Company Engagement in Open Source Software: A case study with Facebook

Term Paper

Open Source and IP in the Digital Society

WS2021
Anh-Thu Tran
Anh Truc Lam

Abstract

Motivation: Nowadays, companies are more open to invest into open source software (OSS) projects as the outcome of these projects are promising in many aspects. The reputation of OSS communities are perceived positively as they produce quality software as well as foster new ideas, innovation and resources extensions. **Aim:** In this term paper written in the course "Open Source and IP in the Digital Society" at the Technical University Berlin, we investigate on the reasons for companies to invest into OSS projects. **Method:** We conducted a literature review and build a case study based on two Facebook OSS projects: PyTorch and React. We identified 6 papers related to this topic. **Results:** The primary goal of this term paper is to present an overview of Facebooks commercial objectives including five main concepts: Software Quality, Outstanding Technical Support, Business Opportunity, Business Model and External Innovation based on the investigations of Zhou et al. (2016, pp. 7–8). Our findings reveals that 1. Software Quality and 2. Business Opportunity are likely the most important reasons for companies to invest into OSS projects.

Inhaltsverzeichnis

1 Introduction

In the age of digitization, to participate in the dynamic OSS community is becoming increasingly important for companies, which targets the question of how to the align OSS projects with commercial objectives (Zhou et al., 2016, p. 3). With the large number of participants and capabilities, open source projects become enormously complex and interrelated, requiring more professionalism and management in OSS development (Germonprez et al., 2018, p. 2). Currently, OSS projects face two contradictory values. On the one hand, OSS is still building on the "[…] traditions and ideals of the free software movement" and egalitarian perspectives (Germonprez et al., 2018, p. 14). But on the other hand, there is an urgent need of professionality of corporate participation, their long-term investment and support, which is a strategically important aspect in companies supply chain (Germonprez et al., 2018, p. 14). For this reason, companies want to reduce costs, leverage the potential for external innovation and raise their profit by getting involved in open source communities (Zhou et al., 2016, p. 2). For example, Linux, Apache, Wordpress and their open source project ecosystems, which played a significant role in the past in open source project development (Gastil & Davies, 2020, p. 4). However, the current situation reveals many risks as companies face various challenges to deploy, maintain and support OSS projects. One of the risks are the financial challenges to sustain such projects, which raise questions of how to invest into long-term projects as it is critically important to maintain external contributors (Germonprez et al., 2018, p. 14). In addition, companies involvement comes with commercial control that could strip communities of their code commit privileges (Zhou et al., 2016, p. 2). Several tech-companies suc as Microsoft, Google and Facebook have already recognized these opportunities and took participation into the open source community in order to develop and provide their own software such as Microsoft's ASP.NET framework and Facebook's OSS React JavaScript library (Kochhar et al., 2019, p. 1). To investigate the reasons for companies to invest into OSS projects, we took the company Facebook and its OSS projects in our investigation. This term paper proceed with the following structure. In section 2, we present definition and concepts of OSS projects. In section 3, we describe our methodology and analysis. After that, we present the results in section 4 and discuss our result to come to a conclusion in section 5.

2 Theoretical Foundations

This section provides a theoretical foundation of important definition and concepts of OSS projects in companies. We highlight and describe two OSS projects of Facebook.

2.1 Definition of Free/Libre Open Source Software (FLOSS) Projects

Initially FLOSS was simply called Free Software (FS), then in 1998 the term Open Source Software (OSS) was introduced to make it clear to users that their software products are not necessarily provided free of charge, but that their source code is freely accessible (Ballhausen, 2019, p. 82). The terms FS and OSS share the same license terms, which are also referred to as the four freedoms. Thus, in order for software to be licensed as FLOSS it needs to meet four criteria (Ballhausen, 2019, p. 83). According to Logothetis & Stylianidis (2016, p. 30), the first grants the user the freedom to run the program for any purposes the user deems fit. The second freedom allows the user to lookup any source code of the program and adapt its implementation to his needs. Users also should have the freedom to reallocate copies of FLOSS in order to assist others. Lastly, any user wields the freedom to allow access to copies of improved code in order to create benefit to the general public (Logothetis & Stylianidis, 2016, p. 30). As software itself is protected under copyright law regardless, if a software is licensed as FLOSS, the four licenses outlined above simply define the distribution terms for FLOSS (Ballhausen, 2019, p. 83). It remains to mention, that it is essential for companies to set up proper FLOSS governance mechanism in order to comply with the licensing requirements and to reduce licensing risk. These help to stay consistent with operational strategy by considering participants' background and motivation as well as align stakeholders' interests (Kemp, 2010, p. 1). The terms FS, OSS and FLOSS can be used interchangeably. However, for this paper we focus on the term OSS.

2.2 Open Source Projects as main driver in companies

In various papers, we find associations of the term open source projects in companies with the term corporate investment, which emphasizes how rapid the changes of open source itself through companies investment was (Germonprez et al., 2018, p. 2). Generally, there exists two types of commercial involvement of a company in OSS (Wagstrom, 2009, p. 8). A firm which is focused in building an active community in order to profit on its related services is called community-focused. If a firm is relying on

product sales it is called product-focused. With these underlying findings, Zhou et al., 2016 (pp. 7–8) developed the model "Dimensions of the community involvement models" as shown in

Figure 1. As a company firstly needs to verify the common objective of the project and the firm's strategy as well as assessing the opportunity to create future revenue. Zhou et al. (2016, pp. 7–8) separated this endeavour into the dimension commercial objective. Next, in order to work within the project community framework companies need to create the underlying environment for the community to engage. For instance, they need to agree on intellectual property (IP) and legal issues, how to allocate resources, and how to integrate their employees into the community (Zhou et al., 2016, pp. 8–9). This aspect is picked up as community involvement action dimension.
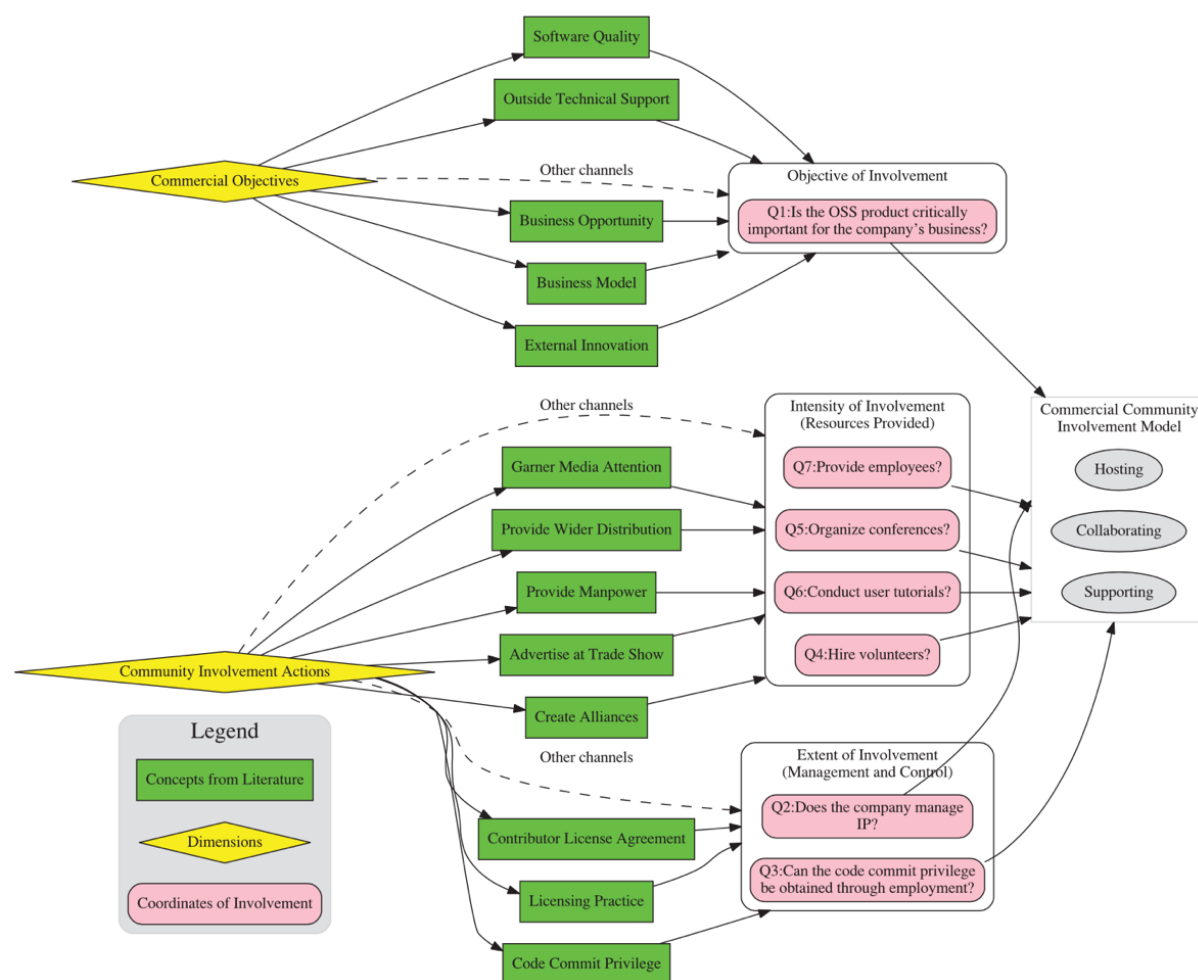

Figure 1 - Dimensions of the community evolvement models (Zhou et al., 2016, p. 8)

The authors identified and assigned serveral concepts to each of the two dimensions, highlighted in yellow by going through academic literature. Zhou et al. (2016, pp. 8–9) then illuminates each concept coloured in green by setting up seven questions, which tries to answer what fuels the companies involvement and how the companies try to reach their goals within the OSS project, coloured in pink (Zhou et al., 2016, pp. 8–9). Zhou et al. (2016, p. 7) investigated the purpose for a company to invest into OSS project and summarized the five key findings in the dimension *Commercial Objectives*:

Concepts from Literature	Description (Zhou et al., 2016, pp. 7–8)
1. Software Quality	Companies utilize external knowledge involvement in order to increase quality of OSS projects.
2. Outside Technical Support	Companies present their code base via OSS to raise outside feedback and technical support. They take advantage of creative ideas and capabilites of external contributors.
3. Business Opportunity	Companies pursue a strategic and long-term goal by keeping free access to their open source platform and cultivate business opportunities, such as middleware software or related services.
4. Business Modell	Companies nurture the open source community by strongly engaging with it. By aligning their business model with the OSS community or supporting community engagement through own employees, companies legitimize themselves to take advantage of newly developed products and services.
5. External Innovation	Companies with fewer resources can benefit by OSS initiatives of larger institutions. They can harness and rely on software quality, and continued improvement.

Table 2-1 - Concepts from Literature - Description

From this model we assessed, that the commercial objectives dimension largely contains answers, which are covering our research question, such that we recycle the concepts for further analysis, namely Software Quality, Outside Technical Support,

Business Opporuntity, Business Model and External Innovation (Zhou et al., 2016, p. 8). A description of each concept can be retrieved from Table 1 above. Here, the first question is: *Is the FLOSS product critically important for the company's business? (Q1).* Critically important can be understood in two perspectives, which is "[…] one, gaining profit directly from the OSS product; two, the OSS product is strongly associated with (or greatly helps to gain) profit." (Zhou et al., 2016, pp. 7–8). We deem the remaining concepts stemming from the second dimension unfit to answer our research question and have excluded this part of the model in our paper. It remains to mention that the paper of Zhou et al. (2016) was published within the past five years, during the creation of our paper, so that we can rely on its currentness and ultimately ensure the quality of our conclusions.

2.3 Open Source Projects in IT-Company Facebook

In the past two decades OSS development proved themselves as it has produced a large number of highly reliable projects, such as Linus Mozilla browser, MySQL database and the Hadoop framework (Kochhar et al., 2019, p. 1). Thus, the OSS framework remains an attractive model to build and deploy software. An important tool for developers to contribute to OSS projects is Git (Kochhar et al., 2019, p. 1). Git is closely following different versions of a software and is also described version control system (VCS). Its intrinsic features are the faciliation of distributed development and the capability to handle over thousand developers. Hence, social coding websites such as GitHub or Gitlab offering Git are enabling over 40 million users to collaborate and partake in OSS projects. Further, even large software companies, such as Google and Microsoft are using Git as their main development platform and started to open source part of their proprietary software as well (Kochhar et al., 2019, p. 1).

Observing the open source landscape one notices, that Facebook, a social networking service, is involved in OSS projects as well (Facebook, 2020). This company provides a social networking site, also named Facebook, which enables users to connect and share information easily with family and friends. Currently Facebook is the world's largest social network, with more than 2.3 billion users worldwide (Wikipedia, 2021). The company engages within numerous projects covering various large areas ranging from Artificial Intelligence, Web Technology and Operating Systems to Security etc. (Facebook, 2020). A member of the company's open source team shares, that Facebook move towards open source comes naturally as in its main mission the

company tries to create a world community. Second to that the firm tries to foster innovation and create better software, as Facebook is facing unique development challenges it has to solve. Here Open source serves the company producing better software and work more transparently (Flory, 2018, p. 1).

We want to take a look at the company Facebook in light of the concepts chosen above. We estimate, whether these can be utilized or need to reformulate to a reasonable extent.

Concepts from Literature	Application
1. Software Quality	We measure the metric "Rate of delivery". This indicates that the higher rates of software version delivery is the better quality software for customers (AltexSoft, 2017, p. 1). We follow the question: *"How often the version of a software comes with improvements that impacts the user?"*
2. Outside Technical Support	Contributors who are not employed by the same company, where the software was developed, contributed to this project. We follow the question: *"Where are the top 10 contributors on GitHub employed?"*
3. Business Opportunity	As a company tries to pursue a strategic goal by using open source as a business tool and benefit from solving their own and unique problems. We follow the question: *"Has the introduction of this software lead to more business opportunites?"*.
4. Business Modell	A company promotes OSS projects to potentially develop new products and services. To verify whether new software is beneficial, we measure Facebook revenue streams. We follow the question: "*Does the open source software create revenue?"*
5. External Innovation	We follow the question: *"Are other companies utilizing products stemming from advantages by these software companies?"*

Table 2-2 - Concepts Description for Case Study

2.4 Presented Case Study

To approach the research question, we want to illuminate reasons why Facebook is engaging in OSS projects based on scientific literatures and GitHub repositories. We identify and explain two OSS projects provided by Facebook: *PyTorch* and *React*.

2.4.1 Role of PyTorch in Open Source Projects

The PyTorch project released in 2016 is contributing in the field of Artificial Intelligence (AI) by providing a Python library for deep learning (Facebook, 2020). It gained popularity in the research community as its performance is similar to other deep learning libraries by making use of the widely adopted computer language Python familiar to many data scientists due to its simplicity (Lorica, 2017, p. 1). The objective is to put researchers first by aiding them in reducing difficulty in creating new models, handling and wrenching data (Paszke et al., 2019, p. 2). Further, on the 2nd of May 2018, Facebook announced PyTorch 1.0, bridging research and production. Developers still enjoy the possiblity to tweak their AI models quickly and experiment freely, but they can also rely on the library's performance, when lifting their models up onto production level. Due to the widespread approval of many AI researchers the framework has been downloaded over 1.1 million times and has been the second-most cited deep-learning framework in April 2018 (Jia, 2018, p. 1).

2.4.2 Role of React in Open Source Projects

The React project released in 2013 is contributing in the field of web technology (Wikipedia, 2020b). It is another open source library of Facebook, which utilizes a JavaScript framework to help reduce complexity in developing user interfaces in the web development field (Facebook, 2020). This framework introduces an innovative take in viewing interchangable content, such as advertisements on websites, which were initially not fully welcomed by the community as it went against the best practises of JavaScript itself. Facebook faced technological issues with the general workflows and best practises in web development at that time. Although the problems were not unique to Facebook, it was them who took the charge and provided a solution, which not only benefitted themselves but others as well (Gackenheimer, 2015, p. 4). React was thus created for a single reason of solving how a website is dealing with displaying data (Gackenheimer, 2015, p. 4).

3 Methodology

In this chapter, we provide an overview about our steps of investigation in the research design. We describe our literature search strategy, as well as our selection process. Additionally, we explain our case study choice and model of commercial objectives.

3.1 Research Design

At the beginning of the course lecture, we defined our main research question:
"Which reasons does the IT-company have to contribute to FLOSS projects?" (Q1).
Due to a lack of resources we need to limit our scope of research on this. Therefore, we specify our investigation and derived the sub-question:
"Is the FLOSS project criticially important for the company's business?" (Q2)

Our main goal is to provide reasons why companies invest into OSS projects, which leads to several challenges on investigations in this term paper. First, there is a broad range of intensive knowledge with regard on OSS projects, which makes it difficult to focus on a specific scope of technologies and OSS projects (Facebook, 2020). Further, as we decided to focus on two OSS projects as described in chapter 2.4, there are no possibilities to validate our theories with interviews based on experts opinion of the company Facebook due to a lack of time and human ressources. We decided on a data-driven approach, which included scientific literatures and OSS projects on Facebooks' GitHub repositories. We summarized our investigation in the following:

1. We defined the main research questions and derived a sub-question, which gave us a clearer focus on our research direction.
2. We chosed OSS projects of the IT-company Facebook as our case study subject. We made our first screen on their current projects and selected two projects.
3. Then, we conducted a systematic literature review to build a theoretical foundation of current investigations of related fields.
4. Based on theories found, we derived an overview of reasons why companies invest into OSS projects based on the model of Zhou et al. (2016, p. 8).

3.2 Literature Search Strategy

We conducted the systematic literature based on the guideline of Xiao & Watson (2019, pp. 102–108). First, we identified digital libraries ACM Digital Library, IEEE Digital Library and Springer Link and evaluate them as suitable for our literature review. These scientific databases possess a broad repository for computer science research (Kaur et al., 2020, p. 4). We took ScienceDirect as an additional digital library. We summarized our steps in Figure 2.
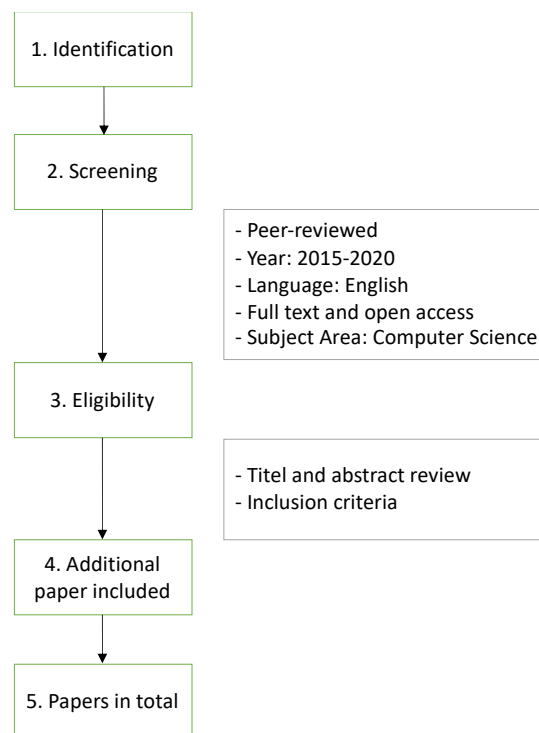
Figure 2 - Selection Process

We build the key word search string based on our first literature screen and subjective perception: *("FLOSS" OR "Open Source Software") AND ("company" OR "enterprise") AND "Facebook"*. We applied this search string in ScienceDirect, ACM Digital Library, SpingerLink and IEEE Digital Library. The previous presented search strings is suitable because of the scope of literature and its content we collected. We screened the paper by applying predefined exclusion and inclusion criteria. We restricted the search for papers to peer-reviewed, open access and free access, journals, English language and from the year 2015 to 2020. According to Kaur et al. (2020, p. 4), it is important to filter paper with the focus on the field of Computer Science to gain a better result. Lastly, by reading the title and abstract we checked for the egibility of those papers,

which left us 6 papers. After we search for other paper manually, we did not include additional paper to our collection. Finally, we found 6 papers in total.

Literature Review				
Key word search string: ("FLOSS" OR "Open Source Software") AND ("company" OR "enterprise") AND "Facebook"				
1. Identification	**ScienceDirect** n=605	**ACM** n=331	**IEEE** n = 22	**Springer** n = 44
2. Screening				
2.1 Peer-Reviewed	n=25	Only journals: n=35	/	/
2.2 Year: 2015-2020	n=21	n=28	n = 11	n = 28
2.3 Language: English	/	/	/	n = 26
2.4 Full text or Open Access	n=7	Media format: PDF n=28	/	N=20
2.5 Subject Area: Computer Science	n=4	/	/	N=11
3. Eligibility				
3.1 Titel and abstract reviewed	n=1	n=2	n=1	n=2
3.2 Inclusion criteria	/	/	/	/
4. Additional Papers Included	n=0	n=0	n=0	n=0
Papers in total	n=1	n=2	n=1	n=2

Table 3-1 - Literature Review Overview

3.3 Case Study Analysis

We built and conducted a case study based on Facebooks' OSS projects to get deeper insights into the reasons for companies to invest into OSS projects. We focus on two main IT technologies: AI und web technologies. Further, these two technologies have a broad research field, thus we put the focus on two specific cases. By limiting to two OSS projects of Facebook, we choose the AI project *PyTorch* and the of frontend web technology project *React* (Facebook, 2020). We described them in section 2.4. First, we collected data such as software version, project repositories and contribution histories on GitHub. The data collection can be found in the appendix Table 5-1 to Table 5-3. After that, we applied the OSS projects on the model "Dimensions of the community involvement models" based on Zhou et. al (p. 8, 2016). We considered this approach as a good construct to get a deeper insights into the topic and to answer our

research question. We report the results in chapter 4.1.2. We picked these two projects as they at first seemingly share the common objective to help developers improving on their capability to operate efficiently in their respective fields. We use different source to retrieve additional information such as Facebook Developer blogposts (Jia, 2018, p. 1). In

Table 5-1, we collected data on the top 10 contributors of React and PyTorch listed in the GitHub statistics. We identified the user name of the top 10 contributors automatically listed on the statistics extracted from the OSS projects on GitHub insights. Next, we identified the occupation of contributors which can be easily found in GitHub profile or LinkedIn. Reacts' top 10 contributors make up half of the commitments (6893 commitments, 50%). Even though there are 5.259.696 users and 1.527 contributors, only a few of people contributes a significant amount of commits on the React OSS project presented in Table 5-2 and Table 5-3. In comparison to React, PyTorch has 5,1 million less users, but PyTorch has more than twice as many commits and has a similar number of contributors.

3.4 Model of Commercial Objectives

The model "Dimensions of the community involvement models" developed by Zhou et al., 2016 (pp. 7–8) was described in chapter 2.2 lays our foundation to analyze and understand our literature and case study more in detail. We limit the model to the dimension "Commercial Objectives" and we derived a table construct in the following Table 3-2, in which we classify our results into five concepts of commercial objectives. We categorize our papers and case study in seven columns: Paper title, author, year and five concepts from literature. According to Zhou et al. (2016, p. 10), we marked records in following two tables if the paper and Facebook OSS projects include the information related commercial objectives defined in Table 2-1 and Table 2-2.

Literature		Commercial Objectives				
		Concepts from Literature				
Paper title	Author (year)	C1	C2	C3	C4	C5
[1] Paper title	Author A (year)	x			x	
[1] Paper title	Author B (year)		x			
[1] Paper title	Author C (year)			x		x

Table 3-2 - Literature Review Construct

4 Results

4.1 Findings

In this chapter, we provide and described our findings of the literature review and the case study analysis. The results are visualized in two overviews summarized in two tables. After that, we discuss critically about the findings. Lastly, we summarize the results, as well as our approach throughout the paper.

4.1.1 Overview of Reasons for Companies to Invest into FLOSS Projects

We found 6 scientific papers, which investigate on open source projects in companies. After screening and analyzing the content of the papers, we have listed the papers with their title, author and year of each paper in the following Table 4-1.

Then, we classified them into five concepts: C1. Software Quality, C2. Outstanding Technical Support, C3. Business Opportunity, C4. Business Model and C5. External Innovation. The following Table 4-1 provides an overview of our findings:

Literature Review		Commercial Objectives				
		Concepts from Literature				
Paper titel	Author (Year)	C1	C2	C3	C4	C5
[1] Understanding community participation and engagement in open source software Projects: A systematic mapping study	Rajdeep Kaur, Kuljit Kaur Chahal, Munish Saini, 2020	x				
[2] Eight Observations and 24 Research Questions About Open Source Projects: Illuminating New Realities	Matt Germonprez, Georg J.P. Link, Kevin Lumbard, Sean Goggins (2018)	x	x	x	x	
[3] Digital Democracy: Episode IV - A New Hope*: How a Corporation for Public Software Could Transform Digital Engagement for Government and Civil Society	John Gastil, Todd Davies (2020)			x	x	

		C1	C2	C3	C4	C5
[4] Moving from Closed to Open Source: Observations from Six Transitioned Projects to GitHub	Pavneet Singh Kochhar, Eirini Kalliamvakou, Nachiappan Nagappan, Thomas Zimmermann, Christian Bird (2019)	x	x	x	x	x
[5] Fifteen Years of Open Source Software Evolution	Francis Bordeleau, Paulo Meirelles, Alberto Sillitti (2019)	x		x	x	
[6] Open Source Technology Development	Kevin Crowston (2016)	x	x			
Sum		5	3	4	4	1
Legend C1: Software Quality C2: Outstanding Technical Support C3: Business Opportunity C4: Business Model C5: External Innovation						

Table 4-1 - Overview of Reasons for Companies to Invest into OSS Projects

C1: Software Quality:

According to Kaur et al. (2020, p. 12) and Germonprez et al., (2018, p. 13), the importance of a healthy and active community is essential to software quality. To promote and maintain software quality, Kaur et al. (2020, p. 12) state that the different stakeholders need to be addressed individually. For example, communities practically raise junior developers by giving helpful answers.

Whereas Germonprez et al. (2018, p. 13) describe that also senior contributors must given due respect. Addressing different stakeholders serves to reduce fluctation, show appreciation of the work and ultimately ensure software quality. These findings are mirrored in the research results of Kochhar et al. (2019, p. 9) and Croswton (2016, p. 480), in which they claimed that a large community greatly reduces bugs and promotes software quality more.

In a survey conducted by Kochhar et al. (2019, p. 9), they found that 58% of the interviewees agreed strongly that software quality increases by open sourcing projects. They replied that the code base usually already follows high coding standards, but external developers still add value to the projects as the internal company developers

only have limited time. Thus, the voluntaries can develop their coding expertise by taking over miniscule tasks and taking part in code clean ups resulting in better software quality. These findings can be summarized as Linux's Law, which states with more developers there will be less severe bugs, as no bugs need to be ranked due to the sheer number of developers.

C2: Outstanding Technical Support:

Germonprez et al. (2018, p. 2) investigated on open source project ecosystem of the Linux Foundation as they describe that foundation support play a major role in the evolution of open source projects. This has been validated by interviews taken by Kochhar et al. (2019, p. 9) where he assessed that outstanding technical support is critical for the stabilty of the OSS project due to the large number of users deploying its software on various platforms.

Based on the reducing fluctations argument in C1, Crowston (2016, p. 480) strengthened the viewpoint that a larger vibrant community, implicating sound suppport, is leading to better sofware quality and vice versa. Because good software enjoyed by many users tends to receive developer attention and ensues a growing community, which further heightens the software quality, satisfies users and attracts more developers. This feedback-loop through outstanding technical support grants OSS projects reliabilty and robustness (Crowston, 2016, p. 480).

C3: Business Opportunity:

According to Germonprez et al., 2018 (pp. 14–15), open source projects demand secure long-term investment. There is an urgent need in long-term investment in innovative market mechanism to coordinate the development of open source projects (Germonprez et al., 2018, pp. 14–15). This can be, for example through crowd marketing or funding campaign. Further, these authors indicated that companies need to align the use of OSS and internal innovation process. This means that companies have to perform internal actions to deal with open source project health risks (Germonprez et al., 2018, p. 18). We have assigned this paper to C3, because the authors described that companies have to set a strategic goal and to approach a long-term investment to deal with future risks.

To make use of ones' business opportunities, Wikipedia set in the early era of social media platform a strategic goal to keep their platform open (Gastil & Davies, 2020, p.

5). This approach provides access in order to increase their company size as mentioned by Gastil & Davies (2020, p. 5). Also, Kochhar et al. (2019, p. 6) explained that the connections between projects within a firm can already lead to synergy. In summary, it promotes success in the company when business opportunities are clearly identified and strategically exploited.

C4: Business Model:

Since OSS projects become increasingly commercialized by large organizations, the traditional view of OSS projects is moving more and more into the background (Germonprez et al., 2018, p. 11). In the present, there exists a shift in utilizing OSS projects as business models for revenue (Bordeleau et al., 2019, p. 65). As the projects' data becomes more subject to ownership, OSS projects tend to lose their advantages of providing data transparently and projects provide less incentive to develop due to lack of data for necessary use cases (Bordeleau et al., 2019, p. 65). From the perspective to build a business model, Germonprez et al. (2018, p. 11) argued that the intention to make profit from these projects should primarly serve the company's supply chain. This means that companies should not keep the OSS projects as the "social constructions of freedom to the forefront" as it was traditionally (Germonprez et al., 2018, p. 11).

Further, there is a huge competion of how to make the most effective use of open source resources. For example, Gastil & Davies (2020, p. 5) described that Facebook is a succesful business model because it stands out comparing to other companies such as Friendster or MySpace. Consequently, Facebook won loyal users and the most skilled programmers (Gastil & Davies, 2020, p. 5). These successes were achieved, especially, at the beginning of Facebook's early stages. For example, Facebook put the relationship aspects of people's real life in the foreground, rolled ads slowlier, used venture capital to increase community participation and focused on meeting the needs of the community (Gastil & Davies, 2020, p. 5). We can say that Facebook used its resources efficiently, aligned its strategy with the communities' requirements and attracted users, which at the same time, are the ground and success of Facebook.

C5: External Innovation:

New contributors involvement changes the OSS projects dynamics by gaining more ideas and skills while solving technical issues and thus, enhance innovation (Kaur et al., 2020, p. 12). However, we did not cross "C5: External Innovation" for most of the paper because we search for more specific information on how smaller companies benefit from innovation of greater companies, which was missing here.

4.1.2 Case Study: Reasons for Facebook to Invest into OSS Projects

As in the previous approach, we analyzed the OSS project React and PyTorch, and categorize them into the five concepts. To evaluate these projects, we need another definition of the five concepts C1 to C5 in order to suit the OSS project requirements as defined in Table 2-2. A result overview is provided in the following Table 4-2.

Case Study		Commercial Objectives				
		Concepts from Literature				
Open Source Project	**Company (Year)**	**C1**	**C2**	**C3**	**C4**	**C5**
[1] React	Facebook (2013)	x		x		
[2] PyTorch	Facebook (2016)	x		x		x
Sum		2	0	2	0	1

Table 4-2- Case Study Overview of Commercial Objectives

C1 Software Quality:

According to PyTorch (2020), the latest version 1.7.0 release was on the 27th of October 2020. In comparison, the previous PyTorch version 1.6.0 was announced on the 28th of July 2020. The span between the release dates is only 4 months, which is an implication that software often needs to meet new quality requirements. Comparing to React, after the initial public release of version 0.3.0 on 29th of May 2013, it reached its first orderly version by moving the decimal point past the zero reaching version on 15th of April 2016 (Wikipedia, 2020b). Since then it grew into version 17.0.1 with multiple versions months apart inbetween. We estimate that both software, especially React, are widely adopted and the projects has enjoyed continuing developments to keep software quality.

C2: Outside technical support:

For PyTorch, we found out that 8 of the top 10 contributors where employed in Facebook as presented in

Table 5-1. One contributor is employed at Google but was a former employee at Facebook. The second ranked contributor was not found on social media platforms. We assume that all of the top 10 contributors have a strong relation with Facebook regardless of time. We find similarities with the React project as well. In

Table 5-1 the top 10 React contributors are identified to be Facebook employees and only one has left the company since.

C3: Business Opportunity:

React was created to intend solving unique problems of Facebook (Hámori, 2018, p. 1). Further, React is also widely used to build a business model for other companies, which utilize React.js for example, to build strong web and mobil applications (Karczewski, 2020, p. 1). In comparison, PyTorch gained partially outside technical support from the OSS community at its development start-up phase (Lorica, 2017, p. 1). PyTorch, however, was not directly intended to solve Facebooks' problems since it was created out of many independent solutions within the OSS community (Lorica, 2017, p. 1). Therefore, we can say that React clearly embraces business opportunity for Facebook while PyTorch is used as an extension to solve Facebooks' problems using deep learning techniques.

C4: Business Modell:

According to Zhou et al. (2016, p. 8), an OSS product is critically important for the company's business when it gains direct profit from the OSS product or have a strong association with profit or receiving help to gain profit. There is no evidence found that PyTorch is critically important for the company's business gaining a direct profit from this OSS product. However, PyTorch is partnered with large enterprises such as Tesla, Amazon and Uber (Wikipedia, 2020a). This is an indication that PyTorch is associated with great profit. Similarly, we derive the same results for the React project as React has a strong association with receiving help to gain profit. Several major websites utilize the React framework such as New York Times, PayPal, Netflix, Uber and Airbnb. In addition to the high popularity of React, companies can save costs being able to use

high-performance software to achieve an optimal user experience and interaction with the website in question (Karczewski, 2020, p. 1).

C5: External Innovation:

React provides external innovation to other companies. For React, we outlined in C3 that many major websites are based on React tools. We can deduct by screening through GitHub repositories and our findings, that web developers are heavily relying on the OSS project initiated by Facebook. However, Facebook does not directly benefit from external innovation provided by other companies. Facebook itself experienced an internal innovation because "Facebook Ads became hard to manage, so Facebook needed to come up with a good solution for it. Jordan Walke worked on the prototype and created React." (Hámori, 2018, p. 1). Comparing to React, PyTorch became the successor of the Torch scientific computing framework and was inspired by primarly three frameworks. PyTorch gained and combined external innovation from the Torch community, from certain Twitter researchers and the from the Python community (Lorica, 2017, p. 1). Therefore, we decided that React has never received external innovation to be successful but PyTorch does in fact received ideas and solutions, especially, at its development start-up phase.

4.2 Discussion

In this section, we discuss about the previous findings and try to answer the questions:

"Which reasons does the IT-company have to contribute to FLOSS projects?" (Q1)

"Is the FLOSS project critically important for the company's business?" (Q2)

Through our research we have found answers to our two research questions. Our investigation shows that *Software Quality* (C1) followed by *Business Opportunity* (C3) are the most common reasons for companies to invest into OSS projects as presented in Table 4-1. From our literature review, we identified that companies have realized the importance of a healthy and active community, which is essential to software quality. The development experience of the respective contributors comes hand in hand with software quality. To guarantee software quality, OSS projects need to maintain the expertise of experienced developers. To gather new ideas, the support of rather inexperienced developers needs to be done by giving feedbacks and support from the senior developers on their work. This is important since these interactions vivid the OSS community and strenghen their collaborations. This, in turn, promotes active participation in OSS projects and keep software quality in a steady improvement cycle. From the case study review, we have identified constant updated software versions of React and PyTorch. The numbers needs to be investigated more in detail. Since software quality has many metrics that can be measured and we only measured timeliness of the OSS projects, more research is needed in the area of software quality as well. We surely can say that software quality is currently in a constant improvement process, which demands from the developers to meet certain software quality standards. Even though, there is a relatively huge community of 1.527 contributors in React and 1.699 contributors in PyTorch, the top 10 contributors made up 24% to 50% of the contributions of these projects in total. This is shown in the appendix Table 5-2 and Table 5-3. The significance of the actual distribution of experience and inexperienced contributors needs more investigations to measure how software quality was actually improved by the contributions of the OSS community. Also, the difference between employed and non-employees of Facebook can play a role in software quality as the top 10 contributors have a strong association with the company Facebook as indicated in Table 5-1 and thus, influence the behavior of each individuals.

As we approach the second research question Q2, we find answers in our research. With regard to the second reason, *Business Opportunity*, we extracted from literature review that OSS is leading to potential new revenue. For example, we find that users indirectly drive other companies' growth when they rely on the respective products from other companies' OSS projects. In general, companies can take advantage of this business opportunity by aligning their business goals with internal practices by promoting OSS community participation. This also comes with responsibilities, as it is important to maintain the health of the OSS community. Looking at Facebook, firstly, we see that this company does not directly profit from the OSS project, as we did not find any revenue streams in our research. Therefore, we conclude that OSS projects are not criticially important for the company Facebook. However, as we switch the perspective onto open source libraries React and PyTorch, we investigated on the association and support of other companies to gain profit. Here, we found strong associations of the OSS projects partnered with other companies as mentioned in section 4.1.2. These companies strongly rely on mentioned OSS libraries to offer their products and services. Most recent findings point at a shift away from software as a product towards as a service as mentioned in section 4.1.1 bottom of segment C4. Here, trends can be analyzed to give a better gauge on the state of OSS projects as a business tool. Further, a possibility remains to update the applied model to summarize the concepts Business Model and Business Opportunites, as related findings are difficult to distinguish.

We perceive the OSS community participation in Facebooks OSS projects React and PyTorch positively as they benefit from the huge number of participants. On the one side, developers are able to greatly improve their skills or gain reputation in the field, aiding to their interpersonal development. On the other hand, the product itself enjoys efficient development cycles as the large community hardly neglects minor bugs and larger strutural errors are circumvented due to mass deployment of all contributors not only by the core members of the project. However, it is necessary to put processes in place to naturally grow a healthy community in order to harness envsioned software quality. Since companies depend on the longevity and robustness of such projects, this is a task, that cannot be neglected. Albeit lacking the full list of companies' motivation, we find ourselves at valueable insights drafted from relevant papers, documents, company websites, repositories and blogs to explain company's action of OSS participation.

4.3 Conclusion

In this term paper, we found reasons for companies to invest in OSS projects. We addressed companies' engagement with OSS projects by taking a closer look at Facebooks' OSS projects, PyTorch and React, as our case studies. We can see that Facebook is actively engaged in OSS projects. We first note that OSS projects and companies' goals are closely related. Since the Facebook company provides important resources such as investment in form of employees (core contributors) for its OSS projects, this provide a basis for a healthy OSS community.

Our research method is as follows. We conducted a literature search by relying on four academic literature databases to find relevant papers to answer our research question. To identify credible results, we adapted an existing model of Zhou et al. (2016, pp. 7-8) to fit our needs. This model covers five key concepts regarding reasons of OSS engagement by companies: Software Quality, Outstanding Technical Support, Business Opportunity, Business Model and External Innovation. We used this model to assess scientific literatures and our case studies. Next, we chose two OSS projects invested by Facebook in the field of AI and web technologies. Additionally, we have screened and collected repository data based on GitHub. We were able to narrow down the most important reasons for a company to engage in OSS and answer the research question at hand. We find that companies highly value OSS communities as they can improve 1. *Software Quality* and create 2. *Business Opportunities*. The first is due to more users get in touch with the software developed and logically raising the quality of the code base, reducing complexity and enhancing its robustness. The second is due to synergy effects resulting, when companies engage in OSS projects largely concurring with their main business fields. Our results have shown that although companies follow their objectives to participate in the OSS communities, they still need to be encouraged to get more input from external developers within the OSS community. This can be seen from the fact that the selected OSS projects have many users, but still receive relatively little input from outside the OSS community. While there is transparent information about Facebook's OSS projects, there is a lack of scientific research and long-term studies. This paper gives a good overview of the commercial objectives and reasons for companies to invest in OSS projects. However, more research is needed to get a comprehensive view on this topic. In the future, more qualitative research should be conducted internally in companies, as well as quantitative assessments of OSS projects.

5 Appendix

Rank	GitHub User ID	Facebook employee	Non Facebook employee	Commits
PyTorch				
1	ezyang	x		1.456
2	gchanan	/	/	1.078
3	apaszke		x	870
4	jerryzh168	x		734
5	soumith	x		734
6	zdevito	/	/	623
7	Yangqing	x		601
8	colesbury	x		582
9	smessmer	x		552
10	zou3519	x		544
React				
1	gaearon	x		1.487
2	bvaughn	x		1.288
3	sophiebits	x		875
4	zpao	x		821
5	acdlite	x		778
6	sebmarkbage	x		731
7	trueadm	x		429
8	petehunt		x	205
9	chenglou	x		152
10	vjeux	x		140
URL (access date: 15.01.2021)				
PyTorch: https://GitHub.com/PyTorch/PyTorch/graphs/contributors				
React: https://GitHub.com/Facebook/React/graphs/contributors				

Table 5-1 - Top 10 Contributors of PyTorch and React

Start Year	Languages	%	Users	Contributors	Commits on Master Branch	Top 10 Contributor and Commits
2013	JavaScript	95,20%	5.250.696	1.527	13.775	6.893 (50%)
	HTML	2,00%				
	CSS	1,20%				
	C++	0,80%				
	TypeScript	0,30%				
	CoffeeScript	0,30%				
	Other	0,20%				

Table 5-2 - React OSS Project on GitHub (2020)

Start Year	Languages	%	Users	Contributors	Commits on Master Branch	Top 10 Contributor and Commits
2016	C++	52,80%	53.373	1.699	32.371	7.748 (24%)
	Python	33,70%				
	Cuda	6,00%				
	C	3,90%				
	Cmake	1,30%				
	Objective-C++	0,60%				
	Other	1,70%				

Table 5-3 - PyTorch OSS Project on GitHub (2020)

List of Literature

AltexSoft. (2017). *What Software Quality (Really) Is and the Metrics You Can Use to Measure It.* https://www.altexsoft.com/blog/engineering/what-software-quality-really-is-and-the-metrics-you-can-use-to-measure-it/

Ballhausen, M. (2019). Free and Open Source Software Licenses Explained. *Computer, 52*(6), 82–86. https://doi.org/10.1109/MC.2019.2907766

Bordeleau, F., Meirelles, P., & Sillitti, A. (2019). Fifteen Years of Open Source Software Evolution. In F. Bordeleau, A. Sillitti, P. Meirelles, & V. Lenarduzzi (Eds.), *Open Source Systems* (Vol. 556, pp. 61–67). Springer International Publishing. https://doi.org/10.1007/978-3-030-20883-7_6

Crowston, K. (2016). Open Source Technology Development. In W. S. Bainbridge & M. C. Roco (Eds.), *Handbook of Science and Technology Convergence* (pp. 475–486). Springer International Publishing. https://doi.org/10.1007/978-3-319-07052-0_29

Facebook. (2020). Open Source projects by Facebook. *Facebook Open Source.* https://opensource.facebook.com/projects

Flory, J. W. (2018). *A look inside Facebook's open source program.* https://opensource.com/article/18/1/inside-facebooks-open-source-program

Gackenheimer, C. (2015). *Introduction to React.* Apress. https://doi.org/10.1007/978-1-4842-1245-5

Gastil, J., & Davies, T. (2020). Digital Democracy: Episode IV—A New Hope*: How a Corporation for Public Software Could Transform Digital Engagement for Government and Civil Society. *Digital Government: Research and Practice, 1*(1), 1–15. https://doi.org/10.1145/3342194

Germonprez, M., Link, G. J. P., Lumbard, K., & Goggins, S. (2018). Eight Observations and 24 Research Questions About Open Source Projects: Illuminating New Realities. *Proceedings of the ACM on Human-Computer Interaction*, *2*(CSCW), 1–22. https://doi.org/10.1145/3274326

Hámori, F. (2018). *The History of React.js on a Timeline*. https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/

Jia, B. (2018). Announcing PyTorch 1.0 for both research and production. *Facebook for Developers*. https://developers.facebook.com/blog/post/2018/05/02/announcing-pytorch-1.0-for-research-production/

Karczewski, D. (2020). *21 Best Open Source React Developer Tools To Use In 2021*. https://www.ideamotive.co/blog/best-open-source-react-developer-tools

Kaur, R., Kaur Chahal, K., & Saini, M. (2020). Understanding community participation and engagement in open source software Projects: A systematic mapping study. *Journal of King Saud University - Computer and Information Sciences*, S1319157820305139. https://doi.org/10.1016/j.jksuci.2020.10.020

Kemp, R. (2010). Open source software (OSS) governance in the organisation. *Computer Law & Security Review*, *26*(3), 309–316. https://doi.org/10.1016/j.clsr.2010.01.008

Kochhar, P. S., Kalliamvakou, E., Nagappan, N., Zimmermann, T., & Bird, C. (2019). Moving from Closed to Open Source: Observations from Six Transitioned Projects to GitHub. *IEEE Transactions on Software Engineering*, 1–1. https://doi.org/10.1109/TSE.2019.2937025

Logothetis, S., & Stylianidis, E. (2016). BIM Open Source Software (OSS) for the documentation of cultural heritage. *Virtual Archaeology Review*, *7*(15), 28. https://doi.org/10.4995/var.2016.5864

Lorica, B. (2017). *Why AI and machine learning researchers are beginning to embrace PyTorch*. https://www.oreilly.com/radar/podcast/why-ai-and-machine-learning-researchers-are-beginning-to-embrace-pytorch/

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 12. https://papers.nips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

PyTorch. (2020). *PyTorch 1.6 released w/ Native AMP Support, Microsoft joins as maintainers for Windows*. https://pytorch.org/blog/pytorch-1.6-released/

Wagstrom, P. A. (2009). *Vertical Interaction in Open Software Engineering Communities*. https://patrick.wagstrom.net/thesis/wagstrom-phd-thesis-single.pdf

Wikipedia. (2020a). PyTorch. In *PyTorch*. https://en.wikipedia.org/wiki/PyTorch

Wikipedia. (2020b). *React (web framework)*. https://en.wikipedia.org/wiki/React_(web_framework)

Wikipedia. (2021). *Facebook*. https://en.wikipedia.org/wiki/Facebook

Xiao, Y., & Watson, M. (2019). Guidance on Conducting a Systematic Literature Review. *Journal of Planning Education and Research*, *39*(1), 93–112. https://doi.org/10.1177/0739456X17723971

Zhou, M., Mockus, A., Ma, X., Zhang, L., & Mei, H. (2016). Inflow and Retention in OSS Communities with Commercial Involvement: A Case Study of Three Hybrid Projects. *ACM Transactions on Software Engineering and Methodology*, *25*(2), 1–29. https://doi.org/10.1145/2876443