Project Management Development – Practice and Perspectives
8th International Scientific Conference on Project Management in the Baltic Countries
April 25-26, 2019, Riga, University of Latvia
ISSN 2256-0513, e-ISSN 2501-0263

ANALYSIS OF SWITCHING COSTS OF USING FLOSS FROM THE PERSPECTIVE OF PROJECT MANAGEMENT: OPPORTUNITIES AND THREATS

Cielo Gonzalez Moyano
Technische Universität Berlin, GHermany

Abstract

Given the increasing use of Free/ Libre and Open Source Software (FLOSS) in companies and the role of project managers leading the implementation of these solutions; it is more than relevant to evaluate the opportunities and threats that FLOSS brings into the project management ambit because of its own nature. These opportunities and threats are analyzed in relation to switching cost and the project management diamond. This analysis shows which of them could influence specific types of switching costs and in which aspects of project management may be taken into account in a FLOSS implementation project.

Keywords: Free/ Libre and Open Source Software, FLOSS, switching costs, project management diamond, open source implementation project.
JEL Code: O3

Introduction

Free/ Libre and Open Source Software is software that could be considered Free Software and Open Source Software. Whitout dealing with the controversy of what specifically implies to be one or another, it is possible to describe it as openly accessible source code, and based on this definition complete business models have been created. As impossible as it could sound to someone, the idea of freedom of distribution, freedom of use, not restrictions over the software and the knowledge behind it, and where one of the most important sources of work to create this knowledge comes from voluntary work, has developed a sustainable business model generally focused on services related to the use of FLOSS software (Donald & Foulonneau, 2014). A few years ago, FLOSS was just considered a topic related mainly to developers who were interested in sharing and creating software for their own needs and interests (Lakhani & von Hippel, 2002;Lakhani & Wolf, 2003;Demaziere, Horn, & Jullien, 2006;Jullien & Zimmermann, 2011). However, this impression is not up to date. Today this business model has penetrate the coorporate environment to the point that even if independent developers are still relevant in the open source community, the companies are one of the strongest investors in Open Source projects. Google, Facebook, Adobe, and even Microsoft consider Open Source as indispensable for their businesses (Sijbrandij, 2016; Vaughan-Nichols , 2018), which is not a surprise, given that it is a central point that attracts innovation and revenues.

FLOSS is generating an impact from the economy to the Information Society. Its spread and influence represents one of the most important achievements in the IT industry, and in other sectors by extension of the relevance of technology in the society. Gone are the days where Open Source Software (OSS) was considered a violation to intellectual property, now successful business models are created around it. Open Source Communities have really clear that customers do not buy technology, they buy solutions to their problems, and the complexity of the problems evolve pregressively with the level of innovation. This generates an economy where knowledge as a mutual resource is more than valuable, it is the key element of evolution. However, knowledge itself does not change the global economy, and Open Source

Communities know that. They have created solutions that throght the years have not just support a global technology growth integrating knowledge around the world, but also have boosted high quality software that compete with proprietary software without laying aside the FLOSS vision and the community support. They have created a sustainable business model from that, and they continue changing according to society's requirements. Normally, these businesses are oriented to offer services for OSS, generate incomes from commercial licences, propietary extensions of Open Source solutions, complements to allow proprietary software to interact with highly accepted OSS, and solutions that immediately solve specific needs. These are just some examples of how adaptable are business model around FLOSS. This adaptability has achieved the point where it is possible to find companies such as Black Duck Software that despite not directly contribute to Open Source projects, helps organizations to make easier the use of FLOSS. They offer services for identifying and managing FLOSS, ensuring compliance with licences, and identifying vulnerabilities. However, what it is even more impressive is that they have more that 2000 clients which include Nintendo, Intel, SAP, Samsung, and Siemens. This shows how far FLOSS has penetrated the corporate environment, that now companies are willing to pay to administrate their Open Source solutions. As is to be expected, companies do not invest where there is not a higher direct or indirect benefit. Therefore, it is relevant to analyze some of the benefits that FLOSS offers, and that attracts companies to use it:

- The opportunity of having a wide community of motivated people creating a product, increases the quality of it and allows to have fixed errors in a not significant amount of time.
- Legal right to use the software has been one of the key points in the consolidated growth of FLOSS, and depending from the Open Source license, it has provided companies the opportunity of use, modify, and even commercialize the product.
- Opportunity to pay for additional services if they are needed is also an option. When the company does not want or is able to provide specific services that a product could need, there are companies that support these processes. As was mentioned before, this is one of the main business models around FLOSS that represents incomes, which is why the services tend to be more approachable and reactive.
- Business agility in a constantly changing environment is a valuable opportunity offered by FLOSS to the companies. Through the use of Open Source Software, companies are able to generate faster solutions to their needs without the complexity of dealing with contracts management.
- Reduction of cost is one of the most significant advantages that FLOSS can offer to companies. Because they are not worried about spending millions of dollars developing software from zero, they can focus on what is really important, genereting value with their products.

The list of benefits could continue, but the question in that case should be why, if there are so many gains on using FLOSS, there are still companies and sectors which do not use Open Source solution. According to the 2018 Open Source Program Management Survey implemented by The Linux Foundation with the support of The New Stack and TODO, 53% of the companies say that they use Open Source Software or have plans to start to use it in a short period of time. This survey also presents that 85% of the IT companies with more that 10000 employees have or will have an Open Source Program, but also it shows that 74% companies with the same amount of employees in sectors such as multimedia, telecomunication, media, and financial are doing exactly the same. However, these percentages dicrease when we start to talk about companies with less than 10000 employees. In this case the use of Open Source is 56% in IT companies and 47% in others (The New Stack, 2018).

With this study, we can conclude that the size and the sector of the company influence how open they are to use FLOSS. However, the percentage associated to medium size companies is relevant, and according to the development process that FLOSS has had in the last years, we can predict this percentage will grow. For this reason, it is relevant to analyze some of the challenges that companies may face at the moment of starting an Open Source Program. The 2018 Open Source Program Management Survey presents as some of the main challenges: the strategy planning or knowing how to approach it (54%), getting executive support and buy it (36%), finding legal staff with Open Source expertise (29%), identify budget and estimating cost (25%), and finding an Open Source Program manager (17%).

Additionally to the previous aspects, there is also the challenge of the lock-in effect, which refers to a situation when the customers are forced to maintain a relationship with a specific provider because of the cost and the uncomfortable situation that could emerge from changing the provider (Eurich & Burtscher, 2014). These costs are called switching costs, and FLOSS business model is also influenced by this situation.While there is several information regarding the relation between the lock-in effect of propietary software, and what it means for open source projects inside of the industry, little is known about the perspective of project management in this kind of projects.

Given that project managers are one of the roles that will be the responsibles of leading the future of Open Source Programs inside of different companies, it is more that logic to evaluate Open Source implementations since the point of view of project management. In order to achieve this, we could evaluate different aspects; however, the focus of this paper is one of them, costs. In specific, this document will evaluate switching cost according to the project management diamond.

In this way this study will attend to contribute to some of the challenges of future Open Source Programs by analyzing from a project management perspective the three types of switching cost associated with the implementation of a FLOSS project. This analysis generates a set of opportunities and threats that could help project managers to understand how to approach a project of this type, how to justificate the project inside of the company and get executive support, and understand which specific aspects should be taken in mind regarding estimating costs at the moment of change from a proprietary to a FLOSS application. By helping project managers to evaluate financial, procedural, and relational switching cost in FLOSS projects with a focus on scope, quality, time and expectations; this document pursues the possibility of increasing the number of project/program managers with the skills to manage a transition to Open Source. As was stated before, this is a skill which is highly needed in today's business environment.

The remainder of this paper is structured as follows. Section II describes the basic concepts needed regarding FLOSS, Switching Cost and Project Management. Section III presents the relation between Switching Cost and FLOSS, and the analysis of this relation from the project management perspective is stated in Section IV. Finally, Section V presents conclusions and future work.

FUNDAMENTAL CONCEPTS

Free/Libre and Open Source Software Concepts

Free/Libre and Open Source Software refers to the software that stands for the four freedoms: freedom to execute the software, freedom to study and analize it, freedom to redistribute it, and freedom to redistribute the own work that is done over the application.

Switching Cost Concepts

Burnham, Frels, et al (2003) define switching cost as the "one time cost that customer associate with the process of switching from one provider to another". Aditionally, they also emphasize the importance of not associate them just with the immediate switching, but also with the process in general. This paper take into consideration this definition, and analized switching cost in the context of a software implementation until the operation is normalized. Normalized is defined as the moment when a company has in operation the software and has a plan for support, future developments in case these are necessary, and has already accepted what they win and what they lose with the change. The switching cost that are analized here correspond with the typology introduced Burnham, Frels, et al (2003) and Blut, Evanschitzky, et al (2016) . This typology is decribed below:

- Finantial switching cost: these costs represent suck cost, lost performance costs, and benefit loss cost.
- Procedural switching cost: these cost represent uncertainly costs, search cost, cognitive costs, and setup costs.
- Relational switching cost: these cost represent personal relationship cost, and brand relationship cost.

Project Management Concepts

The Project Management Triangle states that a change in one of the vertices will affect the others, and in order to maintain the level of quality in a project, it is necessesary to compensate the change in the other vertices.

Most recently, a new model has been introduced. While it is not clear it origins, the Project Management Diamond shows a novel approach. It presents the Expectations as an aspect that could be influenced by scope, time, cost, and quality.

This paper presents the idea of Project Management Plus in Fig. 1. In this perspective, Expectation is influenced by scope, time, cost and quality. If one of them change, the expectations will be affected, and the others should be modified to keep he balance. However, expectations are also able to influence the four coinstraints, and that is why it is managed as one of the other vertices in this document.

ANALYSIS OF SWITCHING COST OF USING FLOSS FROM THE PERSPECTIVE OF PROJECT MANAGEMENT

The main goal of a company for using Open Source instead of proprietary software varies. For some organizations, saving is the most important factor, for others is the flexibility and others is the need of fast developments. While the reason for using FLOSS change from company to company, what normally does not change, is that these companies have to face switching costs. The literature states that switching costs are classified in three categories: procedural, finantial and relational costs.

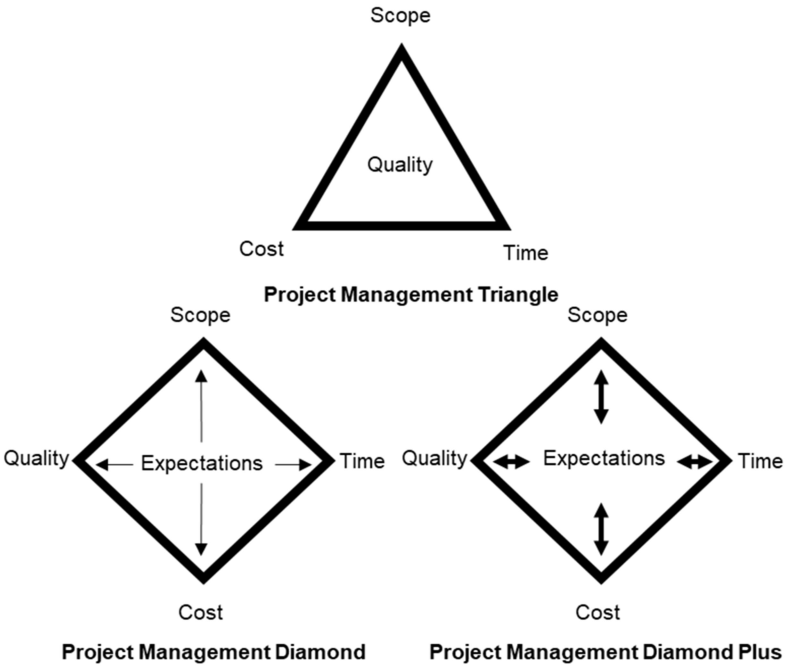
Scope
Quality
Cost
Time
Project Management Triangle
Scope
Quality
Expectations
Time
Cost
Project Management Diamond
Scope
Quality
Expectations
Time
Cost
Project Management Diamond Plus

Fig 2. **Project Management Diamond Plus**

Procedural switching costs, also known as direct switching costs, are the first cost that a company has to confront when it is switching from proprietary to FLOSS. In this category, we can find as examples, the effort and time of finding a new provider, the uncertainty of adopting a new service, set-up costs and cognitive cost. On the other hand, finantial switching costs are easier to quantify because they are associated with loss of financial value. These costs include direct monetary expected costs such as the fee for breaking a contract or the fee to start a new one. However, they also include the cost of the perception of decreasing on financial performance and having sunk costs, which although are subjective in comparison with strict quantity values, have a relevant impact on a switching decision. This decision is mainly influence for the last type of switching costs, the relational switching cost. Personal relationships, emotional bonds, and identification with a brand are some of the results of bonding and loyalty strategies, and have the most important role at the moment of change from one product to another (Blut, et al., 2016).

Normally, a company considers changing from a proprietary to an Open Source application an implementation project, and in this context in the best scenario a project manager will be involve since the decision process. Additionally, in some cases a project manager may also be involved in the analysis of strategies and business decisions (Project Management Institute, 2018). That means that analyzing switching costs in the implementation of FLOSS applications from a project management perspective makes more that sense, it is vital. No matter why the organizational leaders decide to initiate these kind of projects, either because of legal requirements, stakeholders needs, changes in business or technological strategies or improvements in their products, process or services; a project manager is going to be the responsible for initiating, planning, executing, monitoring and controlling, and closing the project. Through these Project Management Processes, the main function of a project manager will be to integrate the Project Management Knowledge Areas by tailoring tools, skills, knowledge and techniques according to the needs of the project. But what should have in mind a project manager when he/she is in charge of a project that consists in changing a proprietary software for an Open Source solution? What are the implications of switching costs in this process? Is an implementation of a FLOSS application as the implementation a proprietary application? Or are there specific opportunities and challenges that a project manager should evaluate regarding these type of projects? One way to unify in some extend all of these questions in one is asking: what are the opportunities and threats regarding switching cost of using FLOSS from the perspective of project management?

In this document, we pursue to answer this question by analyzing opportunities and threats of implementing a FLOSS solution taking into account particularities of Open Source business models. These opportunities and threats are related to switching costs in a way that they could affect them by reducing in case of an opportunity or increasing in case of risk. The paper identifies to which of these switching costs each of the opportunities or threats could be related. Additionally, it also suggests in which of the aspects of project management these opportunities and threats should be taken into account in an implementation project. Fig. 2 presents graphically this goal, and Table 1 and Table 2 show the result.

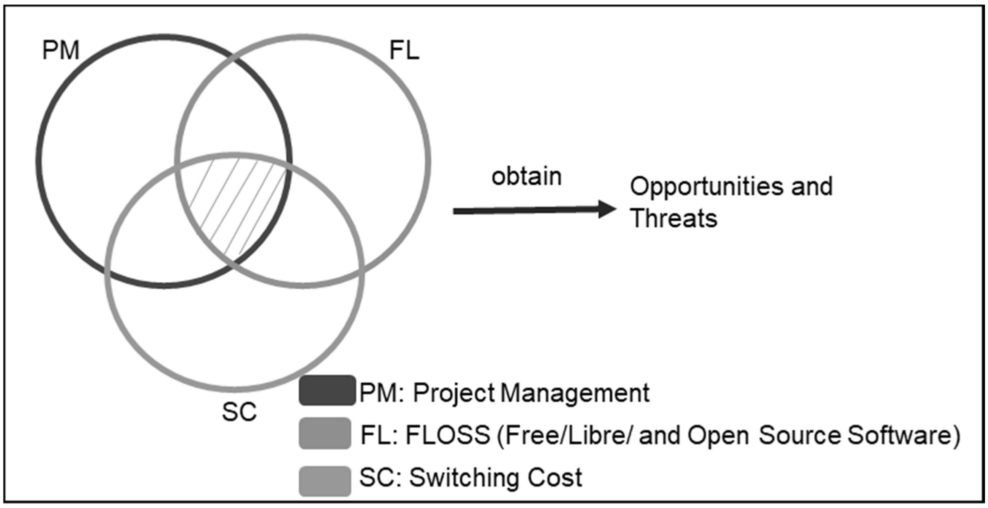
PM
FL
SC
obtain
Opportunities and Threats
PM: Project Management
FL: FLOSS (Free/Libre/ and Open Source Software)
SC: Switching Cost

Fig. 3. **Strategy to obtain opportunities and threats regarding the switching cost of using FLOSS from the perspective of project management**

The opportunities and threats presented in this paper are the result of unifying the most common aspects that are mentioned as benefits or risks in formal and informal literature.

Opportunities

Taking into account that switching costs include all the cost that a company has to face when changing suppliers, and they are not just the cost at the moment of the switching (Burnham, et al., 2003); it is important to analyze the switching cost since this perspective. For example, when changing one software for another, a project manager shall evaluate the financial costs of breaking the previous contract and the suck costs. This includes among other aspects fines, monetary privileges that the contractor receives from the provider and that will not receive any more, and the monetary value that has been invested in the current software to achieve the expected functionality and that maybe could lose with the new application. However, a project manager also has to analyze the future financial implications of the change until the operation with the new software is normalized. In this paper, we consider normalized as the moment when a company has in operation the software and has a plan for support, future developments in case these are necessary, and has already accepted what they win and what they lose with the change.

This holistic analysis of the process shall be generated from the financial, procedural and relational point of view. According to this aspect, this paper analyzes in the Table 1 what would be the particularities of switching to an Open Source Software since the project management diamond perspective. The notation in this table is: Type of Switching Costs (TSC), Quality (Q), Time (T), Scope (S), Cost (C), and Expectations (E).

Table 1

Opportunities of using FLOSS and their relation with Switching Cost and the Project Management Diamond

<table>
<tr><th>Aspect</th><th>Type</th><th>TSC</th><th>Q</th><th>T</th><th>S</th><th>C</th><th>E</th></tr>
<tr><td>1</td><td>Opportunity</td><td>Procedural</td><td>x</td><td>x</td><td></td><td></td><td></td></tr>
<tr><td colspan="8">Normally, to execute research on quality attributes over a proprietary software in case of an auditory, the companies need explicit permission from the provider to access the source code or they need to use black box techniques (Spinellis, et al., 2009). This kind of research are executed throught specialized companies that charge a considerable amount of money. On the other hand, the use of Open Source Software allows an open inspection of the source code and the asociated data which ensure the quality of the applciation. Aditionally, tracking data bases, wikis, and forums make possible a transparent evaluation of the quality an security of the software.
Before deciding to implement a new application in a company, it is possible to have the need of identifying quality aspects that could affect the decision, which could be done easily with OSS.</td></tr>
<tr><td>2</td><td>Opportunity</td><td>Procedural</td><td>x</td><td></td><td></td><td></td><td>x</td></tr>
<tr><td colspan="8">Before acquiring an application, companies perform a market study between the different possible applications. Candidate providers present brochures, proposals, demonstrations of the application and other activities that are required to sell an application. Selling is the main goal, and that is why is not common that they present quality issues in the application. Thus, the probability of having a realistic view of the status of an application is extremely low, making difficult to compare candidate applications since a detailed quality point of view. In this scenario, companies normally take one of two options, either do not evaluate the candidate applications in detail or to hire an external company with expertise in the domain, which recommends one of the providers according to the business model to be implemented.
In comparison, the broad number of Open Source Projects that compete between each other in the same field, allow to compare them in quality aspects without the intervention of commercial or marketing areas (Spinellis, et al., 2009). Having the possibility of comparing the quality criteria of different applications, reduce search and uncertainty cost.</td></tr>
<tr><td>3</td><td>Opportunity</td><td>Procedural</td><td>x</td><td></td><td></td><td></td><td>x</td></tr>
</table>

Aspect	Type	TSC	Q	T	S	C	E
The access to general historical information related to bugs and solutions from Open Source Projects is valuable at the moment of finding the causes to specific problems and possible solutions (Spinellis, et al., 2009). Compare with proprietary software, where companies have access just to the bugs and solutions that are related to them, Open Source Software offers an efficient way to improve and monitor quality.							
4	Opportunity	Finantial				x	
Costs that have been paid for the proprietary software are going to be sunk cost. However, the advantage is that the costs of proprietary licensing shall not be taking into consideration anymore.							
5	Opportunity	Procedural			x	x	
The setup of an application is an opened and known process, which in most of the cases is documented. When it is not documented, it is possible to extract the knowledge. This allows to retain control over the own computing environment and customize it (Varian & Shapiro, 2003). The company gains flexibility to face unknown requirements, and with this to unknown costs.							
6	Opportunity	Relational	x				x
Depending on the selected Open Source application, the brand recognition of the company could increase due to the well-known application. Additionally, if the application is well known, that means that there is an important amount of people supporting the application. This increases the quality of the product.							
7	Opportunity	Procedural		x	x	x	x
Open source commonly uses open interfaces (Varian & Shapiro, 2003), but also they create interfaces with the most common proprietary software. This provides flexibility, and ensure that the costs, time, and scope of the application is controlled. Interfaces between systems are a key point during an implementation project, and having the opportunity of controlling them, reduce time and cost of negotiations. Furthermore, it is a benefit that could be used to engage users.							
8	Opportunity	Procedural /Finantial	x	x			x
The opportunity of having a wide community of motivated people creating a product increases the quality of it and allows to have fixed errors in a not significant amount of time. A fast respond generates positive expectations on final users and increases the perception of having a high performance.							
9	Opportunity	Procedural /Finantial		x		x	
Legal right to use the software has been one of the key points in the consolidated growth of FLOSS, and depending from the Open Source license, it has provided companies the opportunity to use, modify, and even commercialize the product.							

<table>
<tr><th>Aspect</th><th>Type</th><th>TSC</th><th>Q</th><th>T</th><th>S</th><th>C</th><th>E</th></tr>
<tr><td colspan="8">This is an enormous advantage in comparison to proprietary software because the company does not have to spend time or money in contracts or licenses. These processes are normally tedious, and depending on the provider and internal policies, it could take several months to concrete the purchase. Furthermore, if in the future the company wants to change the Open Source Software for another application, the suck cost will be minimal in comparison with a proprietary application.</td></tr>
<tr><td><b>10</b></td><td>Opportunity</td><td>Procedural/ Relational</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td colspan="8">Opportunity to pay for additional services if they are needed is also an option. These services could go from training, manuals, or a complete customized environment, implementation, data migration, until more continuous services such as support and personalized developments. When the company does not want or is able to provide specific services that a product could need, there are companies that support these processes. This is one of the main business models around FLOSS that represents incomes, which is why the services tend to be more approachable and reactive. Reactiveness and business agility in a constantly changing environment is a valuable opportunity offered by FLOSS to the companies (Araújo & Gava, 2012), and it helps to develop a strong relationship between the providers and their clients because it generates the feeling of having a partner who understands the urgency of your needs and reacts according to this.</td></tr>
<tr><td><b>11</b></td><td>Opportunity</td><td>Finantial/ Relational</td><td></td><td>x</td><td></td><td>x</td><td></td></tr>
<tr><td colspan="8">Reduction of cost is one of the most significant advantages that FLOSS can offer to companies. Because companies are not worried about spending millions of dollars developing software from zero, they can focus on what is really important, generating value with their products. This generates a feeling of high productivity and at the same time generates a stronger connection between FLOSS companies and their clients.
However, it is important to mention that in some cases the Total Cost of Ownership (TCO), which represent not just the initial price, but also the cost of training, support, and upgrade, does not differ significantly between specific proprietary software and Open Source in specific cases. For example, some studies mention that the difference in TCO between Linux and Windows is just within 10% and 15% (Varian & Shapiro, 2003). This is not a significant variation, but if the relation Cost/Benefit is analyzed, it is understandable why Open Source Software presents an advantage in relation to proprietary software.
This does not mean that there are not relevant economic advantages with other software, but this aspect should be analyzed carefully. An example of this is the Beaumont Hospital. In 2013 they had budgetary complications in the IT department that represented 17 million. As a measure to fix this situation, they decide to change all the applications to open source solutions. This change included basic products like email and desktop applications in the first phase, and the core applications of the business in the second phase. The reduction in the initial costs was 95,8% and in the running cost of five years was 95,3% (Fitzgerald & Kenny, 2004). So, the economic benefit of using OSS could be different depending on the strategy of the company and the applications are needed.</td></tr>
<tr><td><b>12</b></td><td>Opportunity</td><td>Finantial</td><td></td><td></td><td></td><td>x</td><td></td></tr>
<tr><td colspan="8">As it has been shown in this document, switching costs are relevant at the moment of chance an application. Companies procure flexibility at the moment of taking a decision, and</td></tr>
</table>

Aspect	Type	TSC	Q	T	S	C	E
they do not want to be lock-in to a specific solution. Although it is not possible to say that Open Source Software is lock-in-free, it is clear that for its own nature of FLOSS it is easier to change to another vendor because there is access to file formats, data, system calls, APIs, interfaces, and communication standards. This information is generally well documented, but if it is not the case at least there is access to the source code. That is something that is not possible in proprietary software. This means that the time and money that is invested in an Open Source Solution, is not going to be completely lost if the company decide to change the application.							
13	Opportunity	Relational	x				x
Although reliability is difficult to measure, studies show that Open Source Software is relatively equivalent or even more trustworthy than proprietary solutions. One of the main factors that make Open Source more reliable is that its developers are also its own users which ensure quality in the product. Furthermore, there are thousands of developers cross-checking their work, so the possibility of detecting errors or security problems is higher (Pandey, R. K. & Tiwari, 2011).							
14	Opportunity	Finantial/ Relational			x	x	
There are several licenses in the Open Source scenario, and these licenses could be classified as protective, non – protective, and public domain. Depending on the type of license, companies can even modify the software without returning the code back to the community (Public Research Centre Henri Tudor, 2014). It this case, sensitive information can be retained by paying the license or even without doing it if the license allows it. This ensures that a company could work with a specific open source application without being worried about not sharing changes.							
15	Opportunity	Procedural			x		x
Experimenting over an open source application in order to discover the functionalities is easier than with proprietary software (Dedrick & West, 2013), and with that, the possibility of finding benefits and gaps are higher. The reason for having this possibility is that a version of OSS is always accessible through the internet. In contrast with proprietary software, wherein the best cases you can access a just a trial before buying the software, this represents an enormous advantage because it allows to reduce the anxiety for the uncertainty and get early knowledge of the application.							

Risks

The same considerations that are taking into account to analyze the opportunities, are taken in the analysis of risks in the Table 2.

Table 2

Risks of using FLOSS and their relation with Switching Cost and the Project Management Diamond

Aspect	Type	TSC	Q	T	S	C	E
1	Risk	Relational		x		x	x
The same that commercial projects fail, Open Source projects are also in risk of not							

<table>
<tr><th>Aspect</th><th>Type</th><th>TSC</th><th>Q</th><th>T</th><th>S</th><th>C</th><th>E</th></tr>
<tr><td colspan="8">succeed. However, this risk could be mitigated or transferred. As a proposal to mitigate this situation, we suggest evaluating open source application as a company would evaluate a proprietary application. That means to evaluate years on the market, the number of users, and type of clients using the application and financial statements in case that is possible to get this information.</td></tr>
<tr><td><b>2</b></td><td>Risk</td><td>Procedural/ Relational</td><td>x</td><td>x</td><td></td><td></td><td>x</td></tr>
<tr><td colspan="8">The open source communities are effective at resolving issues but they are not obligated to that. There is not a contract which forces them to answer in a specific amount of time, or indeed to answer. This may restrict companies from using OSS. Nevertheless, for most of the OSS, there are agencies or freelancers who can offer this services at a cost and with a contract. So, this risk could be entirely avoided.
Another option to mitigate this risk is to have internal staff with the necessary technical expertise.</td></tr>
<tr><td><b>3</b></td><td>Risk</td><td>Procedural</td><td></td><td>x</td><td>x</td><td>x</td><td>x</td></tr>
<tr><td colspan="8">It has been mentioned that OSS is less user-oriented that proprietary software, and because of this, it is difficult for not technical users to adopt it. Without discussing how precise is this statement, it is important to have in mind that this is a commonly mentioned aspect that is not related to the implementation of open source solutions in companies (Mora, et al., 2016). In this scenery, the recommendation to avoid this risk is to evaluate the application using one of the selection models of FLOSS solution. Models like the Multi-Attribute Decision Making (MADM), Navica Open Source Maturity Model (OSMM), Quality Model for Open Source Selection (QMOSS), OpenSource Maturity Model (OMM), Open Business Quality Rating (OpenBQR), QualOSS, Software Quality Observatory for Open Source Software Model (SQO-OSS), Quality Platform for Open Source Software (QualiPSo), Method for Qualification and Selection of Open Source Software (QSOSv2), EFFORT, IRCA, and CapGemini could be used to evaluate not just the usability but also the software since product and organization attributes (Mora, et al., 2016).
Additionally, to reduce the resistance to FLOSS applications, it is important to dedicate time and budget for extensive training. It should be made clear that this effort is not exclusive for FLOSS applications since it is also a necessary process in proprietary software implementations.</td></tr>
<tr><td><b>4</b></td><td>Risk</td><td>Finantial/ Procedural</td><td></td><td></td><td></td><td>x</td><td>x</td></tr>
<tr><td colspan="8">Administrate different Open Source applications could be difficult because it is necessary to manage at the same time different compliance requirements which came with the different types of licenses. This is a situation that has been generated for the increasing use of FLOSS applications, but there are already companies such as Black Duck Software or White Source which provide solutions to manage this situation.</td></tr>
</table>

Conclusions

In the scenery of a FLOSS implementation project, the opportunities that FLOSS brings into the context according to the business models that support it and its own characteristics, are significant. Mostly, it influences procedural switching costs. From 15 relevant opportunities that are analyzed, 50% of them influence procedural switching cost, 27% financial and 23% relational. Additionally, regarding the number of aspects of project management that shall be taken into account, the result is balanced. The opportunities shall be analyzed in equal proportions in relation to quality, time, scope, cost and expectations.

On the other hand, it was not found a strong connection between the 4 risks of FLOSS presented in this paper and their impact in financial switching cost. However, a meaningful association exists at the moment of manage the expectations in a FLOSS implementation project.

In future work, the author plan to evaluate selection model of FLOSS since the perpective of project management.

References

Araújo, L. & Gava, R., 2012. Market Proactiveness Anticipating moments zero. In: *Proactive Companies.* London: Palgrave Macmillan, pp. 9-22.

Blut, M. et al., 2016. *Securing business-to-business relationships: The impact of switiching cost.* s.l., s.n., pp. 82-90.

Burnham, T. A., Frels, J. K. & Mahajan, V., 2003. Consumer switching costs: A typology, antecedents, and consequences.. *Journal of the Academy of marketing Science, 31(2),* pp. 109-126.

Burnham, T., Frels, J. K. & Mahajan, V., 2003. Consumer switching costs: A typology, antecedents, and consequences.. *Journal of the Academy of marketing Science, 31(2)*, pp. 109-126.

Dedrick, J. & West, J., 2013. Why firms adopt open source platforms: a grounded theory of innovation and standards adoption. In: *Proceedings of the workshop on standard making: A critical research frontier for information systems.* s.l.:s.n., pp. 236-257.

Demaziere, D., Horn, F. & Jullien, N., 2006. *How Free Software Developers Work: The Mobilization of 'Distant Communities'.* [Online] Available at: https://ssrn.com/abstract=1301572. or http://dx.doi.org/10.2139/ssrn.1301572

Donald, M. & Foulonneau, M., 2014. *Guide to Open Source Business Models - Leveraging a Free and Open Source Software framework to develop commercialization strategies for IT Research & Development projects.* s.l.:10.13140/2.1.4393.7281.

Eurich, M. & Burtscher, M., 2014. *The Business-to-Consumer Lock-in Effect.* s.l.:s.n.

Fitzgerald, B. & Kenny, T., 2004. Developing an information systems infrastructure with open source software. *IEEE Software, 21(1),* pp. 50-55.

Jullien, N. & Zimmermann, J.-B., 2011. *FLOSS in an industrial economics perspective.* s.l.:Revued'économie industrielle , Éd. techniques et économiques;De Boeck Université.

Lakhani, K. R. & von Hippel, E., 2002. *How open source software works: "free" user-to-user assistance.* s.l.:Research Policy 32 (2003) 923–943.

Lakhani, K. & Wolf, R., 2003. *Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects.* [Online] Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=443040

Mora, M., Marx Gómez, J., O'Connor, R. & Gelman, O., 2016. An MADM risk-based evaluation-selection model of free-libre open source software tools. *International Journal of Technology Policy and Management, 16(4),* pp. 326-354.

Pandey, R. K., R. K. & Tiwari, V., 2011. Reliability issues in open source software. *Proceedings of the International Journal of Computer Applications, 34(1).*

Project Management Institute, 2018. *A Guide to the Project Management Body of Knowledge. Sixth Edition.* s.l.:s.n.

Public Research Centre Henri Tudor, 2014. *Guide to Open Source Business Models - Leveraging a Free and Open Source Software framework to develop commercialization strategies for IT Research.* Louxenbourg: s.n.

Sijbrandij, S., 2016. *Forbes.* [Online] Available at: https://www.forbes.com/sites/forbestechcouncil/2018/07/16/how-open-source-became-the-default-business-model-for-software/

Spinellis, D. et al., 2009. *Evaluating the quality of open source software..* s.l., s.n., pp. 5-28.

The New Stack, 2018. *The New Stack.* [Online] Available at: https://thenewstack.io/survey-open-source-programs-are-a-best-practice-among-large-companies/

Varian, H. R. & Shapiro, C., 2003. *Linux adoption in the public sector: An economic analysis..* s.l.:s.n.

Vaughan-Nichols , S. J., 2018. *ZDNet.* [Online] Available at: https://www.zdnet.com/article/whats-the-deal-with-microsofts-open-source-friendly-patents/