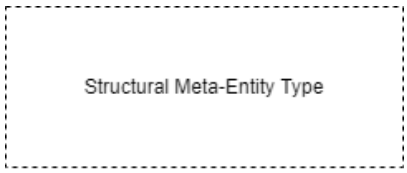
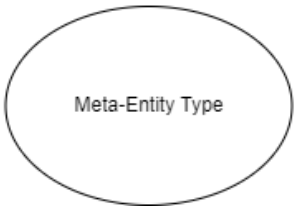
Thus, the Theory of Planned Behavior (TPB) OSS adoption framework has been used (see Section 4.2). For the application software adoption, it needs to be an organization that would be willing to adopt it; therefore, the Resource-Based View (RBV) theory is used (see Section 4.3). These two theoretical frameworks are linked in a new model using the Technology-Organization-Environment (TOE) framework (see Section 4.4), which forms the theoretical contribution of this thesis, called the *Free and Open-Source Software Adoption* (FOSSA) reference model.

4.1 Free and Open-Source Software adoption (FOSSA) reference model

The proposed *Free and Open-Source Software Adoption* (FOSSA) reference model was created based on the Theory of Planned Behavior (TPB) and the Resource-based View (RBV) Theory. These two theories were unified using the Technology-Organization-Environment (TOE) framework to produce a comprehensive reference model for the purposes of this study.

According to Gutzwiller (1994), the components of a reference model can be represented using the notation shown in Table 1 (Gutzwiller, Thomas, A. 1994, p. 24).

Table 1: Metamodel component notation based on Gutzwiller (1994) .

	A structural meta-entity type component is used to describe the complex composition or outline of people and companies.
	A meta-entity type component serves to represent the primary constructs of the phenomenon described by the model. Every meta-entity type has a short textual definition that describes the essential attributes and relationships of every metamodel component.
	The arrows in the diagram describe the influences among different components involved in the model, so relationship is indicated in each case.

The proposed *Free and Open-Source Software Adoption* (FOSSA) reference model is shown in Figure 2.

The reference model illustrates the behavior of individuals that belong to an SME towards the adoption of FOSS and aims to explain the importance of IT and non-IT related factors that influence the performance of a company in the marketplace. Therefore, the relationship among the software used by the company to fulfill its needs, the organization strategies and environment factors are combined with individual perceptions of behavior that directly influence the decisions of the company towards the adoption or non-adoption of FOSS.

The following sections describe in detail the various parts of the FOSSA model and the theoretical background in which they are based.

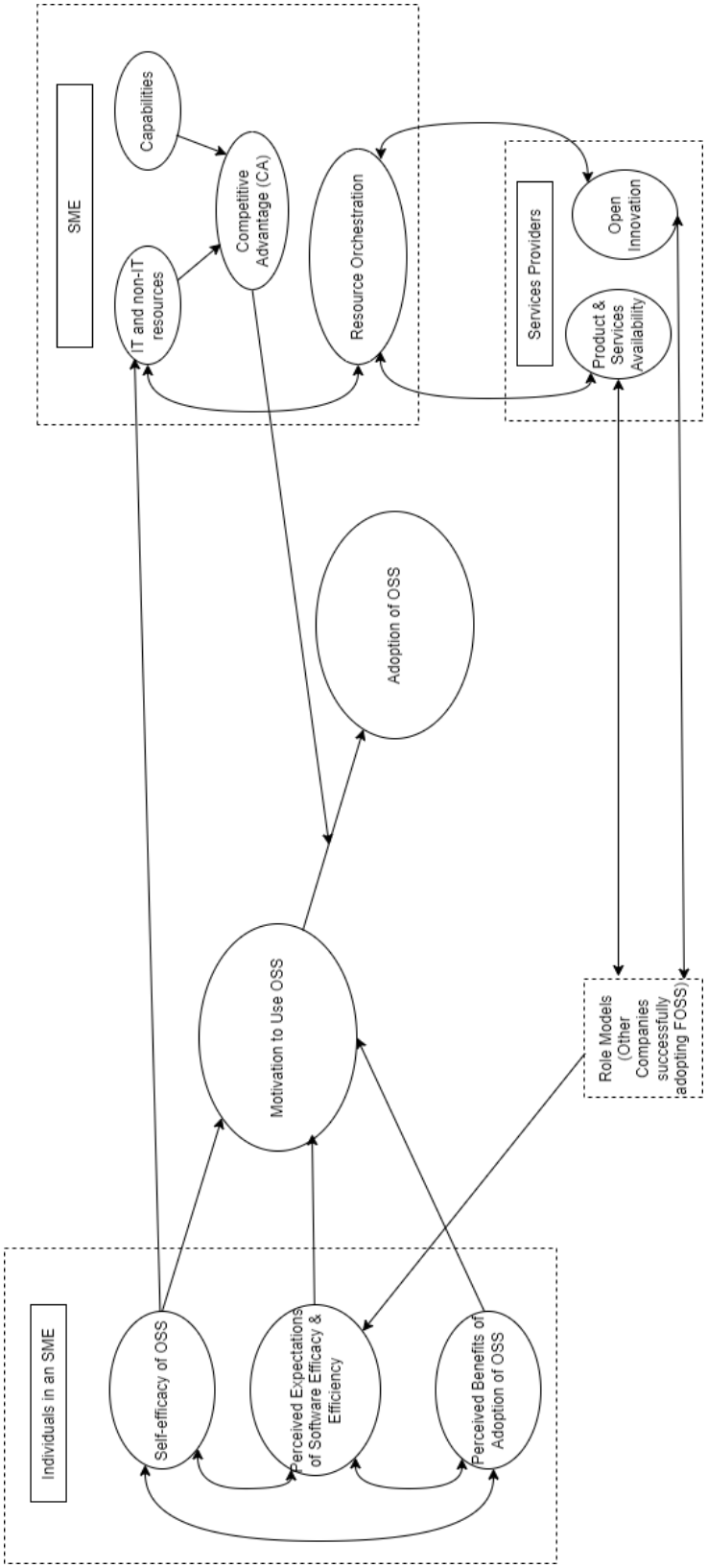


Figure 2: The Free and Open-Source Software Adoption (FOSSA) reference model.

4.2 Theory of Planned Behavior adapted towards Open-Source Adoption for Individuals in a Company

Human behavior has a keen impact in organizations to which they belong. The decision that an organization makes to engage in developing, implementing, adopting, and participating in an OSS community can also be seen as a social concept. The purpose of this section is to provide a model that can guide us to understand the behavior of an individual that could result in a company to adopt OSS.

To better understand the barriers or enablers for a company to adopt Open-Source Software, I adapted the Theory of Planned Behavior (TPB) (Ajzen 1991). The resulting model is shown in Figure 3.

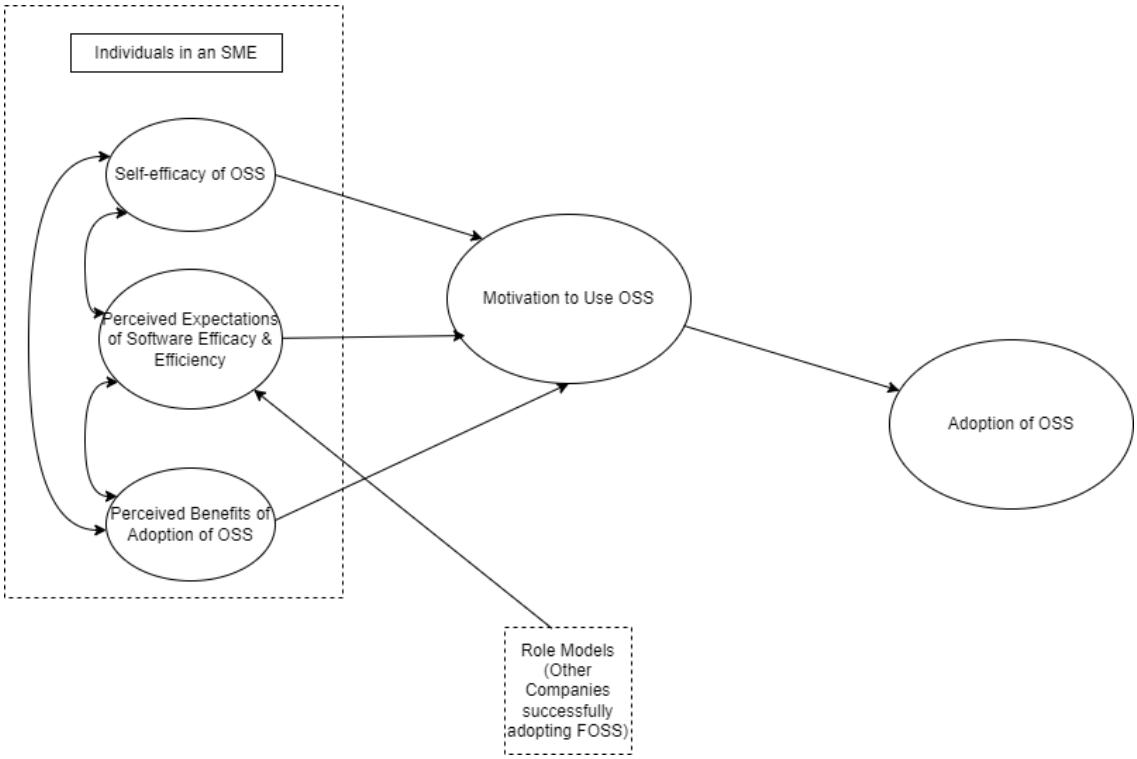


Figure 3: TPB for an SME to adopt Open-Source Software based on (Ajzen 1991).

By using the Theory of Planned Behavior (TPB), this thesis work aims to evaluate the behavior of individuals that work for an SME along a dimension of the approval or disapproval of OSS adoption in their companies, and to create a model that encompasses any judgments related to this phenomenon from these individuals (Ajzen and Fishbein 2010).

4.2.1 Perceived Benefits of Adoption of OSS (Attitude toward the behavior)

This construct represents the benefits of OSS adoption as perceived by individuals in a company. The following fourteen benefits have been listed according to the Swiss Open-Source Survey (2021).

1. Open standards and interoperability
2. Extended open-source components and tools
3. Leveraging a community of likeminded individuals with similar goals or needs
4. Increased Security & Business Continuity
5. Digital Sovereignty
6. Increase Stability
7. Costs savings
8. Increased Innovation
9. Transparency and trust through access to source code
10. Digital sustainability, release of digital knowledge
11. Open-Source providers
12. Flexibility: Easy adaptation to own needs through access to source code
13. Employer appeal
14. Promotion of the local IT industry

4.2.2 Perceived Expectations of Software Efficacy & Efficiency

This construct indicates the expectations of OSS adoption with respect to how well the software will perform its functions, and the efficacy with which it will perform them.

1. The beliefs of the individuals within the company, that the key external stakeholders involved in its digital strategy processes (e.g., customers, suppliers, vendors, etc.) will find their interactions with the company to be efficient and efficacious after adopting OSS.
2. The beliefs of the individuals within the company, about whether the parties have an efficient and efficacious digital performance themselves, so that the company's digital daily processes would be jeopardized if OSS is implemented.

These descriptive and normative beliefs produce perceived standards, or what could also be seen as a perceived “social pressure” to engage or not in the adoption of OSS instead of closed software.

4.2.3 Self-efficacy of OSS (Perceived behavioral control)

The self-efficacy construct is a representation of how confident individuals feel about their own abilities to successfully use OSS and the control they will have over the process. It may include the following factors:

1. Members of the organization’s perception about the ease or difficulty of adopting OSS.
2. Expectancy of success is the perceived probability of succeeding at adopting OSS in the company.
3. Confidence in their ability to adopt a complete OSS ecosystem in the company.
4. Culture of the organization plays an important role. Is the organization open to try new things?

Self-efficacy beliefs can influence choice of activities, preparation for an activity, effort expended during the adoption of OSS, as well as thought patterns and emotional reactions towards this adoption.

4.2.4 Motivation to use OSS (Intention)

This construct represents the general disposition to use OSS. It captures the motivational factors that influence the adoption of OSS. They are indications of how much effort the company is willing to exert to implement an OSS ecosystem. The stronger the motivation, the more likely they will succeed.

The performance of the process of adopting OSS depends at least to some degree on non-motivational factors such as availability of required opportunities and resources (e.g., time, money, skills, cooperation of others, lawyers with OSS knowledge, etc.). If an organization has the required opportunities, resources, and motivation to perform the OSS implementation, it will have a larger probability of succeeding at adopting OSS.

4.2.5 Role Models, other companies successfully adopting FOSS (Environmental Factors)

According to Ajzen et al. (2010), environmental factors can influence any barriers or enablers in an organization to adopt OSS. The role model of the individuals who would like to copy what executives of other companies are doing, is considered an environmental factor.

Another environmental factor can be whether the software originates from a country or organization that is familiar to the company, and can influence perception about the OSS (For example, Ruby in Japan, Threema in Switzerland)

4.2.6 Adoption of OSS (Behavior of interest)

This construct represents the task of adopting a complete OSS ecosystem and constitutes the desired end state of the model.

Whether an organization achieves the adoption of an OSS ecosystem depends jointly on motivation and ability. Furthermore, motivation would be expected to influence performance to the extent that the company has self-efficacy, and performance should increase with self-efficacy because the organization is motivated to try and adopt OSS.

According to the theory of planned behavior, self-efficacy of OSS adoption, together with motivation to use OSS can be used directly to predict the success in adopting OSS.

Two reasons can be offered for this hypothesis:

1. Holding motivation constant, the effort expended to bring the adoption of OSS to a successful conclusion is likely to increase with self-efficacy of OSS. For instance, if a company is confident that it can implement OSS, it is more likely to persevere until it succeeds.
2. Holding effort constant, self-efficacy (how much the company believes it will be able to successfully adopt OSS) can be a measure of actual efficacy (how successful they will be) as it impacts motivation. However, it should be noted that self-efficacy may not be particularly realistic when a company has relatively

little information about adopting OSS. This could be described as an instance of the so-called Dunning-Kruger effect (Kruger and Dunning 1999), in which a company might be overconfident about its abilities due to lack of knowledge regarding the complexity of the task ahead. Self-efficacy might also be impacted when requirements or available resources have changed or when new and unfamiliar elements have entered the situation.

4.3 Resource-Based View Theory

I use the Resource-Based View theory (RBV) to explain, identify and define how FOSS can be seen as an Information System resource and their relationship with competitive advantage in an SME. Furthermore, RBV theory complements the proposed framework on the understanding of the drivers or barriers that hinder or drive the adoption of FOSS into an SME. The resulting model is shown in Figure 4.

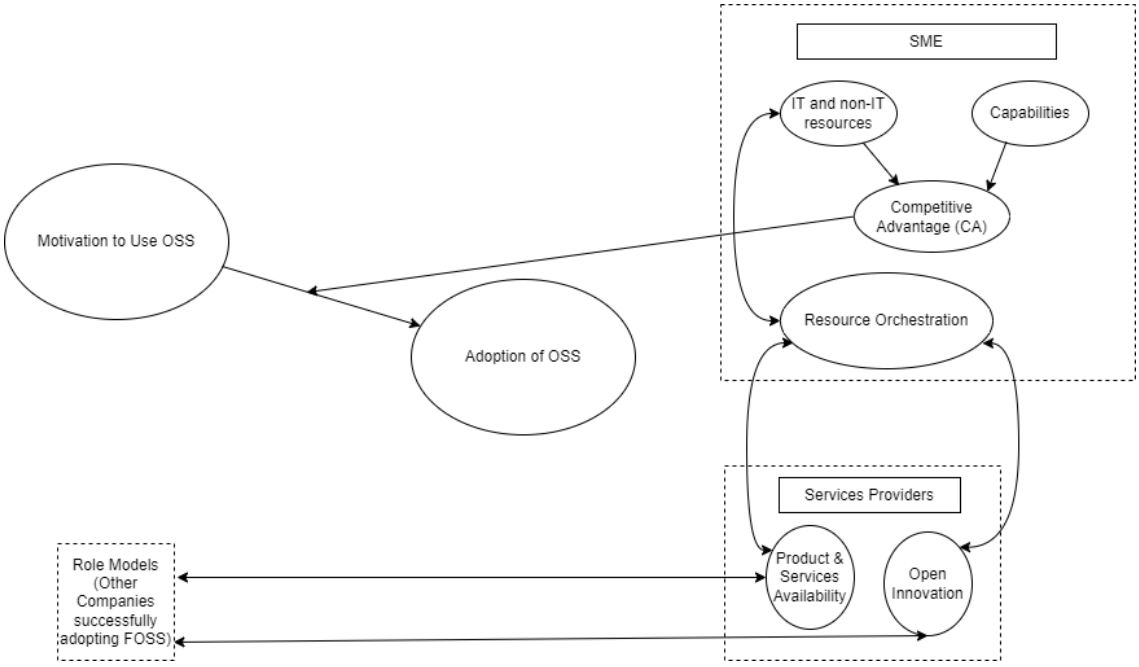


Figure 4: Resource-Based View theory adapted for SME adoption of FOSS.

According to Dwivedi et al. (2012), Resource-based View (RBV) theory has been talked through in strategic management and Information Systems (IS). The authors state that primarily, the impact of resources on competitive advantage (CA) and the identification of critical resources in an organization has been discussed in previous studies. However, the orchestration of resources appears to influence the impact of

resources on CA and other organizational factors such as corporative environmental performance, profitability, and strategic alliance. Therefore, the authors propose a new concept on the IS literature called “resource orchestration”, to respond to the internal interaction between IT resources with non-IT resources in an organization and the process of IT resource interaction with other resources within the company which they called “resource impressionability” (Dwivedi et al. 2012, p. 110).

An SME can be seen as a union of resources and capabilities enabling the company to continue its life (Dwivedi et al. 2012). The RBV theory asserts that an SME (or any organization for that matter) owns clusters of resources (IT and non-IT resources) that guide the company to augment its competitive advantage decided by the characteristics of its resources.

Birkinshaw et al. (2009) put forward the causality between resources and competitive advantage. The authors stated that by having distinctive and peculiar resources that may be “durable, rare, appropriate, non-substitutable, immobile or imperfectly mobile, difficult for others to imitate and have value in the firm’s environment and marketplace” (Birkinshaw and Goddard 2009, p. 81) can generate and sustain competitive advantage. This statement applies also to SMEs and can explained whether its IT resources might adopt FOSS.

This study will focus on Information Technology resources on an SME. There are three key IT assets (Ross et al. 1996): A highly competent IT human resource, a reusable technology base, and a strong partnering relationship between IT and business management.

4.3.1 IT Human resources

When an SME has competent staff who knows how to solve business problems and find business solutions in the Information Technology area, then the company has a valuable IT human asset.

If an SME does not have IT human resources, it is possible to outsource the technical work through an experienced service provider who can deliver the system faster (Ross et al. 1996).

The perceived availability of open-source support providers has grown in Switzerland. This growth has made 67% of people in companies believe that the introduction of open-source software and its maintenance and further development is possible (Open Source Studie Schweiz 2021).

4.3.2 IT resources

The technology asset consists of shareable technical platforms and databases (Ross et al. 1996). They can be divided into tangible resources such as hardware and intangible resources such as software, databases, etc.

4.3.3 Elements of Competitive Advantage

According to Birkinshaw (2009) and Dwivedi (2012) (Birkinshaw and Goddard 2009; Dwivedi et al. 2012), the three attributes of IT resources that help an organization create or attain CA are value, rarity and appropriability.

For this study, these three attributes are described as follows:

Value

In Section 4.2.2, the value that an individual perceives in the adoption of FOSS was described through his expectations of efficacy & efficiency of the FOSS software. Thus, the implementation of FOSS in a company could be seen as a resource that will offer value, if the SME strategy adopts FOSS to improve its efficiency and effectiveness. According to the Swiss Open Source Study (2021), 97% of the companies and public authorities use open-source software. They see interoperability by design as the most important reason to implement FOSS. Digital Sovereignty is among the first five reasons of implementing open source along with the high security that FOSS offers to its users.

Rarity

Although FOSS software is available to an enormous community, it also offers the possibility to have access to the source code. Therefore, the SME could potentially use this feature to be creative and based on FOSS, develop new features on the software that would help to differentiate its services, products, or both. The Swiss Open Source Study (2021) states that business innovation is one of the reasons of using open-source software, with 76.9% of the responders that see how quickly open-source software can help without complicated procedures to create business innovation. Interestingly, they also assert that their internal IT departments can often use open-source tools and components without external help. However, one could question whether this is also valid for SMEs: Would this apply to a very small company that has no IT department? Can a medium company afford an expert to quickly solve open-source technical requirements instead of calling a proprietary software expert?

Open-source support providers have grown in Switzerland (Open Source Studie Schweiz 2021). This is important because support for the introduction of open-source software and its maintenance and further development is a pre-requisite for the adoption of FOSS solutions in a business environment, and as a result an SME can create new strategies to possess value and rarity by implementing and developing FOSS solutions and increase its competitive position.

Appropriability

When the SME capability can appropriate the returns resulting from its competitive position in possessing value and rarity, then it is safe to consider that the SME has competitive advantage. This has been the case of the very few Swiss companies that have used open-source software in the Swiss financial field (Swissdec 2021). Based on FOSS, they have modified the software to comply with the Swiss financial regulations and sell it to other Swiss companies.

However, according to Galliers et al. (2020) a survey conducted to understand the strategic planning behavior in organizations, suggested that implementation of strategies was a common concern, because typically resources are not made available,

management is hesitant, technological constraints arise or organizational resistance emerges.

4.3.4 Elements of Sustainable Competitive Advantage

After the company is successful in establishing competitive advantage due to its strategic use of IT resources, the company needs to be able to make this advantage durable. Barney (1991) stated that to gain a sustainable competitive advantage in the RBV theory, it is necessary to consider that heterogeneity of resources and imperfection of resource mobility could lead to the creation of competitive advantage in a company. Furthermore, Wade et al. (2004) stated that to create a durable competitive advantage an IT resource should have low imitability, substitutability and mobility. As stated by Dwivedi et al. (2012) "these attributes could be expressed in terms of what is desirable: Non-imitability, non-substitutability, and immobility" (Dwivedi et al. 2012, p.156).

Non-imitability

One of the bases for a sustainable competitive advantage is non-imitability. This attribute of the resource makes it difficult for other companies to imitate. Even though competitive advantage no longer comes from product differentiation (Gupta 2018), FOSS offers both the possibility to get a deep knowledge of what customers need, and the possibility to modify these products to fulfill the necessities in a specific region or country.

Nonetheless, this attribute is not seen as important for the Swiss companies that were asked in the Swiss Open Source Study (2021). The study goes further by implying that this argument is perceived as more theoretical, because regular open-source users do not have the technical skills or the financial resources to implement a change quickly. So, this can be clearly seen as a barrier to the application of free and open software in a Swiss SME.

Non-substitutability

According to Dwivedi et al. (2012) “when an organization is in possession of a rare and inimitable resource, competitors may seek to match up by acquiring substitute resources” (Dwivedi et al. 2012, p.156).

This attribute is per definition what manufacturers do to achieve vendor lock-in. They facilitate the integration of solutions from their own ecosystem and consciously and intentionally make it difficult to integrate open-source substitute solutions. By doing this, manufacturers make sure that their software is non-substitutable and position their companies in a competitively superior position that cannot be easily matched by open-source software.

In a scenario where open-source solutions would be substitute resources of proprietary software, this would mean that substitutability is an attribute of open source that might limit the companies’ ability to sustain their competitive advantage through vendor lock-in. Therefore, the most important obstacle to the adoption of open-source software is the vendor lock-in with existing proprietary systems.

Immobility

“Immobile or imperfectly mobile resources make it difficult for competitors to attain instant CA by attracting resources away from rivals, purchasing them like commodities or even mergers and acquisitions with companies possessing strategically important resources” (Dwivedi et al. 2012, p.157).

As an example of immobility, a Swiss IT-oriented SME might possess an innovative important software based on FOSS, and it is more likely to be acquired by a large company that wishes to acquire its competitive advantage. Nevertheless, this is not likely to be done by non-IT SME companies, which could be a barrier rather than a driver to adopt open-source solutions. However, if the non-IT SME can outsource its IT resources with an OSS service provider that can adjust its products or services to the needs of the SME, then this resource can create value for the company.

4.3.5 Information System Resources and Capabilities

According to Ross et al. (1996) a relationship between IT and business management, where both share risk and responsibility for the decisions made to successful create and sustain IT processes that help the company to achieve its goals, is considered an IT asset and a capability. However, Silva et al. (2018) found that in an SME, informal meetings between Business and IT senior managers are commonly practiced. These meetings are normally flat and informal. The authors go further to describe how these informal practices in an SME lead to a poor and ineffective IT resource and capability management, which is a barrier of adopting innovative solutions such as open source.

4.3.6 Resource Orchestration

Taher M. (Dwivedi et al. 2012, p. 160) added a new concept, *resource orchestration.* This term refers to the result of the combination of all resources that a company possesses. Also, it explains how a resource is associated among others in an organization, and how IT and non-IT resources in a company interact. This construct is used in this study because the orchestration of resources guides the SME to execute its objectives efficiently and effectively.

4.4 Technology-Organization-Environment (TOE) Framework Oriented to the Adoption of FOSS.

The TOE framework comes in handy for this study to unify the adoption of FOSS in an SME. First, it has been already explained based on the Theory of Planned Behavior, how the behavior of individuals that belong to a company can influence the SME to adopt FOSS. Second, it has been described from the point of view of an SME using the Resource-Based View, how its strategic management and Information Systems could be integrated while adopting FOSS as an IT resource. Finally, through establishing relationships among constructs that belong to each theory, the reference model FOSSA is completed.

4.5 Relationships in the FOSSA reference model

An individual perception about the ease or difficulty of adopting OSS (self-efficacy of OSS) can often be used as a substitute for a measure of actual control of integrating

OSS as an IT resource. This perception of one or more individuals from the SME can be influenced by the perceived benefits of adoption of OSS, and the perceived expectations of the software efficacy & efficiency.

If business and IT management of the company can agree that in the foreseeable future an implementation of open-source ecosystem can be integrated along with the rest of its resources and capabilities to create Competitive advantage (CA) for the SME, then the motivation to use FOSS will influence its adoption. However, for this effort to succeed, it is necessary that each resource acts as one instrument. The successfulness of the project relies more on the overall outcome of various resources (resource orchestration) including the adoption of a FOSS ecosystem.

Furthermore, the ability to adopt and the continuity to implement and maintain a FOSS ecosystem as an instrument in the SME might or might not be part of the company. This ability can either be outsourced to established services providers according to the product & services availability on the market or can be crowdsourced. Gupta (2018) describes “leveraging the expertise and insights of both users and experts outside the company, often called open innovation or crowdsourcing, has been on the rise in recent years” (Gupta 2018, p.85). Large companies are adopting open innovation to make use of the knowledge of outside experts, suppliers, and lead users in addition to their own internal network.

The success of the adoption of an open-source ecosystem from a role model (other companies) that either outsourced or crowdsourced this IT resource, will influence the perceived expectations of software efficacy and efficiency in the SME. The rational to this statement is that when an individual of the SME hears that it was possible for other SME to adopt a FOSS ecosystem and hears all the benefits that this strategy brought to their company, the individual starts to believe the efficiency and efficacy of FOSS. The opposite is also true, if the role models have not succeeded, then the perceived negative standards would lead the SME to not to adopt FOSS.

The FOSSA reference model aims to explain the organizational effectiveness and efficiency on the adoption of OSS. To achieve this goal, the reference model as a theoretical study must be tested and compared to relevant empirical data, which is described in Chapter 5.

5 Results

The proposed FOSSA theoretical model (see Chapter 4) aims to explain the drivers that influence positively to the adoption of OSS and the barriers that hinder this process in an SME. One of the IT research tasks in Information System is to make sure that a theoretical model develops an understanding of how and why an IT system, in this case OSS adoption in an SME, works or does not work. Furthermore, human knowledge helps to create well-informed actions that can be used to successfully gain goal-achieving ends (March and Smith 1995).

Therefore, a qualitative survey research methodology was implemented to reach more understanding of the phenomenon of OSS adoption in an SME. The collection of different perspectives about this topic brought additional knowledge to tie together the proposed reference model and the tasks faced by practitioners to obtain a much-enriched answer to the research question of this thesis: *What are the barriers and drivers that hinder the adoption of FOSS in Swiss SMEs?*

5.1 Data Collection

A series of eight problem-centered interviews were conducted from 26 of April to the 16 of May 2022. The process model of the problem-centered interview is described in Figure 5 and was based on the model proposed by Mayring (1995).

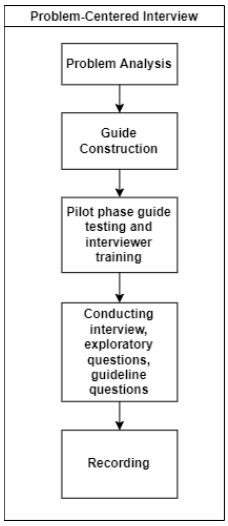


Figure 5: Problem-Centered Interview process based on (Mayring 1995)

The information obtained from these interviews aims to attest or contradict the validity of the theoretical model proposed and to bring new factors that might lead to understand the adoption or non-adoption of OSS in SMEs.

The interviewees were drawn by convenience sampling and included CEOs of SMEs, OSS Service Providers, an IT Consultant, a Cloud Expert, and a Research Expert. Two sets of interviews were prepared, one for Service Providers and one for OSS Users (see Appendix A). Even though service providers can be OSS users, they could also provide more information about how their clients perceive the adoption of OSS and how they cover their customers' needs.

5.1.1 Problem-Centered Interview

The main reason to hold problem-centered interviews was to capture the subjective perspective of the respondents. This is important because in the FOSSA theoretical model, a perception of an individual is a central point.

An open, semi-structured questioning was conducted to every participant. The goal was to allow them to speak as freely as possible to come to an open conversation. Nevertheless, the interview was centered on the adoption of OSS in SMEs through an interview guide structured (see Appendix A) according to the interviewee's role (Service Providers or OSS Users). These guidelines are a list of questions that were created beforehand to elicit the subjective experiences of the persons with the adoption of OSS in SMEs during the interview.

Every interview differed in the sense that the guideline could not be followed in the order that the questions were created. For example, a service provider tended to speak more about partners and clients, going off on a tangent during the discussion. However, thanks to the specific guideline made for service providers, I could bring the interviewee back to the topic focus.

On the other hand, a second service provider did not have specific information about SMEs as direct clients, but to partners/clients that work with SMEs. Through their

reactions and comments about how the company operates, a new connection to the existing construct in the FOSSA reference model was developed. Large companies have partners that are in direct contact with SMEs, and they are at the same time clients to the large company, so the guideline created for service providers could still be used but thematized within another perspective.

During the interviews, it was important that every question was understood by the respondents, so that they could develop their own context in the interview. By doing this, some spontaneous counter-questions could be created to provide more information and a better fit for each respondent, which in turn added valuable information to this thesis work.

5.2 Data Analysis

The qualitative content analysis was developed inductively out of the data collected. In this sense, a category system was created by doing a pilot test to achieve methodological strength (Mayring 2014) based on two initial interviews with service providers that participated in the study.

The goal of this data analysis is to reduce the material to the point that the essential information is captured (Mayring 1994) and to search for patterns within the data to attest or contradict the validity of the theoretical FOSSA model proposed in this master's thesis work.

Coding was implemented as an exploratory problem-solving technique with no algorithms or formulas followed (Saldaña 2013). Coding helped in labeling and linking the data obtained from the respondents with constructs, some already identified in the FOSSA model and others to be discovered. The coding on this thesis aimed to categorize data. These categories were later grouped, regrouped and relinked to combine clarification and meaning according to Saldaña (2013).

5.2.1 Phenomenological Analysis

To better understand the essence of the information a phenomenological analysis was first used (Mayring 1994). The basic idea of a phenomenological analysis is that the perspective of the individual should be considered, their subjective structures of meaning, their intentions. The aim of this analysis is not to stop at the surface of appearances, but to get to the essence of things (Mayring 1994).

In this sense, the phenomenon of OSS adoption in SME is compared in different contexts. First in the context of the individual as how important the Free and Open-Source Movement is for the person, their benefits, perceptions of self-efficacy of OSS and perception of OSS efficacy and efficiency. The second context is that of the company where the person works. This allows observing both the personal and the business-oriented reaction to the same topic and to identify any invariants among the different opinions, considering the potential consequences of their business decisions.

5.2.2 Qualitative Content Analysis

To analyze interview transcripts, the technique of content-analytical summarization was used (Mayring 2014). This analysis helped this thesis work to examine systematically the information, and to discover the key concepts mentioned in the interviews with the goal of mapping them to the constructs of the FOSSA model. By using this technique, inductive category development was done going line-by-line within the interview transcripts. The determination of the creation of categories is fully explained in Section 5.2.3.

Figure 6 describes the process model of inductive category formation used in this thesis according to Mayring (2014).

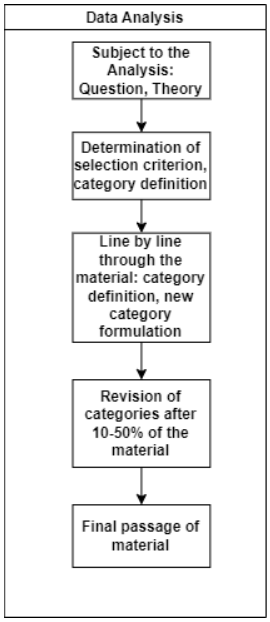


Figure 6: Process model of inductive category formation based on (Mayring 2014)

When there was approximately 50% of the data, the collected category system was revised. There was a need for a check as some overlaps were found. The degree of abstraction was also checked to fit the research question. Some of the categories were merged to a more abstracted concept and regrouped to map them to the constructs of the FOSSA model. A final general revision was done to ensure that constructs and quotations were in place.

5.2.3 Coding Rules

A pilot test was conducted using the first two interviews where the first constructs were created by using:

- Concepts that involved sentences given by the respondent.
- Keywords from sentences given by the respondent.
- Using the exact sentence given by the respondent, as the main concept involves the whole sentence, for example: “an SME should focus to sell a product and not on technology”, and later this category was regrouped in a more abstract category named “SME Business Strategies - Goals”.

As the interviews continued, their transcripts were created by Otter (otter.ai 2022) and downloaded in Atlas.ti (Atlas 2022). To help code the information, some coding rules were applied to limit the decision to assign categories. The following rules were applied:

- First, the text quotes in the material, in which the category was addressed, were identified. The new categories were coded "in vivo" and the already created categories were selected if the text quote matched the concept of the category.
- To map the categories from the constructs of the FOSSA model, every category was assigned a specific color based on the phenomenological analysis (to cover both the individual context and company related opinion context), as shown in Table 2.

Table 2: Color description according to phenomenological analysis (Mayring 1994).

Color	Context of Opinions
Yellow	Positive statements about OSS adoption
Purple	Negative statements about OSS adoption
Gray	Statements about a hybrid adoption of OSS and proprietary software
Brown	Statements related about the company (business strategies, IT related statements, marketplace, etc.)
Green	Statements related to stakeholders (developers, service providers, decisions makers, partners, etc.)
Turquoise	Statements related to sectors not open to adopt OSS
Pink	Statements related to open innovation
Light Blue	Statements related to global and large companies
Orange	Statements related to concepts of OSS
Red	Statements related to concepts of proprietary software
Dark Blue	Statements related to the use of cloud platforms

The category assignment was an iterative process with various passes over the text as new categories were identified. As a second step, categories were grouped according to

the constructs of the FOSSA model and new groups that did not match the constructs of the FOSSA model, but that were important were created.

After the final revision, a total of 172 categories were identified and grouped into 17 constructs where 12 correspond to the constructs in the FOSSA model and 5 were new constructs created from the information given by the respondents. All identified constructs are described in Table 3.

Table 3: New Constructs identified, and FOSSA model constructs grouped

Construct	Source
Perceived Self-Efficacy of OSS	FOSSA model
Perceived Efficacy & Efficiency of OSS	FOSSA model
Perceived Benefits of Adoption of OSS	FOSSA model
Motivation to use OSS	FOSSA model
IT and non-IT resources	FOSSA model
Capabilities	FOSSA model
Competitive Advantage	FOSSA model
Resource Orchestration	FOSSA model
Product & Services Availability	FOSSA model
Open Innovation	FOSSA model
Role Models	FOSSA model
SME Business Strategies - Goals	Identified from interviews
IT Change Management	Identified from interviews
Company Neutral Position about Software	Identified from interviews
Large Company Strategies	Identified from interviews
SME Business Needs with Proprietary Software	Identified from interviews

5.2.4 Data Results

After the qualitative content analysis was done, several results were ready to be shown. One part of the research question asks for the barriers that hinder the adoption of OSS in an SME. According to the conducted interviews, the top-level categories of barriers that hinder the adoption of OSS by companies are shown in Figure 7.

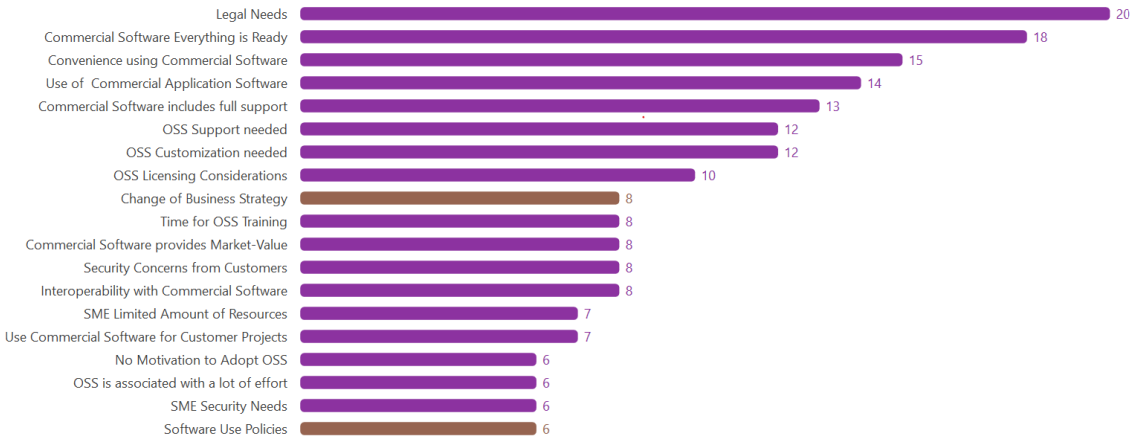


Figure 7: Top Ten Scores of Barriers (non-adoption) of OSS by companies

A second part of the research question asks for the drivers that influence the adoption of OSS in an SME. The top-level categories of drivers for OSS adoption are shown in Figure 8.

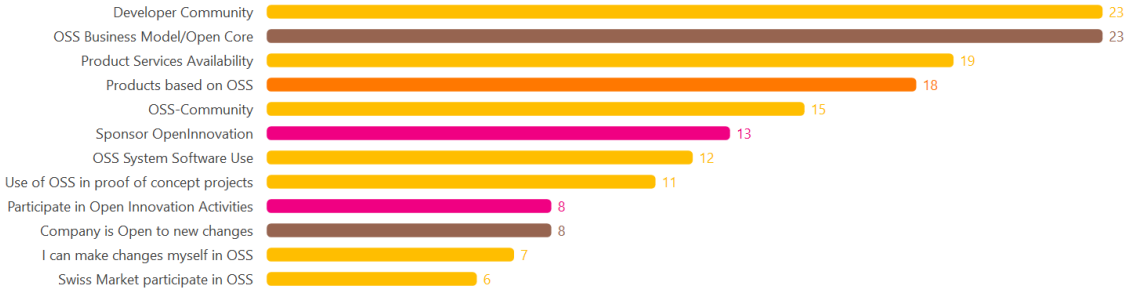


Figure 8: Top Ten Scores of Drivers (adoption) of OSS by companies

The same respondents were asked, in an individual context, their opinion of the drivers that influence and barriers that hinder the adoption of OSS, so that their roles in the company were not important. The results are shown in Figure 9 and Figure 10.

Figure 9: Top Ten Scores of Barriers of OSS adoption by individuals

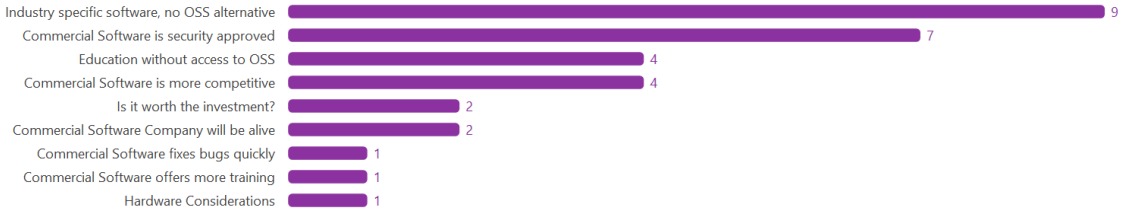


Figure 9: Top Ten Scores of Barriers of OSS adoption by individuals

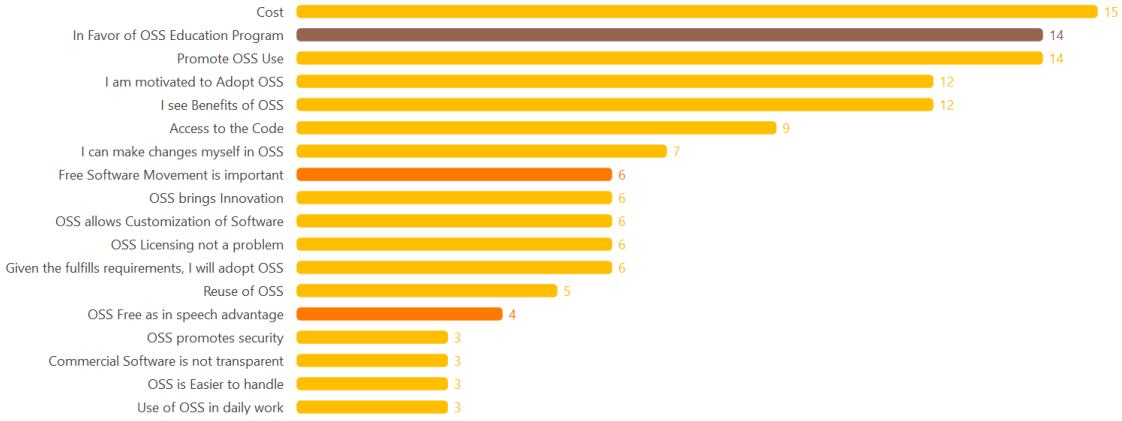


Figure 10: Top Ten Scores of Drivers for the adoption of OSS by individuals

As the theoretical FOSSA model was proposed in this thesis work, part of the contribution is to validate or invalidate it based on the respondent's information. For this purpose, the groups of categories in Table 3 are shown in the following series of figures (Figure 11 to Figure 26).

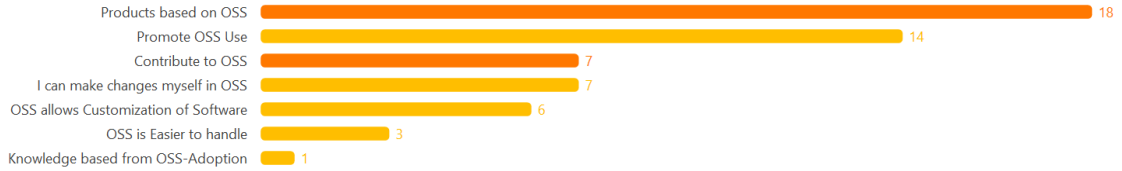


Figure 11: Perceived Self Efficacy of OSS

Figure 12: Perceived Efficiency & Efficacy of OSS

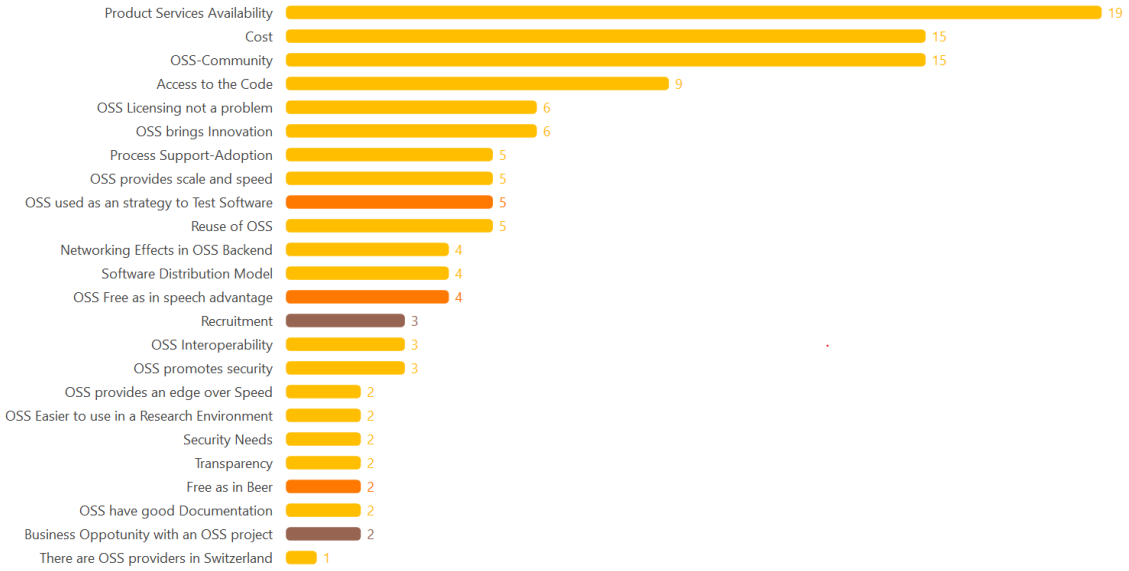


Figure 13: Perceived Benefits of OSS

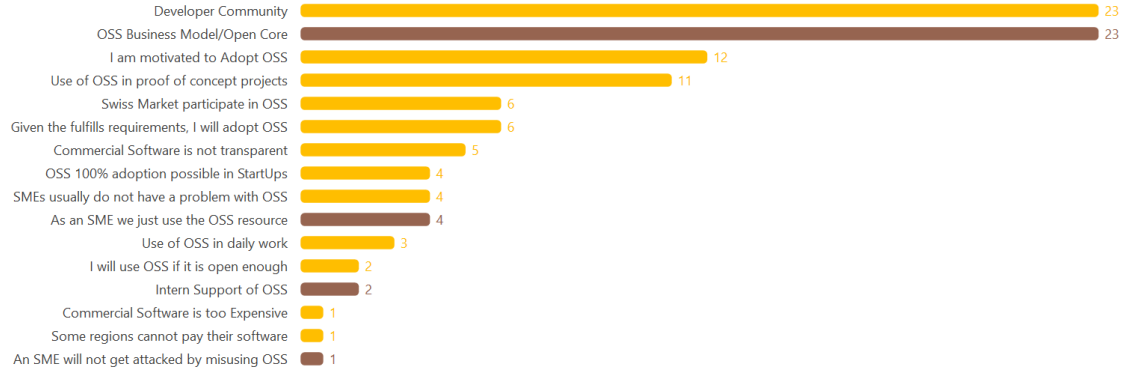


Figure 14: Motivation to adopt OSS

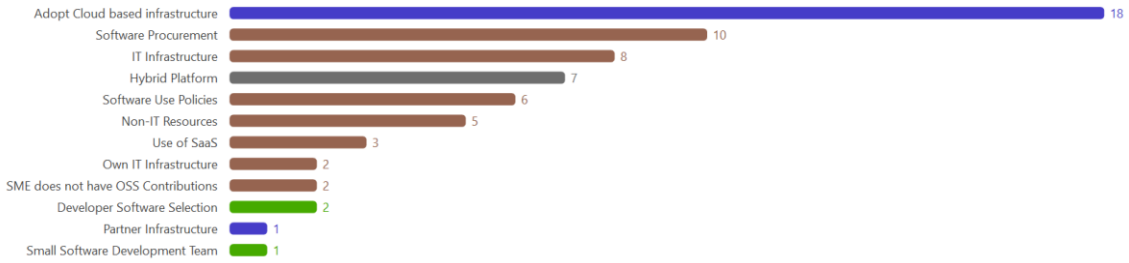


Figure 15: IT and non-IT Resources

Figure 16: Capabilities

Figure 17: Competitive Advantage

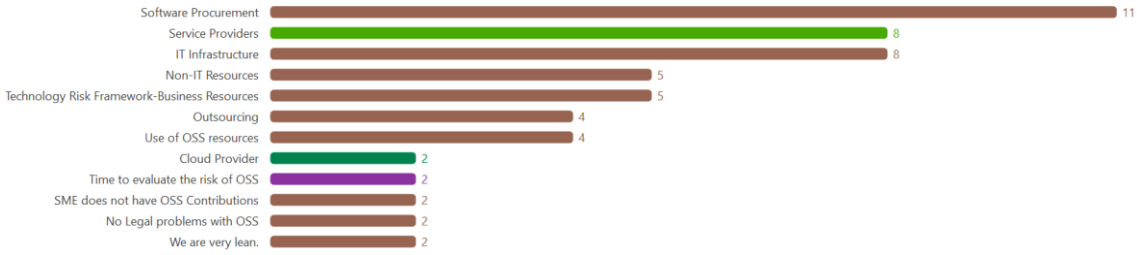


Figure 18: Resource Orchestration

Figure 19: Product and Services Availability

Figure 20:Open Innovation

Figure 21:Role Models

Figure 22: IT Management

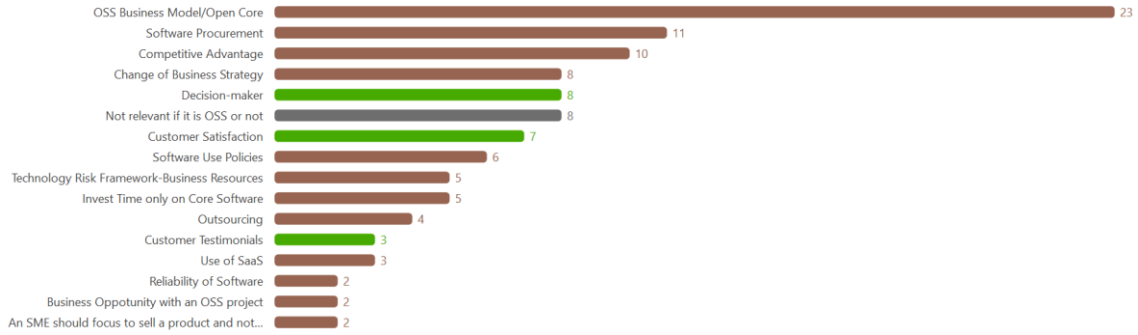


Figure 23: SME Business Strategies – Goals

Figure 24: Company Neutral Opinion about Software

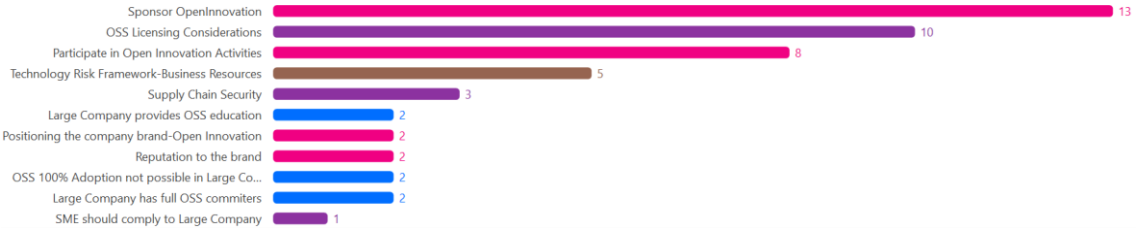


Figure 25: Large Company Strategies

Figure 26: SME Needs with Proprietary Software

These graphics shown the frequency that each category occurs. The interpretation of the results is discussed in Section 6.

6 Discussion

According to the results in Section 5.2.4, there is a discrepancy between individual context and company related opinion context. It has been observed frequently that the personal beliefs of the interviewee with respect to open source were contradictory with their business decisions since the latter could have consequences beyond their personal preferences.

As an example, “Legal Needs” is the top category that appears as a barrier that hinders the adoption of OSS in SMEs. Interesting is the fact that cloud and research experts did not mention any concept that could match this category as a barrier, hence their absence in the graphic, see Figure 27. However, the interviewees who have a legal impact in their choices of software mentioned that in general, they must comply with the security policies of their external partners/customers, government regulations and even their legal team to make sure that no breach of confidentiality, contracts or rules endangered their relationship with their customers. This is an important constrain that clarifies why most SMEs and organizations in general are reluctant to adopt OSS, because if legal

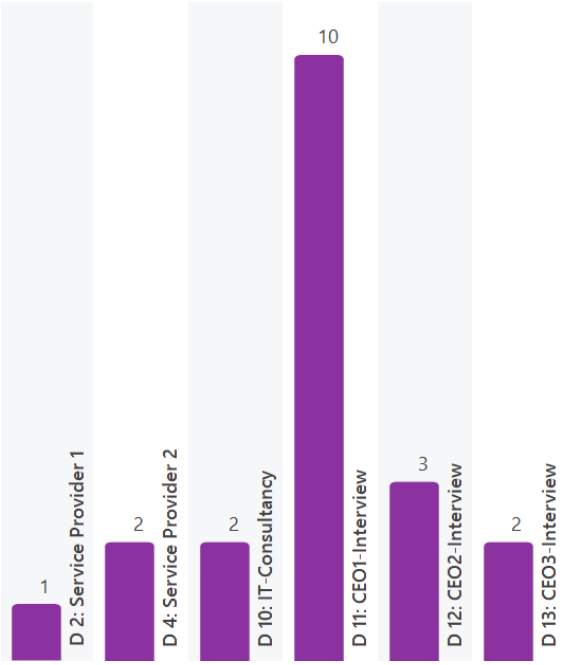


Figure 27: Interviewees that stated legal needs as a main barrier to adopt OSS in their companies

implications of adopting OSS can be seen as a risk, the company must decide how it is going to handle this risk (Königs 2017).

According to the International Organization for Standardization (2018), the possible decisions are:

- do nothing further
- consider options to address the risk
- conduct further analysis to better understand the risk
- maintain existing controls
- reconsider the objectives

As one person who was interviewed stated: "*the risk is playing a big, a bigger role than before, right? So you take more into account, risk aspect and the legal aspect, […], so you consider that more than you did before. Because it's rather, it's more your responsibility."*

The point of stress for an SME is not only to choose a decision from the list, but also to create and implement a plan of business strategies to continue developing linkages with their customers, and to become or continue being considered a technology-driven SME (Asheim et al. 2003). Nevertheless, there are other SMEs where technology does not play an important role or that are technology indifferent. These SMEs might tend to decide for the first decision on the list: do nothing further.

However, the use of OSS is not always considered as a risk when it comes to other SMEs' needs. For example, several interviewees stated, that they use OSS in their daily work whenever they need a proof-of-concept implementation, see Figure 28, where there are no legal consequences on their software choices. As one respondent says "*so security is, is an obstacle that all new software that is being used by an employee needs to be approved by, they [customers] will not only look at the technology but they [customers] will look at the company behind the service. They [customers] will look at agreement, look at what kind of data is being harvested, etc."*

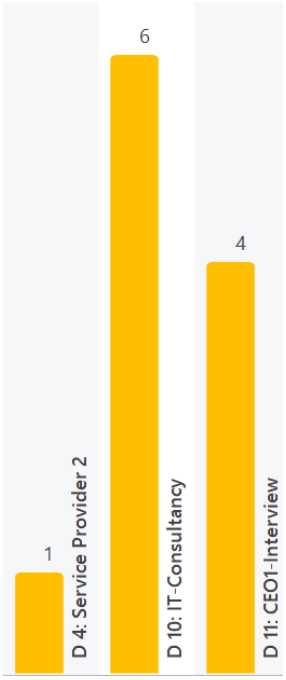


Figure 28: Interviewees that use OSS to implement proof-of-concept projects

This last statement opens the discussion that even though an open-source project can be more transparent than a closed software, the adoption of one of them relies more on the impact of responsibility because whenever one adopts an open-source project, the responsibility, if a security issue arises, resides with the company; whereas if the SME adopts a closed software, they can transfer this responsibility and any legal issues to the software vendor. This bring us to section 3.1.3, where Hoepman et al. (2007) (Hoepman and Jacobs 2007) stated that what determines a system "secure enough" is the risk associated with the use of the system.

Looking down in Figure 7, the second and third categories are more in the field of perceived ease of use and usefulness which are two popular constructs in models such as the Technology Acceptance Model (TAM). The category "Commercial software everything is ready" goes in the direction of the internal interoperability of software by all staff in the company. The category "Convenience using commercial software" goes in the direction of external interoperability with partners and customer interaction.

The fourth category of Figure 7 ("Use of Commercial Application Software") shows us that commercial software is used in the application software area of the company. This

statement was already mentioned in section 3.2.3 where Huysmans et al. (2008) (Huysmans et al. 2008) found that in the organizational context, the lack of internal knowledge of OSS projects such as Open-Office is a barrier to reject the adoption of OSS.

Furthermore, based on the interviews, a second barrier that was found states that people are used to the software. In Switzerland, "apply editing and design of texts, tables, presentations, diagrams, images, sounds, videos and algorithms." are taught in the second half of the 2. cycle (from third grade to sixth grade of primary school) (Erziehungsdirektion des Kantons Zürich). According to Hackbartha et al. (2001) "users perceive a system easier to use as they gain more knowledge and confidence through direct experience in using the system" (Hackbartha et al. 2001, p. 221).

Thus, the perceived ease of use determines the individual intention to use a system. If a person starts to use commercial software in primary school, by the time they start working in a company, their system experience with commercial software is significant to influence their decisions towards commercial software. As one of the respondents stated: "*I think the other major blocker is just familiarity of people, […] for example, like in the case of Office applications that everyone is already used to the commercial versions, and there is no strong motivation to switch."*

On the other hand, "OSS System Software Use" occupies one of the top ten spots in the drivers than influence the adoption of OSS in a company, as seen in Figure 8. This result is not surprising as many server system platforms used by companies have a version of Linux installed. Even cloud service providers have products that are based in open-source components. Acording to one responder "*many of the services that the latest builds are based on open-source components such as the Linux kernel or other libraries such as Jupyter notebook or Python libraries or many other So, yes, I think the answer is yes, we use open-source and thoroughly for building our [cloud] services and also for other purposes for internal applications or for some, some office applications are also based on open-source software".* Figure 29 shows the frequencies that this category was mentioned by the respondents from interviews.

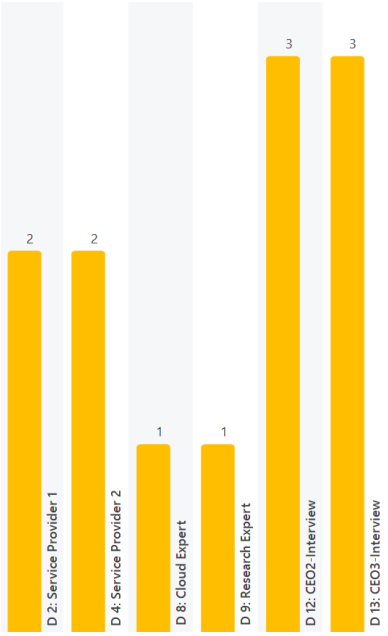


Figure 29: OSS System Software Use in Companies

The fifth category “Commercial software includes full support" in Figure 7 goes in the direction where commercial software offers all support needed to consume the software, with just one contract. On the other hand, the sixth category, “OSS support needed” goes in the opposite direction as the previous category. Since most SMEs need to focus on their core product or service, instead of the time and effort that they would use to get the OSS support needed, they might as well just sign a contract with a software vendor who gives them full support.

The last statement is also true for the next category of barriers that hinder the adoption of OSS in SMEs, “OSS customization needed”. As one of the respondents who is an OSS service provider stated “*I'm saying this because that's how we came up with all the, with the solutions for SMEs, which is a bit out of the box, that you're ready to use various standards that feels all setup 90% of SME’s needs. That's how I mean, from our perspective, this is how the SMEs that's what they'd like with [Service-Provider], so maybe the enterprise version, which is the paid one, because everything is ready and settled and ready to use at a reasonable, reasonable price. […] But again, just making sure that you don't have too much customization because otherwise, it becomes maybe more expensive to go on the operating sales, with lots of customization, rather than*

going for an enterprise, I mean, paid version, but which is ready to use this. So it's just to find the right balance."

OSS licensing considerations is a main barrier for the adoption of OSS, especially for large companies. Respondents that work for large companies, all stated that the need of the review for open-source licenses is paramount. Some OSS solutions impose restrictions on the software use, for example some of them require that any modification that someone does to the software needs to be also published as open source. One respondent said "*if you make any modifications to the source codes in your own use of the software, then your modifications also need to be open-source. Therefore, it's infectious […] in the sense that it infects its open-source quality to other software that uses it. So, of course, for many companies, this is troublesome, because if they are using open-source software internally for developing their own products or services, they don't necessarily want to release this code that they are developing to the world. So, I think for my customers, the most important aspect has been license analysis and license selection. So, they tend to devote the time to inspect the licenses that they need to use, to determine if they are allowed, if it is possible for them to use them without any damaging effects to their own development, because then and only then, they get the green light to proceed with the usage of the software […] also [Large company] in this sense is very careful about specifying which licenses it is possible to use and which not both internally in employee systems, but also in their own products. So during the development of new services or new products, the types of open-source licenses that can be used is very carefully controlled precisely to avoid these undesirable effects."* As an SME, these conditions that a large company imposes are to be followed to be able to do business with them, because an SME may only be part of the supply chain for large companies.

Security concerns from customers are also a main topic for people from an organizational context point of view; even though literature review shows (Section 3.1.3) that there is no clear evidence that OSS is less secure than proprietary software, it seems that not everyone is convinced. The risks that a company has to accept for every decision in software procurement is high, and the consequences and damage are ever greater, if an entire customer's system infrastructure fails as a result of a ransomware

attack or sensitive information leaks out. The company responsible for the security breach in the supply chain most probably would need to file for bankrupcy, specially if it is an SME.

Therefore, it is no surprise that a decision maker chooses to comply with its customers by consuming proprietary software even if that means to lose control over certain aspects of the software that they use for their daily operations. This perspective changes when the individual context is in place. As we can see in Figure 10, “I am motivated to Adopt OSS”, “I see Benefits of OSS”, “Access to the Code” and “I can make changes myself in OSS” are from the fourth to seventh place. Having motivation, access to the code and knowledge of OSS is, for the developer community, enough to propose changes and influence the direction of the software. As already stated, the free and open source software movement believes in the opportunity that everyone can use it, modify it and redistribute it. As Riehle (2007) (Riehle 2007) stated, the consumption of open source software in companies should be a business strategy because the adoption of OSS:

- lets companies grow their user communities, and
- builds an ecosystem around their products and services.

The last statement would also let companies have complete control over their IT systems and avoid third party rules and regulations imposed on their software systems. Thus, the company can maintain its digital sovereignty and safety.

Safety comes along with digital sovereignty because it frees the company from being influenced by the decisions of other companies or governments.

Nonetheless, decision makers need most of the time to give up their digital sovereignty and safety with respect to the software they use, to gain time to focus on the core products and services of the company. As one respondent stated: “*I think businesses need to focus on selling their stuff, not dealing with infrastructure. So you cannot do the open-source only, then you will need to hire people to manage that on your behalf like have an IT department or something like that. And small SMEs should not have an IT department they should have all their efforts focused on winning in the marketplace, not*

the infrastructure." A second strategic decision that arised with the information provided by the respondents is that the decision to consume commercial sotware is mostly made to transfer risks associated with the software. Therefore, an important construct was created and needs to be in the proposed theoretical model FOSSA. The concept was abstracted and named as "SME Business Strategies – Goals".

Along with the construct "SME Business Strategies – Goals", it is important to consider a second new construct "IT Change Management". This construct refers to the concept of leading IT change. It is important to notice that change processes consist of different phases; thus time is an important consideration and crucial mistakes can have disastrous effects such as slowing down the process and negating prior gains (Kotter, John, P. 1995). Again, time is a limited resource in an SME and adverse effects are too risky for a decision maker to authorize it. Plus, the human factor should also be taken into consideration. One of the respondents said: "*And I will say this, like change management with technical teams is not the easiest from my experience"*. Therefore, unless the software is critical for the core business of the company, there is not going to be a sense of urgency, which means that the transformation from commercial software to open source software will be difficult to achieve.

A third new construct: "Company neutral opinion about software" was created to match several ideas that the respondents shared. First, most of the respondents have migrated or are in the process of migrating to a cloud platform. They use SaaS or FaaS services with different software vendors and are no longer worried or interested about infrastructure or installation of software; therefore the choice of open source or proprietary software is irrelevant to them.

The fourth construct "Large Companies strategies" was also created as all of the respondents named at least one idea that matched the concept. This refers to limitations imposed on SMEs by the large companies that they deal with, either as customers or as suppliers. This sustains the statement made by Asheim et al. (2003) where it is stated that "Outsourcing and contracting out by large companies create business opportunities for small firms which can contribute to the development of clustering in which a few

large companies act as final producers, and as customers for their small firm suppliers" (Asheim et al. 2003, p.6).

Finally, the last new construct created from the data is "SME Business needs with proprietary software". It came to my attention that there is commercial software that can not be replaceable and is crucial for core operations for several companies. One of the responders stated "*to be specific, […] Microsoft Visio, a drawing app, where many people draw, network plans, and things like that is one which doesn't exist for the Mac, for example. And that's something which hardly any other tool can read. So there are a few tools around which seem to be so engrained in the industry, or like these kinds of drawings, that it still sets the proprietary standard for the file format. It doesn't work on Mac. So […] you're limited to Windows […] sometimes some people do have to have actually Windows and Visio, just because their environment forces it on them."*

Furthermore, the findings of these new contructs helped to enrich the FOSSA reference model. The updated FOSSA reference model can be seen in Figure 30.

6.1 Future work

The updated FOSSA reference model is a guideline used to clarify the reasons for an SME to adopt or not a complete open-source ecosystem. As future work, a maturity model based on FOSSA would be ideal to guide decision makers in clarifying with more detail insights about areas that need focus for improvement towards adoption of OSS in their SMEs.

Furthermore, a case control study of a selected group of SMEs (for example it can be a group of IT-oriented SMEs) that can be tracked over time to look at the development of their efforts to adopt or not OSS solutions and give results of either success or failure. One group of SMEs (CASE) could be those companies that are willing to adopt OSS solutions, and the second group (CONTROL) might be a group of enterprises that do not make any effort of consuming open-source ecosystems, and that are consuming commercial software. At the end of the study, the results are analyzed retrospectively

and would provide representative information about the process that SMEs need to follow to achieve digital sovereignty, safety, and security.

It should be noted that to increase its reliability, the case control study should be a long-term study with the appropriate support by companies and authorities.

6.2 Conclusions

“When it comes to digitization, it’s the big IT giants, especially from the U.S., that determine what’s possible – and what’s not.” (Tillmann Braun 2022). Indeed, proprietary software makes it easier for companies to focus on their core products or services instead of the technology that they are consuming; however, they are compromising their digital sovereignty as proprietary software makes sure to close its source code, so it is harder to verify it, replace it, or make it interoperable with any third-party software that does not belong to the software vendor. Some vendors go even further as to control their customers data. For example, Microsoft announced that in 2025 it would offer exclusively cloud-based delivery models (Tillmann Braun 2022). This means that all users will be forced to join the Microsoft cloud platform and their data would be stored in a place owned by the U.S.-based company. Nevertheless, responsibility associated with the important task of software procurement falls on decision makers in the companies as well as governments.

Furthermore, many governments have their own barriers to adopt open-source software. For example, the recent article about the military Swiss government “Digital Independence has Limits” (Marco Zwahlen 2018). Several Swiss politicians denied a motion to protect the digital independence of their military systems arguing that the text of the motion was vague, vast and had arguments in opposition with each other. Additionally, they argue that the implementation of a digital independence from the traditional criteria (performance, cost, etc.) might be unfeasible.

In conclusion, to answer the research question of this thesis work:

What are the barriers and drivers that a Swiss SME can have for adopting FOSS?

It is necessary to include human, business, and environmental aspects to provide a complete view of the phenomenon of the adoption of OSS in the complex environment in which Swiss SMEs operate.

The starting point to explain the barriers and drivers that a Swiss SME can have for adopting FOSS is the FOSSA reference model (Figure 30). This proposed model helps to understand which immediate constraints exist in Swiss SMEs that are preventing them from adopting a FOSS ecosystem. These constraints are represented by different constructs defined in the FOSSA model that need to be considered collectively by decision makers from SMEs, large companies, and governments.

The updated FOSSA model (Figure 30) encompasses the behavior of individuals that belong to an SME towards the adoption of open software, the business-related factors that impact the performance of the company in the marketplace, and the environmental elements that have an influence on the company. The FOSSA model offers a structured approach to evaluate individual and organizational reasons and concerns for OSS adoption in a Swiss SME.

Ultimately, decision makers must reach the best conclusions for their own organizations and create digital strategies, so that their security, safety, and digital sovereignty prevail.

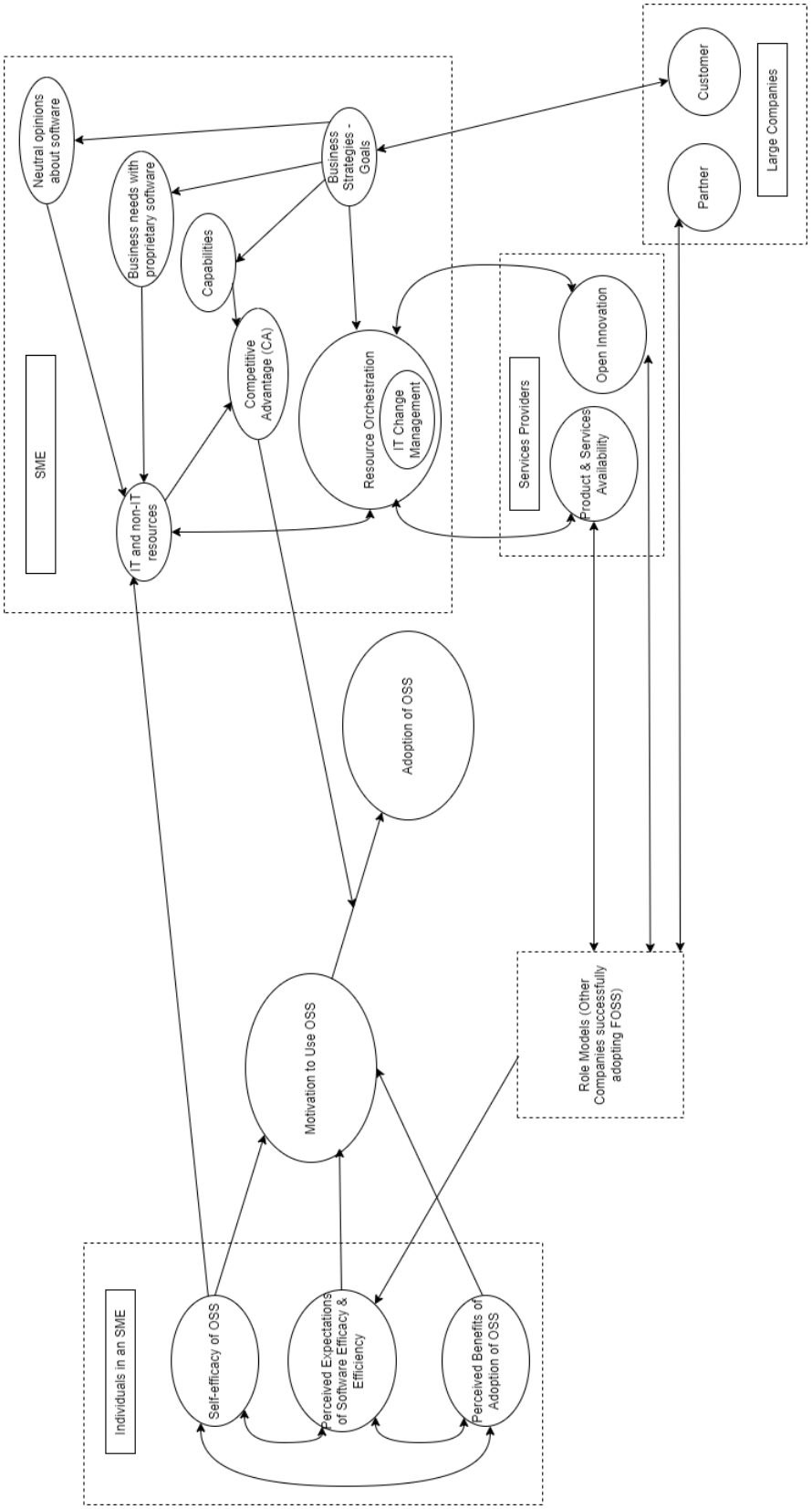


Figure 30: Updated FOSSA Model

References

Ajila, Samuel; Wu, Di (2006): Empirical study of the effects of open source adoption on software development economics. In *The Journal of Systems and Software* (80), pp. 1517–1519.

Ajzen, Icek (1991): The theory of planned behavior. In *Organizational Behavior and Human Decision Processes* 50 (2), pp. 179–211. DOI: 10.1016/0749-5978(91)90020-T.

Ajzen, Icek; Fishbein, Martin (2010): Predicting and Changing Behavior. The Reasoned Action Approach. New York, USA: Psychology Press.

Aksulu, Altay; Wade, Michael (2010): A Comprehensive Review and Synthesis of Open Source Research. In *JAIS* 11 (11), pp. 576–656. DOI: 10.17705/1jais.00245.

Asheim, Bjørn Terje; Arne Isaksen; Claire Nauwelaers; Franz Tödtling (2003): Regional Innovation Policy For Small-Medium Enterprises. Glos GL50 UK: Edward Elgar.

Atlas (2022): atlas.ai. Version 22: Atlas. Available online at atlas.ai.

Balla, David; Maliosz, Markosz; Simon, Csaba (2020): Open Source FaaS Performance Aspects. In Jay Pfaffman (Ed.): Transforming High School classrooms with Free/Open Source Software: it's time for an Open Source Revolution. 2020 43rd International Conference on Telecommunications and Signal Processing (TSP). Milan, Italy, 7/7/2020 - 7/9/2020: IEEE, pp. 358–364.

Barney, J. B. (1991): Firm resources, and sustained competitive advantage.

Behring Center (Ed.) (2021): Apple Macintosh Personal Computer. Natural Museum of American History. Available online at https://americanhistory.si.edu/collections/search/object/nmah_334371.

Benzerga, Shéhérazade (2016): Digital Transformation Patterns. In : Proceedings of the 23rd Conference on Pattern Languages of Programs (PLoP).

Birkinshaw, Julian; Goddard, Jules (2009): What is your management model? In *MITSloan, Management Review* 50 (2), p. 13.

Brooks (1987): No Silver Bullet Essence and Accidents of Software Engineering. In *Computer* 20 (4), pp. 10–19. DOI: 10.1109/MC.1987.1663532.

Chesbrough, Henry; Bogers, Marcel (2014): Explicating Open Innovation. Clarifying an Emerging Paradigm for Understanding Innovation.

Costa, Carlos J. (2013): Proceedings of the Workshop on Open Source and Design of Communication. New York, NY: ACM (ACM Other conferences). Available online at http://dl.acm.org/citation.cfm?id=2503848.

David Silva; Miguel Mira da Silva; Ruben Pereira (2018): Baseline Mechanisms for Enterprise Governance of IT in SMEs.

Detwiler, Bill (2021): Open source developers are unique: Tips for working with and managing community-centric dev teams,. IBM's vice president for open technology and developer advocacy shares his insights into what makes working with open source developers unique and talks about some of the coolest open tech projects IBM is working on. Tech Republic.

Dwivedi, Yogesh K.; Wade, Michael R.; Schneberger, Scott L. (2012): Information Systems Theory. New York, NY: Springer New York (28).

Eidgenössisches Departement des Innern EDI Bundesamt für Statistik BFS (2021): Wie die Technologie die Transformation des Immobiliensektors beschleunigt. In *KMU-Portal für kleine und mittlere Unternehmen*, checked on 11/26/2021.

Erziehungsdirektion des Kantons Zürich: Lehrplan 21. Medien und Informatik. Edited by Lehrplan für die Volkschule des Kantons Zürich.

Evans, Dylan Jones (2003): Regional Innovation Policy for Small-Medium Enterprises. Intellectual Properties and Innovation Management in Small Firms. Review Article. In *SAGE Publications*.

FHNW-Strategylab-Strategieentwicklung-Digitales-Zeitalter (2021).

Fitzgerald, Brian; Mockus, Audris; Zhou, Minghui (2019): Towards engineering Free. Communications of NII Shonan meetings / editors, Brian Fitzgerald, Audris Mockus and Minghui Zhou. Singapore: Springer.

Galliers, Robert D.; Leidner, Dorothy E.; Simeonova, Boyka (2020): Strategic Information Management. Theory and Practice. Fifth Edition.

Gibbs, Samuel (2014): From Windows 1 to Windows 10: 29 years of Windows evolution. Microsoft's Windows operating system was first introduced in 1985. Over 29 years later a lot has changed, but what things have stayed the same? The Guardian. Available online at https://www.theguardian.com/technology/2014/oct/02/from-windows-1-to-windows-10-29-years-of-windows-evolution.

González-Blázquez, José Luis; García-Holgado, Alicia; García-Peñalvo, Francisco José (2021): Open approach of scaled agile for organizations and communities dedicated to the development of Open-Source projects. In Marc Alier, David Fonseca (Eds.): Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21). TEEM'21: Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality. Barcelona Spain, 26 10 2021 29 10 2021. New York, NY, USA: ACM, pp. 752–757.

Gupta, Sunil (2018): Driving digital strategy. A guide to reimagining your business. Boston, Massachusetts: Harvard Business Review Press (1).

Gutzwiller, Thomas, A. (1994): Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informatio. Heidelberg, Germany: Physica-Verlag.

Hackbartha, Gary; Groverb, Varun; Yi, Mun Y. (2001): Computer playfulness and anxiety: positive and negative mediators of the system experience effect on perceived ease of use. In *Elsevier Science B.V.*

Haff, Gordon (2021): The State of Enterprise Open Source 2021: Four results that may surprise you,. Red Hat Blog. Red Hat. Available online at https://www.redhat.com/en/blog/state-enterprise-open-source-2021-four-results-may-surprise-you, checked on 10/22/2021.

Hedgebeth, Darius (2007): Gaining competitive advantage in a knowledge-based economy through the utilization of open source software. In *VINE* 37 (3), pp. 284–294. DOI: 10.1108/03055720710825618.

Hevner, Alan R.; March, Salvatore T.; Park, Jinsoo; Ram, Sudha (2004): Design Science in Information Systems Research.

Hoepman, Jaap-Henk; Jacobs, Bart (2007): Increased security through open source. In *Commun. ACM* 50 (1), pp. 79–83. DOI: 10.1145/1188913.1188921.

Hon, Mun-Wai; Russell, Greg; Welch, Michael (2010): Open Source Software Considerations for Law Enforcement. In *IT Prof.* 12 (6), pp. 18–23. DOI: 10.1109/MITP.2010.121.

Huysmans, Philip; Ven, Kris; Verelst, Jan (2008): Reasons for the non-adoption of OpenOffice.org in a data-intensive public administration. In *FM*. DOI: 10.5210/fm.v13i10.2238.

IBM Archives (2021): Mainframe. Family Tree and Chronology. IBM Archives. Available online at https://www.ibm.com/ibm/history/exhibits/mainframe/mainframe_FT1.html, checked on 11/21/2021.

International Organization for Standardization (2018): ISO 31000:2018(en) Risk management — Guidelines. International Organization for Standardization. Available online at https://www.iso.org/obp/ui#iso:std:iso:31000:ed-2:v1:en, checked on 5/25/2022.

Johansson, B.; Sudzina, F. (2008): ERP systems and open source: an initial review and some implications for SMEs. In *Journal of Enterprise Information Management* 21 (6), pp. 649–658. DOI: 10.1108/17410390810911230.

Karels, Michael (2003): Comercializing Open source Software. In *Commun. ACM*.

Königs, Hans-Peter (2017): IT-Risikomanagement mit System. Wiesbaden: Springer Fachmedien Wiesbaden.

Kotter, John, P. (1995): Leading change: Why transformation efforts fail. Edited by Harvard Business Review (3).

Kruger, Justin; Dunning, David (1999): Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments. In *Journal of Personality and Social Psychology* 77 (6), pp. 1121–1134. DOI: 10.1037/0022-3514.77.6.1121.

Kuhn, Bradley; Stallman, Richard (2001): Freedom or Power.

Lara Torralbo, Juan Alfonso; Aljawarneh, Shadi A.; Radhakrishna, Vangipuram; N., Arun (Eds.) (2021): International Conference on Data Science, E-learning and Information Systems 2021. DATA'21: International Conference on Data Science, E-learning and Information Systems 2021. Ma'an Jordan, 05 04 2021 07 04 2021. New York, NY, USA: ACM.

Leckel, Anja; Veilleux, Sophie; Dana, Leo Paul (2020): Local Open Innovation: A means for public policy to increase collaboration for innovation in SMEs. In *Technological Forecasting and Social Change* 153, p. 119891. DOI: 10.1016/j.techfore.2019.119891.

Lee, Deishin; Mendelson, Haim (2008): Divide and Conquer: Competing with Free Technology Under Network Effects. In *Production and Operations Management* 17 (1), pp. 12–28. DOI: 10.3401/poms.1070.0005.

Lewis, Nathan; Case, Andrew; Ali-Gombe, Aisha; Richard, Golden G. (2018): Memory forensics and the Windows Subsystem for Linux. In *Digital Investigation* 26, S3-S11. DOI: 10.1016/j.diin.2018.04.018.

Li, Yan; Tan, Chuan-Hoo; Xu, Heng; Teo, Hock-Hai (2010): Open Source Software Adoption. Motivations of Adopters and Amotivations of Non-adopters.

Macredie, Robert; Mijinyawa, Kabiru (2011): A theory-grounded framework of Open Source Software adoption in SMEs. In *European Journal of Information Systems* 20 (2), pp. 237–250. DOI: 10.1057/ejis.2010.60.

Mansfield-Devine, Steve (2017): Open source software: determining the real risk posed by vulnerabilities. In *Network Security* 2017 (1), pp. 7–12. DOI: 10.1016/S1353-4858(17)30005-3.

March, Salvatore T.; Smith, Gerald F. (1995): Design and natural science research on information technology. In *Decision Support Systems* 15 (4), pp. 251–266. DOI: 10.1016/0167-9236(94)00041-2.

Marco Zwahlen (2018): Digitale Unabhängigkeit hat Grenzen. Der Ständerat hat sich gegen eine Motion ausgesprochen, welche die absolute ditale Unabhängigkeit der militärischen Systeme vorantreiben wollte. Edited by Eidgenössisches Departement für Verteidigung, Bevölkerungsschutz und Sport. Eidgenössisches Departement des Innern

EDI Bundesamt für Statistik BFS. Available online at https://www.vbs.admin.ch/de/aktuell/meldungen/parlament.detail.news.html/vbs-internet/parlament/2018/180531b.html, checked on 5/19/2022.

Mayring, Philipp (1994): Qualitative Inhaltsanalyse.

Mayring, Philipp (1995): Einführung in die qualitative Sozialforschung.

Mayring, Philipp (2014): Qualitative Content Analysis. Theoretical foundation, basic procedures and software solution. In *Gesis Leibniz-Institute für Sozialwissenschaften*.

Mayring, Philipp (2016): Einführung in die qualitative Sozialforschung.: Beltz Verlagsgruppe.

Miller, Keith W.; Voas, Jeffrey; Costello, Tom (2010): Free and Open Source Software. In *IT Prof.* 12 (6), pp. 14–16. DOI: 10.1109/MITP.2010.147.

MITRE (2021): CVE-2021-44228. With assistance of Apache Software Foundation. MITRE. Available online at https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228.

Nagle, Frank; Wilkerson, Jessica, Dana, James; Hoffman, Jennifer, L. (2021): Vulnerabilities in the Core. Preliminary Report and Census II of Open Source Software, checked on 12/25/2021.

Odoo Community (2021): Compare Odoo Editions. Odoo Community (open-source) and Odoo Enterprise (licensed). Odoo Community. Available online at https://www.odoo.com/page/editions, checked on 12/8/2021.

OECD Publishing (2019): Measuring the Digital Transformation: A Roadmap for the Future. Roadmap: Measuring open source software. Paris.

Open Source Studie Schweiz. Forschungsstelle Digitale Nachhaltigkeit (2021).

OpenExpo Europe (2021): The Future of the Open Source: Survey the new stack, The Linux Foundation and the TODO Group. Available online at https://openexpoeurope.com/en/the-future-of-the-open-source-survey-the-new-stack-the-linux-foundation-and-the-todo-group, updated on 4/4/2021, checked on 12/8/2021.

otter.ai. Version 22 (2022): Otter. Available online at otter.ai.

Paulson, J. W.; Succi, G.; Eberlein, A. (2004): An Empirical Study of Open-Source and Closed-Source Software. In *IIEEE Trans. Software Eng.* 30 (4), pp. 246–256. DOI: 10.1109/TSE.2004.1274044.

Peppard, Joe; Ward, John (2004): Beyond strategic information systems: towards an IS capability. In *The Journal of Strategic Information Systems* 13 (2), pp. 167–194. DOI: 10.1016/j.jsis.2004.02.002.

Petrinja, Etiel; Succi, Giancarlo; El Ioini, Nabil; Sillitti, Alberto (2013): Open Source Software: Quality Verification. Berlin, Heidelberg: Springer Berlin Heidelberg (404).

Pfaffman, Jay (Ed.) (2020): Transforming High School classrooms with Free/Open Source Software: it's time for an Open Source Revolution. 2020 43rd International Conference on Telecommunications and Signal Processing (TSP). Milan, Italy, 7/7/2020 - 7/9/2020: IEEE.

Rees, J; Altinkemer, K.; Sridhar, S. (2006): Vulnerabilities and Risk Management of Open Source Software. An Empirical Study. In *CERIAS Tech Report 206-75*.

Riehle, Dirk (2007): The Economic Motivation of Open Source Software: Stakeholder Perspectives. In *Computer* 40 (4), pp. 25–32. DOI: 10.1109/MC.2007.147.

Ross, Jeanne W.; Beath, Cynthia; Goodhue, Dale (1996): Develop Long-Term Competitiveness through IT Assets. In *Sloan Management Review*.

Saldaña, Johnny (2013): The coding manual for qualitative researchers. 2nd ed. Los Angeles: SAGE.

Silic, Mario; Back, Andrea (2016): The Influence of Risk Factors in Decision-Making Process for Open Source Software Adoption. In *Int. J. Info. Tech. Dec. Mak.* 15 (01), pp. 151–185. DOI: 10.1142/S0219622015500364.

SME Portal for small and medium-sized enterprises (2021): Software: Tips for choosing the right one. Eidgenössisches Departement des Innern EDI Bundesamt für Statistik BFS. Bern, Switzerland. Available online at https://www.kmu.admin.ch/kmu/en/home/concrete-know-how/sme-management/it-security-and-infrastructure/information-technology-infrastructure/software.html.

SME Portal for small and medium-sized enterprises; KMU (2021): Unified payroll procedure. ELM software (unified payroll procedure), developed and certified by Swissdec, allows companies to organize and regulate the transmission of payroll data to social insurance agencies and authorities. Eidgenössisches Departement des Innern EDI Bundesamt für Statistik BFS.

Stallman, Richard (2009): ViewpointWhy "open source" misses the point of free software. In *Commun. ACM* 52 (6), pp. 31–33. DOI: 10.1145/1516046.1516058.

Startwerk (2010): Buchhaltung leicht gemacht. Gestern Mittwoch wurden in Zürich die CH Open Source Awards vergeben. In der Kategorie Business Case teilt sich Run my Accounts den ersten Platz mit Doodle. Edited by Open Source Awards. Zürich, CH. Available online at https://www.startwerk.ch/2010/09/02/open-source-awards-buchhaltung-leicht-gemacht/.

Swissdec (2021): Verzeichnis zertifizierte Lohnbuchhaltungen - Swissdec.

Synopsys Editorial Team (2019): Top open source licenses and legal risk for developers. Available online at https://www.synopsys.com/blogs/software-security/top-open-source-licenses/.

The Linux Foundation (2021): Setting an Open Source Strategy. Open Source Guides. With assistance of Ian Varley, Software Architect at Salesforce, Guy Martin, Director, Open at Autodesk, Autodesk, Joe Beda, co-founder of Kubernetes at Google and co-founder and CTO at Heptio, Sarah Novotny, Kubernetes community program manager at Google, Chris Aniszczyk, CTO at CNCF, Luke Faraone, software engineer at Dropbox, Jim Jagielski, Open Source Chief at Consensys. The Linux Foundation. Available online at https://www.linuxfoundation.org/tools/setting-an-open-source-strategy/, checked on 10/22/2021.

The Open Source Initiative (2007): The Open Source Definition. Available online at https://opensource.org/osd, checked on 12/25/2021.

Tillmann Braun (2022): Umdenken bei der Digitalisierung ist Pflicht. Digitale Unabhängigkeit. With assistance of Diana Künstler. Edited by funkschau Business Netzwerk. Germany. Available online at https://www.funkschau.de/markt-

trends/umdenken-bei-der-digitalisierung-ist-pflicht.195636.html, updated on 4/22/2022, checked on 5/19/2022.

Wade, Michael; Hulland, John (2004): Review: The Resource-Based View and Information Systems Research: Review, Extension and Suggestions for Future Research. In *MIS Quarterly* 28 (1).

Wen, Shao-Fang; Kianpour, Mazaher; Kowalski, Stewart (2019): An Empirical Study of Security Culture in Open Source Software Communities. In *IIEEE Trans. Software Eng.* Available online at https://doi.org/10.1145/3341161.3343520.

Appendix A. Interview Guidelines

Guideline for Service Providers:

1. Are you aware of the free software movement?
    a. What does “free” means to you? Free as in Speech?
    b. What is free software for you?
2. What makes SMEs consider or not open-source software?
3. What aspects matters most to your clients about open source?
    a. From question 3 possible answers leads to question 4: (Network community, digital sovereignty, security, is it functional, fulfills the need, can get the necessary support, upgrade that breaks the software, one of the problems is that it can change at any moment)
4. How would (or does) OSS software benefit SMEs on their daily work?
   (Ask follow-up questions.: Cost: How much money do you estimate the cost is reduced by using OSS?
   Networking effect: Why do your colleagues/company use OSS?
   Digital sovereignty: I have control of my data: Can you give me an example of OSS helping you retain control of your data, have most of the tools that I need: What about the security part of using OSS?)
5. Is the Swiss market receptive to open-source services and product providers?
6. What are the barriers that you have experienced in selling your services to SMEs?
7. How does your company integrate the use of FOSS with commercial software?
8. What are the strongest arguments for convincing an SME to use open source not only in a backend system, but also in their application software?
9. Do your clients refer your services to SMEs?
10. Do you sponsor open innovation activities in conjunction with SMEs for example coding competitions, hackathons, challenges etc.?
11. Do you think that it will be possible to adopt a 100% OSS ecosystem in a SME?
12. It is stated that a platform should not be “black and white”. A platform can be hybrid, what are your thought about this statement?
13. Would your company deploy more FOSS if there were more SME companies interested in FOSS products?

Guideline for CEO's, Researchers

1. Are you aware of the free software movement?
    a. What does "free" means to you? Free as in Speech?
    b. What is free software for you?
2. What makes your company consider or not open-source software?
3. What aspects matters most for you about open source?
    a. From question 3 possible answers leads to question 4: (Network community, digital sovereignty, security, is it functional, fulfills the need, can get the necessary support, upgrade that breaks the software, one of the problems is that it can change at any moment)
4. How would or how does OSS software benefit on your daily work?
   (Ask follow-up questions.: Cost: How much money do you estimate the cost is reduced by using OSS?
   Networking effect: Why do your colleagues/company use OSS?
   Digital sovereignty: I have control of my data: Can you give me an example of OSS helping you retain control of your data, have most of the tools that I need: What about the security part of using OSS?)
5. Do you use any OSS Applications like Open-Office for example?
6. Do you have in-house software/system development?
    a. Yes,
        i. have you considered being part of the OSS community and release some of your software?
        ii. How would (does) you company benefit or not by sponsoring open innovation activities for example coding competitions, hackathons, challenges etc.?
7. Do you use FOSS? Where and how?
    a. Yes
        i. How is the software deployed? (on premises or cloud (SaaS, PaaS, IaaS), or other)
        ii. Does your organization outsource some or all of IT products and services? Which ones?

iii. How would you react when you notice that another organization is using FOSS? Would you be interested in knowing how are they adopting it?
iv. Have you seen any competitive advantage of using any FOSS application software? Is there a direct connection to business performance?

b. No
i. Why not?
ii. What would make you consider using FOSS?

8. What is your role in the decision process for selection and use of software systems in your company?
9. Have you proposed the use of FOSS for either system software or application software to your superiors/team?
    a. Yes, what did you propose? What were their responses?
    b. No, why not?
    c. What would it take for you to propose the use of FOSS in your company?
10. Have you or your company had any negative or positive effects that might have led to not using FOSS?
11. Would your company deploy more FOSS if there were more service providers of FOSS products?
12. Would you be willing to pay to migrate your information to an open-source software?
13. Would training will be needed in case of the adoption of OSS?
14. Since OSS is developed without a precise requirement profile, how would your company handle the process support to get to the point of becoming an optimal process support for your company? Is customization a problem? Would you need support to customize your clients' needs to the OSS software?
15. Do you think that an early education about OSS would have a positive effect in the adoption of OSS from future entrepreneurs?
16. Is the Swiss market receptive to OSS and are enough OSS service providers?
17. How does your company integrate the use of FOSS with commercial software?
18. Do you think that it will be possible to adopt a 100% OSS ecosystem in a SME?

19. Would your company deploy more FOSS if there were more clients interested in FOSS products?