Paper to be presented at the DRUID 2012

on

June 19 to June 21

at

CBS, Copenhagen, Denmark,

Open Source Software and Open Content Comparison: Same Same but Different

Daniel Ehls
Hamburg University of Technology
Institute for Technology and Innovation Management
Daniel.Ehls@tu-harburg.de

Elda Dedja
Hamburg University of Technology
Institute for Technology and Innovation Management
elda.dedja@gmail.com

Cornelius Herstatt
Hamburg University of Technology
Institute for Technology and Innovation Management
c.herstatt@tu-harburg.de

Abstract

The advancement in Information and Communication Technologies has facilitated a new form of large scale collaborative production: Open Source. Along with Open Source Software, that is the most successful example of this type of production, this phenomenon is expanding beyond the software industry, such as content creation. In this paper, we explore the similarities and differences among Open Source Software and Open Content by a structured literature review and a detailed comparison framework. We find differences in terms of involved actors and the innovative product outcome, but also similarities in terms of task and process. In our analysis, the involved contributors and their intensions

differ meaningfully, indicating a more reward driven orientation within OSS, whereas Task and Process result being very similar indicating comparable organizational set-ups. Based on our findings, we derive implications of the comparison and expose further research avenues.

Jelcodes:M19,Z0

Open Source Software and Open Content Comparison: Same Same but Different

Abstract

The advancement in Information and Communication Technologies has facilitated a new form of large scale collaborative production: Open Source. Along with Open Source Software, that is the most successful example of this type of production, this phenomenon is expanding beyond the software industry, such as content creation. In this paper, we explore the similarities and differences among Open Source Software and Open Content by a structured literature review and a detailed comparison framework. We find differences in terms of involved actors and the innovative product outcome, but also similarities in terms of task and process. In our analysis, the involved contributors and their intensions differ meaningfully, indicating a more reward driven orientation within OSS, whereas Task and Process result being very similar indicating comparable organizational set-ups. Based on our findings, we derive implications of the comparison and expose further research avenues.

Open Source Software and Open Content Comparison: Same Same but Different

Open Source Software (OSS hereafter) is a growing phenomenon where geographically distributed volunteers working together developing software in a collaborative way (Lerner & Tirole, 2002; von Hippel & von Krogh, 2003; Pénin, 2008; Lakhani & Wolf, 2005). Big success stories of this type of collaboration like Linux, Open Office, VLC Media Player or Mozilla, have become part of today's most demanded software and have been widely analyzed by scholars. First studies have investigated about "why should thousands of top-notch programmers contribute freely to the provision of a public good" (Lerner & Tirole, 2002) and their motives for participation (e.g. Roberts et al., 2006; Lakhani & Hippel, 2003, Hars & Ou, 2001). Additional studies analyze governmental structures (Shah, 2006; West & O'Mahony, 2005), as well as competitive dynamics and firm interactions like "men on the inside" (Bonaccorsi & Rossi, 2003; Dahlander & Wallin, 2006; Lee, 2011). Further studies from Crowston et al. (2010), Zwass (2010), von Krogh & von Hippel (2006) provide a more general OSS specific overview analysis.

Apart from software, we are witnessing the extension of this type of collaborative product creation in other digital goods such as text, music, video etc., commonly identified among scholars as Open Content (Cheliotis, 2009; Okoli, 2009). The most well known example of the Open Content (OC hereafter) production model is Wikipedia (Cheliotis, 2009) besides further popular examples like LibriVox, ccMixter or Open Street Map. This newly emerging trend is receiving a lot of attention in the press and scholar environments (Okoli, 2009). However, the scholarly work studying aspects of this phenomenon, conversely to OSS, is still in its nascent phase, with studies limited in total amount and often in isolated examples.

Moreover, to the best of our knowledge, there are no studies that compare uniting and differencing aspects of OSS and OC. Following this research gap, the purpose of this paper is, first, to shed light on the underlying concepts of OSS and OC, and, second, to find similarities and divergences among the two phenomena. In order to do so, we develop a framework consisting of four major clusters and an overall of 13 elements, according to which the two Open Source streams are compared. Following this structured approach, we identify gaps in the OC as well as in the OSS field, which can serve as future research avenues in this field.

This paper is organized as follows: Section 2 introduces the Open Source taxonomy and provides definitions of OSS and OC. In the following section, we explain the methodology and the comparison framework used for our study. In section 4, we present the results of the comparison between OSS and OC and lastly, after identifying gaps in the literature, we draw conclusions and implications for future research.

2. Open Source Taxonomy

Defining Open Source as well as OSS and OC is a challenging task as no common definition and categorization is yet agreed. Various scholars present their own definitions, focusing on different concepts with partly overlapping scope. For von Krogh & von Hippel (2003) the term Open Source stands for the type of license applied to the product created. On the other hand, Nov & Kuk (2008) extend this definition by stating that Open Source is a model based on the collaboration of people, who contribute their time and knowledge to create digital goods that are released to the public (Nov & Kuk, 2008). In this study, we focus on two main types of digital co-creation: OSS and OC. The borders among these two concepts are fuzzy, where some scholars define them as overlapping and others as exclusive.

In the first group of studies, both OSS and OC are part of the same major phenomena, under the name of Information Goods (Cheliotis, 2009) or Open Content in a wider sense (Okoli, 2009). Cheliotis (2009) and Okoli (2009) categorize the digital products into:

- Functional/ Objective goods (software, texts, encyclopedias, scientific publication)
- Cultural/ Artistic goods (videos, music, poetry, fiction)

In this scheme, OSS is a subcategory of functional/objective goods (figure 1).

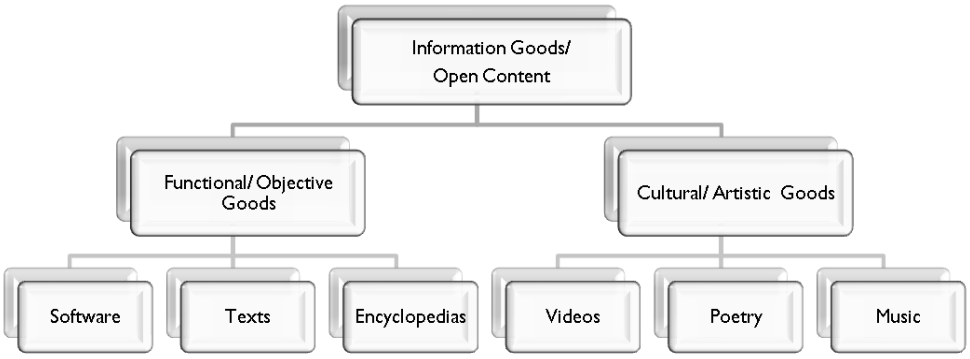

Figure 1: Information goods categorization[1]

However, for Nov & Kuk (2008) and Cedergren (2003), software is a separate category of digital goods[2]. Raymond (2002) emphasizes the difference among software and other types of content, as he states: "Music and most books are not like software, because they don't generally need to be debugged or maintained" (Raymond, 2002). Furthermore, the content-software separation is applied due to the

[1] Source: Self-construction, modeled after Cheliotis (2009), P.229 and Okoli (2009), P.2

[2] Zwass (2010) additionally classifies software in a separate category namely, Procedural Content and every other type of user-created digital content as Declarative Content (text, audio, video etc.)

distinction in terms of whether the file is executable, meaning procedural, or not (Raymond, 2002; Cedergren, 2003; Rosenzweig, 2006; Nov & Kuk, 2008; Zwass, 2010). Thus, the second group of authors categorizes digital goods in OSS – executable files, and OC – body of knowledge (encyclopedias, multimedia, etc.). This second approach of categorization is shown in figure 2.

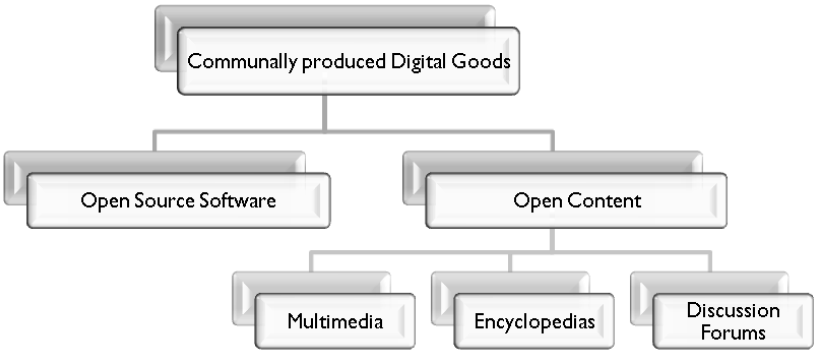

Figure 2: Digital Goods Categorization[3]

In this paper, we follow the latter categorization[4] (Cedergren, 2003; Nov & Kuk, 2008; Raymond, 2002), where OSS means strictly software and OC represents all the other non-executable information goods, but both are united under the umbrella term "Digital Open Source" goods.

Open Source Software Definition

A common definition related to the term "Open Source Software" is not yet agreed. Some sources highlight the ideological aspects of freedom[5], while further sources focus on openness[6]. The "free" term is often criticized for its close meaning to freeware as software available free of charge. On the other hand "open software" is criticized of being not descriptive, as it is generally understand for only providing access to source code, without any reference to other important freedoms for the user (Free Software Foundation, 2010; Okoli, 2009). Thereafter, a third group of authors (Pénin, 2008; Ghosh et al., 2002; Dalle & David, 2005; Benussi, 2006) coined a term to group both concepts together: "Free/Libre/Open Source Software" (FLOSS). Thus:

[3] Source: Self-construction, modeled after (Cedergren, 2003; Nov & Kuk, 2008, P.2849; Raymond, 2002)

[4] This choice does not reflect a preference. However, an evaluation of degree of "cultural/ functional" aspects is not necessary and a clear distinction between OSS and OC for research is possible. Furthermore most research focuses on OSS besides other Open Source forms.

[5] For the software to be free, it should respect four user's essential freedoms to run, study, change and distribute it (e.g. Free Software Foundation, 2010; von Hippel, 2001; Glott et al., 2007; Berlecon, 2002)

[6] "Open Source Software" is software which allows the developers to access the source code, modify and distribute it under the same license as the original software (Lerner, 2001; Open Source Initiative, 2010; Crowston, Annabi, & J. Howison, 2003; Koch & Schneider (2002) and Rossi (2004).

> FLOSS is software, which is distributed via an "open" license and allows the access to its source code. Hence the type of the final product is the program (executable file) together with its source code and the license applied.

This understanding is also followed in this study, but for reasons of readability we will refer to it as OSS.

Open Source Content Definition

Open Content is understood as "body of digitizable knowledge" (Oreg & Nov, 2008), consisting of information products (apart from software) such as text, image, sound, video and combinations of them (multi-media) (Wunsch-Vincent & Vickery, 2007; Cedergren, 2003; Clarke, 2004;Oreg & Nov, 2008). However, the ambiguity of "free" and "open" has been transferred in the creation of digital goods beyond software. According to the Free Software Foundation (2010), the definition of "Free Software" can be extended to any kind of work. Therefore, "Free Content" (or sometimes termed as free cultural works) is any work or expression which can be used, studied, modified and distributed by anyone, for any purpose (Möller, 2008). In contrast, Liang (2007) and Preston & Zurer (2003) define Open Content as all the material as text, sound, images that any user can use, distribute and modify, without the restrictions of the copyright.

However, to avoid the ideology discussion and in analogy to FLOSS we define Open Content as:

> Open Content is the body of digitizable knowledge, which is distributed via an "open" license and allows the access to its source code and instructions.

However, both, the OSS and OC, definitions, lack of explicit mentioning of the communal model of developing, for which open source has received most of the attention from scholars. To authors such as Lerner & Tirole (2002), von Hippel & von Krogh (2003), Pénin (2008), Lakhani & Wolf (2005) and Zwass (2010), a major aspect of the open source phenomenon is the collaborative way of developing the software from a large number of distributed members of a community. These open collaboration innovation projects mean, several contributors share knowledge openly/freely with each other during the development process (Raasch, Herstatt, & Abdelkafi, 2008) and reveal openly the output of the process (Baldwin & Hippel, 2011). However, the communal mode is only a potential option in the production; highlighting the broad opportunities open access provides but not necessarily need to take place[7]. Hence, accessibility determines the degree to which external contributors could influence this production (West & O'Mahony, 2008) and we follow Baldwin and von Hippel (2011) that many, but not all open source software projects have these collaborative characteristics and hence term this specialized form "Open Collaborative Innovation project".

To further contrast OSS as well as OC and to enrich the terms we utilize a comparison framework to find out similarities and differences. The next sections provide the methodology and introduce the comparison framework.

[7] Indeed, most open source projects follow a power law distribution or the cave model, as described later in our comparison model in the cluster 'actors'.

3. Methodology and Comparison Framework

Research on OSS includes a large body of literature, but conversely, the study of OC remains still a nascent research phenomenon. Therefore, it is necessary to study the main concepts belonging to OSS/OC, investigate the differences, find similarities, and hence provide insight about transferability of research findings. In order to do so, in the view of constructivism, we follow a qualitative approach consisting of explorative literature research (Creswell, 2009; Huff, 2008) and sort the sources according to our developed comparison framework.

Literature Collection

Our initial research activity has been a literature review spanning top-rated journals[8], books, surveys in the relevant disciplines of technology, information, management, organizational science and other related fields based on a set of predefined keywords, and leading researchers, chosen following expert interviews and number of cites, as shown in figure 3 and 4.

Initial Keywords
peer production
co-creation
community innovation
user innovation
commons based production
open source software
open content

Figure 3: Initial Keywords

Initial Authors
E.von Hippel
G. von Krogh
K. Lakhani
S.Shah
J.Lerner

Figure 4: Initial Authors

[8]The ranking of the journals was found by using the Journal Citation Reports Tool provided by ISI Web of Knowledge. A short list of the journal rankings is available upon request from the authors.

In order to extend our collection of scholarly work, we used a two-fold approach:

I. Snowball approach: We collected additional important terms and keywords during the analysis of the first set of scholar work. For every new relevant keyword, a new search in the scholar databases was conducted.
II. Bibliographical Trailing: In the first set of papers, backward citations were followed to reach to important additional studies in the field.

The initial set of papers consisted of 41 papers found by using the initial set of keywords and the main authors in the field. After our two fold approach, our set of sources consisted of a total of around 100 studies[9]. The literature consulted is cited as appropriate, but in the interest of readability, only the works essential to the arguments are stated throughout the paper.

Comparison Framework

Considering the large number of concepts of open source phenomena found in the literature review phase, the need for using a common comparison framework has emerged. Zwass's 'taxonomy of co-creation' (2010) and the 'Open Source Innovation Model' introduced by Raasch et al. (2009) provide a suitable framework. Both frameworks reflect an open source innovation or open collaborative innovation process, which consists of open standards and voluntarily collaboration among actors involved in the process (Pénin, 2011; Raasch et al., 2008) and they exhibit a clustering of salient aspects of the open source phenomena. For a holistic view we combined the two research themes. This new framework, which we name 'Zwass-Raasch Comparison Framework' (figure 5), serves as our groundwork to compare the concepts of OSS and OC by four main clusters:

- *Actors Involved:* Actors involved in the development of an OSS/OC project are the most important aspect in these projects. The characteristics studied are the amount of participants involved, their type (individuals or companies), their diversity (age, sex, and nationality), their skills and their motivations to participate in an OSS/OC project.
- *Task:* The tasks undertaken by the communities of OSS/OC emerged from Zwass' taxonomy, where its main characteristics to study are: the modularity of task, effort intensity, and the barriers to participation. Additionally, the cost of development is considered.
- *Process:* This element is common in both of the selected frameworks. The main characteristics we look at in this study are the duration and the evolution (phases) of the process, the governance in the community and the supporting technological tools for this process to take place.
- *Final Product:* This category combines the innovative outcome in the Open Source Innovation Model (Raasch, et al., 2009) and co-created value in Zwass's Taxonomy of Co-creation (2010). The main characteristics analyzed are: the digital product quality and the licenses applied for the final product.

[9] Our literature collection considers papers up to July 2011

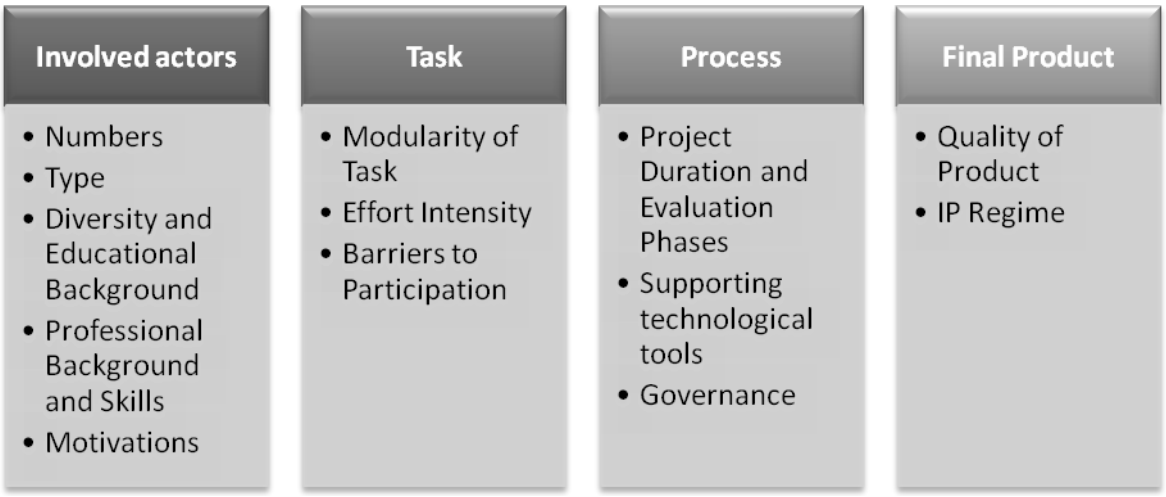

Figure 5: Zwass-Raasch Comparison Framework

4. OSS and OC Comparison

This section presents a comparative analysis of the concepts related to OSS and OC utilizing the four comparison framework clusters. Similarities and differences are discovered, and blank research sports are identified.

4.1. Involved Actors

The volunteer contributors of OSS and lately OC projects have puzzled the scholars for many years. Therefore, many studies have been carried out to learn who are these people and why do they spend their time and effort in such initiatives, as can be seen in figure 5.

Actors						
Source	Domain	Numbers	Type	Diversity and Education	Professional Background and Skills	Participation Motivs
Bryant et al. (2005)	OC					✓
Cedergren (2003)	OC	✓				✓
Chang & Yang (2009)	OC					✓
Glott, Schmidt, & Ghosh (2010)	OC			✓		✓
Muller-Seitz & Reger (2010)	OC					✓
Nov (2007)	OC			✓		✓
Nov, Naaman, & Ye (2010)	OC					✓
OECD (2007)	OC					✓
Rettberg (2005)	OC				✓	
Schroer & Hertel (2009)	OC			✓	✓	✓
Wikipedia (2011)	OC	✓	✓			
Wunsch-Vincent & Vickery (2007)	OC				✓	✓
Youcheng & Fesenmaier (2003)	OC					✓
Comino, Manenti, & Parisi (2007)	OSS	✓				
David et al. (2003)	OSS	✓		✓	✓	✓
Ghosh & Prakash (2000)	OSS	✓				
Ghosh (2006)	OSS		✓			
Ghosh (2007)	OSS		✓			
Ghosh et al. (2002)	OSS			✓	✓	✓
Hars & Ou (2001)	OSS			✓		✓
Henkel (2004)	OSS		✓		✓	
Hertel et al. (2003)	OSS					✓
Krishnamurthy (2002)	OSS	✓	✓			
Lakhani & Panetta (2007)	OSS					✓
Lakhani & Wolf (2005)	OSS			✓	✓	✓
Lerner & Tirole (2005a)	OSS		✓			
Markus, Manville, & Agres (2000)	OSS		✓			✓
Moon & Sproull (2008)	OSS	✓				
Osterloh & Rota (2007)	OSS					✓
Raymond (1999)	OSS					✓
Shah (2006)	OSS					✓
Torvalds (1998)	OSS					✓
von Hippel (2001)	OSS					✓
Weiss (2005)	OSS	✓				
Lioubareva & Feledziak (2007)	OSS, OC	✓				
Oreg & Nov (2008)	OSS,OC				✓	✓

Figure 6: Actors Literature

Numbers

Open Source projects give the opportunity for various contributors to share their knowledge and voluntarily collaborate in these projects. Participation of sufficient contributors thereby is critical, as it facilitates activity, enables quality control by peer review and finally attracts more contributors as well as it enables the projects to grow. Several studies claim a large amount of developers involved in the OSS communities, such as in Debian (Lioubareva & Feledziak, 2007) or in the Linux Kernel (Moon & Sproull, 2008), each with more than 1000 participants; furthermore the average contributor amount centers

around 325 (David et al., 2003). However, in many OSS projects, it is very common to find the "cave"[10] model of production rather than the "community" model (Krishnamurthy, 2002). Several authors (Comino, Manenti, & Parisi, 2007; David, Waterman, & Arora, 2003; Ghosh & Prakash, 2000; Krishnamurthy, 2002; Weiss, 2005) confirm this finding by stating that the majority of OSS projects are developed in limited groups of developers. Comino et al. (2007) find in a SourgeForce.net study, that projects which consist of more than 16 developers account for less than 1% of their sample. Weiss (2005) also finds out that almost 58000 projects in SourgeForge.net have only one developer, and only 5 projects out of the sample have more than 40 developers. Contrasting these findings, surveyed projects within the first group are well-known and established - probably iconic projects. These circumstances may affect the higher number of participation, while also support the cave model and following a power law distribution.

Within the OC stream, well-known examples such as Wikipedia or the Open Directory Project consist of a enormous number of contributors (15 040 568 and 50.000 members respectively (Cedergren, 2003; Wikipedia, 2011). However, smaller projects such as 'Take the Bus', which is a collaborative platform for music creation, consists of only 30 artists (Lioubareva & Feledziak, 2007).

Type

Various studies have been conducted to understand the type of actors involved in OSS projects and many of them come to the same conclusion: The principal participants in OSS projects are individual contributors (Krishnamurthy, 2002; Lerner & Tirole, 2005; Markus, Manville, & Agres, 2000), for-profit companies (Ghosh, 2006; Henkel, 2004; Krishnamurthy, 2002) as well as universities and non-profit organizations (Ghosh, 2007; Ghosh, 2006; Krishnamurthy, 2002).

To our knowledge, there are no studies about the types of actors in OC projects. However, in the cases of Wikipedia and Open Directory Project, both projects are results of a massive collaboration of individuals. However, Wikipedia is supported by Wikimedia Foundation, Inc. which is a non-profit organization, whereas for example the "Open Directory Project" and the "Freebase" community are owned by commercial organizations (Netscape respectively Google). The participation involvement of commercial parties also in production is therefore likely; however, further studies are required into this aspect to reach further conclusions.

Diversity and Educational Background

Open source communities are a great example of diverse and self-selected people working together. Gender and age of OSS developers is according to Ghosh et al. (2002) between 14 and 73 years, with an average age of 27 years. OSS communities, however, represent themselves very homogeneously in terms of gender diversity with only approximately 1% of the sampled contributors are female. Similar observations are reported by David et al. (2003) and Lakhani & Wolf (2005). The average age of Wikipedians is 25-33 (Glott, Schmidt, & Ghosh, 2010; Nov, 2007; Schroer & Hertel, 2009) and the range of age is from 10 to 85 years (Glott, Schmidt, & Ghosh, 2010; Schroer & Hertel, 2009). In the case of Wikipedia, a higher percentage of female participation is observed.

[10]Individual developer that carry out OSS projects on their own, without the collaboration of others

People contributing to OSS are well educated in terms of educational background. Most participants hold an undergraduate degree followed by a master degree (Hars & Ou, 2001; David et al., 2003; Ghosh et al., 2002). The major contributions to Wikipedia come from actors with secondary education (33.66%) followed by undergraduates (25.99%) as well as Masters and PhD (23.04%) (Glott et al., 2010).

Professional Background and Skills

The majority of participants in OSS projects are highly experienced programmers with an average of 11-14 years practice in programming and 4-5 years experience in OSS programming (Henkel, 2004; Lakhani & Wolf, 2005)[11]. According to Ghosh et al. (2002), 83% of the participants are people working in the IT sector. The second bigger group embodies students, consisting of 16% of the sample. Furthermore, most OSS contributors are employed, followed by the second largest group consisting of students (David et al., 2003; Ghosh et al., 2002).

In Wikipedia, Oreg & Nov (2008) observe that common occupations include computer scientists, academic researchers and graduate students. However, these occupations might change when looking at further examples of content creation. For instance, in a music collaboration platform, other professional backgrounds can be expected such as musicians, composers etc. Concrete insights are not available yet. Similarly to software, the Wikipedia survey (Schroer & Hertel, 2009) finds that major parts of contributions come from employees (53%) and students (32%). Contrary to the software context, the contributors to OC are mostly amateurs (Oreg & Nov, 2008; Rettberg, 2005; Wunsch-Vincent & Vickery, 2007), who just like to contribute to the context creation activities. I.e. according to Benkler (2002), people that contribute to Wikipedia are just people that like to write. In most of the OC initiatives, anyone can contribute and there are no skill expectations for joining such communities (Oreg & Nov, 2008).

Motivations

Understanding why people volunteer and contribute their work for free was one of the key questions asked in open source research. To categorize open source motivations, we follow the taxonomy of human motivations in intrinsic and extrinsic motivations (Ryan & Deci, 2000). In terms of intrinsic motivations enjoyment during programming and altruistic feelings are two of the major motivations to contribute. Additionally, the belief that software should be free/ open is another important intrinsic motivation. The extrinsic motivations are more diverse and range from personal need to the desire of the developers to improve their programming skills, built reputation – whether among peers or to improve job opportunities. Additionally, a for profit intension is increasingly observed, including statements like: "It's part of my job', 'I make money out of my contribution', or 'I participate for professional reasons'. Figure 7 provides an overview of motivations[12].

[11] Lakhani & Wolf (2005) state that the average experience in programming is 11.86 years and OSS experience in 5.31 years, whereas Henkel (2004) finds out that the experience in programming is 14.2 and OSS experience is 4.9 years.

[12] A detailed analyzes particularly for OSS contribution motivations is forthcoming by von Krogh et al. (2012)

	Motivations	Lakhani & Wolf (2005)	Raymond (1999)	Hars & Ou (2001)	Oreg & Nov (2008)	Osterloh & Rota (2007)	von Hippel (2001)	Hertel et al. (2003)	Lakhani & Panetta (2007)	David, Waterman, & Arora (2003)	Shah (2006)	Markus et al. (2000)	Torvalds (1998)	Ghosh et al. (2002)
Intrinsic Motivations	Enjoy programming	✓	✓					✓	✓		✓	✓	✓	
	Believe that software should be open	✓								✓				✓
	Desire of feeling competence			✓										
	Altruism: the desire to help others in the community				✓					✓		✓		
	Share knowledge													✓
Extrinsic Motivations	User need	✓	✓	✓					✓	✓	✓	✓		
	Improve programming skills	✓		✓	✓	✓		✓	✓	✓				✓
	Reputation building		✓	✓	✓			✓	✓			✓		
	Make money and getting paid	✓		✓							✓			✓
	Improvement of job opportunities			✓					✓					✓
	Provide alternatives to proprietary software									✓				
	Improve OSS software							✓						✓
	Participate in a new form of cooperation													✓

Figure 7: OSS Motivations

In OC studies, similar motivations explain the participation of actors in OC communities: Altruism and task enjoyment are the preeminent intrinsic motives, while extrinsic motivations are rather weak. Figure 8 shows these findings.

	Motivations	Oreg & Nov (2008)	Muller-Seitz & Reger (2010)	Glott et al. (2010)	Chang & Yang (2009)	Wunsch-Vincent & Vickery (2007)	Bryant et al. (2005)	Schroer & Hertel (2009)	Nov, Naaman & Ye (2010)	Youcheng & Fesenmaier (2003)	OECD (2007)	Cedergren (2003)	Nov (2007)
		Wikipedia	Wikipedia	Wikipedia	Baidupedia	User-created Content	Wikipedia	Wikipedia	Flickr	Travel Community	Open Educational Resources	Wikipedia	Wikipedia
Intrinsic Motiations	Altruism	✓	✓	✓	✓		✓	✓		✓	✓	✓	✓
	Task enjoyment		✓		✓			✓	✓	✓	✓	✓	✓
	Share knowledge			✓				✓					
	Belief that information should be freely available			✓								✓	
	Commitment to community								✓				
Extrinsic Motivations	Reputation building	✓	✓			✓	✓		✓		✓	✓	
	Opportunity to learn and self-develop	✓		✓				✓	✓			✓	
	Fix errors			✓	✓								
	Connecting with peers in a virtual community					✓	✓					✓	
	Wikipedia is a better medium for encyclopedia			✓									
	Desire to express oneself					✓							
	Make money and getting paid			✓							✓		

Figure 8: OC Motivations

4.2. Task

The category task comparison includes three main aspects: modularity of task, effort intensity required for the completion of the task and barriers to participation as displayed in figure 9 in relation to authors.

Task				
Source	**Domain**	**Task Modularity**	**Effort Intensity**	**Participation Barriers**
Benkler & Nissenbaum (2006)	OC			✓
Benkler (2002)	OC	✓		
Muller-Seitz & Reger (2010)	OC			✓
Nov (2007)	OC		✓	
Schroer & Hertel (2009)	OC		✓	
Zhang & Zhu (2006)	OC			✓
Anthony, Smith, & Williamson (2007)	OSS			✓
Bonaccorsi & Rossi (2003a)	OSS	✓		
Giuri et al. (2006)	OSS	✓		
Ke & Zhang (2008)	OSS		✓	
Koch & Schneider (2002)	OSS		✓	
Kogut & Metiu (2001)	OSS	✓		
David et al. (2003)	OSS		✓	
Lakhani & Panetta (2007)	OSS			✓
Lakhani & Wolf (2005)	OSS		✓	
Osterloh, Rota, & Kuster (2002)	OSS	✓		
Raasch et al. (2008)	OSS			✓
Moon & Sproull (2008)	OSS	✓		✓
Osterloh & Rota (2007)	OSS			✓
von Hippel & von Krogh (2003)	OSS			✓
von Krogh & von Hippel (2003)	OSS			✓
von Krogh, Spaeth, & Lakhani (2003)	OSS			✓
Oreg & Nov (2008)	OSS,OC			✓

Figure 9: Task Literature

Modularity of Task

Modularity is the extreme form of task decomposition (Moon & Sproull, 2008) and refers to the breakdown of the task into small components, that can be produced independently and asynchronously (Benkler, 2002). This independent break down of the task allows contributors to choose what and when to contribute, maximizing their autonomy (Benkler, 2002) and making easier the parallel work of distributed contributors (Osterloh, Rota, & Kuster, 2002) hence fostering large-scale collaboration.

Researchers like Giuri et al. (2006), Kogut & Metiu (2001), Osterloh et al. (2002) and also practitioners like Linus Torvalds (in: Bonaccorsi & Rossi, 2003a) highlight the potential of software to be modularized as key success factor. Modularity allows OSS developers to code software modules in a parallel and independent modus, which increases the speed of development cycle (Kogut & Metiu, 2001) and the probability of high-quality solutions (Moon & Sproull, 2008).

Modularity in OC is presented by Benkler (2002) with two examples. In Wikipedia, the modularity is achieved through the wiki pages, which allow the distributed collaboration and effortless integration of independent contributions. The second example of modularization is the Distributed Proofreaders initiative. Distributed proofreaders check scanned book pages independently before submitting them to one single e-book.

Effort Intensity

Effort intensity is the amount of resources spent on the OSS/OC task (Ke & Zhang, 2008) measured as hours spent or code provided by individuals in an OSS/OC project.

In OSS projects records show that a programmer adds on average 21000 lines of code (Koch & Schneider, 2002) and spends between 14.3 (Lakhani & Wolf, 2005) and 11.4 hours/week (David, Waterman, & Arora, 2003). The majority of this time contributed goes to activities such as coding, debugging and designing algorithms (David et al., 2003).

The few studies on Wikipedia provide similar numbers. According to Schroer & Hertel (2009), the average time spent on Wikipedia is approximately 15.5 hours/week (133 min/day), even though the authors admit that their sample has a high number of administrators leading to a high average time spent. In a second study the average time spent contributing in Wikipedia is 8.27 hours/week (Nov, 2007).

Barriers to Participation

As the internet and web IT technologies are becoming widely spread and easily accessible, more people face lower barriers for participating in distributed forms of production like open source project (Zwass, 2010). However research points to necessary technical skills, joining processes and costs of development as barriers for participation.

Participation in OSS projects is granted to any individual that has sufficient programming skills to modify the source code (Lakhani & Panetta, 2007; Raasch et al., 2008). However the privilege to add/modify source code, especially within the code repository, is given only to a few trusted developers (von Krogh & von Hippel, 2003). Also contributors that behave according to the "joining script" (von Krogh, Spaeth, & Lakhani, 2003) of the community they want to adhere to, are more likely to be accepted in the project and ultimately gain committer access.

Being more technical "coding" in nature, OSS communities expect from their developers a certain level of expertise (Oreg & Nov, 2008) and hence posing a "skill" barrier for participation. Based on Wikipedia research by Zhang & Zhu (2006) anyone can become an author even without logging in. Adding to that study Müller-Seitz & Reger (2010) show that "content-related" permissions differ among various visitors,

signed-on users and administrators and actually impede the full freedom to contribute similar to OSS with gradual access and permission rights.

A third barrier for contribution is the cost of development for an OSS/OC task. But interestingly, both initiatives are characterized by a low cost of development within five categories:

- Cost of developing the task (von Hippel & von Krogh, 2003)
- Cost of revealing the good developed (von Hippel & von Krogh, 2003)
- Cost of coordination (Moon & Sproull, 2008; Osterloh & Rota, 2007)
- Cost of participation (Anthony, Smith, & Williamson, 2007; Lakhani & Panetta, 2007)
- Cost of integration (Benkler & Nissenbaum, 2006)

4.3. Process

The process category is analyzed in terms of project duration time and evaluation phases, supporting technological tools, and project governance. Figure 10 displays the categories linked to corresponding studies.

Process				
Source	**Domain**	**Project Duration and Evaluation Phases**	**Supporting Technological Tools**	**Governance**
Muller-Seitz & Reger (2010)	OC			✓
Bryant et al. (2005)	OC		✓	✓
Chesney (2006)	OC	✓		
Rettberg (2005)	OC	✓		✓
Wikipedia (2011)	OC	✓	✓	✓
Ortega et al. (2008)	OC		✓	
Crowston & Howison (2005)	OSS			✓
Gabriel & Goldman (2002)	OSS			✓
Ghosh (2005)	OSS			✓
Glott et al. (2007)	OSS			✓
Bonaccorsi & Rossi (2003a)	OSS		✓	✓
Johnson (2006)	OSS			✓
Comino, Manenti, & Parisi (2007)	OSS	✓		
Krogh et al. (2003)	OSS	✓		
Kogut & Metiu (2001)	OSS		✓	
Lee & Cole (2003)	OSS			✓
Madey, Freeh, & Tynan (2002)	OSS		✓	
Markus (2007)	OSS			✓
Krishnamurthy (2002)	OSS	✓		
O'Mahony & Ferraro (2005)	OSS			✓
O'Reilly (1999)	OSS			✓
David et al. (2003)	OSS	✓		
Lakhani & Panetta (2007)	OSS		✓	
Lerner & Tirole (2005a)	OSS			✓
Rossi (2004)	OSS			✓
Raymond (1999)	OSS			✓
SourceForge (2011)	OSS	✓		
Weiss (2005)	OSS	✓		
Lioubareva & Feledziak (2007)	OSS,OC			✓
Oreg & Nov (2008)	OSS,OC		✓	

Figure 10: Process Literature

Project Duration and Evaluation Phases

According to David, Waterman, & Arora (2003), most of the projects surveyed in their sample lasts on average for 1.9 years. The average time between the launch of the project and the first commit of code is approximately 40.8 days (Krogh et al., 2003). During that time OSS projects follow a six step evolution process (Comino et al., 2007; Krishnamurthy, 2002; Weiss, 2005) shown in figure 11 and also applied by SourceForge.net (SourceForge, 2011) for its project status indication.

Figure 11: OSS project phases

The planning phase initializes the project, followed by the pre-alpha stage, which refers to the software development stage. Alpha stage refers to the first release of software, which is ready for the first software testing. Beta is the next stage in which the software is almost complete and needs testing for minor bugs. Lastly, the stable and mature phases are the last stages of a project, in which the software is considered to be developed successfully as it fulfills the needs of the initiator (Comino et al., 2007).

Focusing on OC and Wikipedia, it is difficult to define a duration time of the content creation process. Wikipedia articles are understand as never completed but ever-evolving, under a continuous editing process (Chesney, 2006; Rettberg, 2005; Wikipedia, 2011). To the best of our knowledge, there are no scholarly works that study the stages of a content creation. However, several evolution processes are in place offering research opportunities. As uncommented example the Wikipedia community outline about the article creation (Wikipedia, 2011) is shown in figure 12 .

Figure 12: Wikipedia article evolution phases

Supporting Technological Tools

The technological environment sustaining OSS/OC projects should facilitate easy access and low cost involvement. By lowering barriers to participate for volunteers, the tools sustaining this process are critical for success of such initiatives and can be summarized in three groups:

- Development tools
- Communication tools
- Project management tools

OSS projects are developed by the aid of tools such as text editors, compilers, debuggers, and source code repositories, which are widely available (Lakhani & Panetta, 2007). Communication is achieved and maintained through e-mails, File Transfer Protocol and newsgroups (Bonaccorsi & Rossi, 2003a). Project management tools include bug tracking, discussion forums, Current Version System (CVS), and wikis for

documentation (Madey, Freeh, & Tynan, 2002). The tools presented are not exclusive and are used simultaneously. For instance, Apache has a common repository hosted centrally at http://httpd.apache.org/ which enables code development, a developer mailing list for the communication and Current Version Control archive and bug reporting database as project management tools (Kogut & Metiu, 2001).

Wikipedia is based on a "wiki" engine, a technology that allows multiple individuals to modify user-contributed content (Anthony et al., 2007; Benkler, 2002; Rettberg, 2005). Additional development tools include editing, browsing, and citation tools (Wikipedia, 2011). The most used communication channel in Wikipedia is the discussion page, where authors discuss the changes of content (Bryant, Forte, & Bruckman, 2005). Also, project management is maintained via tools such as page history, which is a database of the recent changes of a page; watchlist, as a surveillance for specific pages in which authors are interested and archives used for discussions and changes in the content (Bryant et al., 2005; Wikipedia, 2011). This easy and user-friendly interfaces reducing technological barriers (Ortega, Gonzalez-Barahona, & Robles, 2008) and enable straightforward contribution needing only basic computer literacy skills for content contribution (Oreg & Nov, 2008) besides the content skills.

Governance

The governance of a community refers to the coordination and safeguarding of community interactions (Markus, 2007). To ensure coordination and safeguarding different governance mechanism, such as distribution of rights and responsibilities and the policies followed have been identified (Zwass, 2010). Accordingly we focus in this section on leadership and structural organization as central aspects of governance.

Raymond (1999) introduced the concept of "the bazaar" to explain the unmanaged and decentralized mode of organization in the OSS projects. However, that anarchistic image of the OSS development has led to the confusion (Bonaccorsi & Rossi, 2003; Lerner & Tirole, 2002) that OSS projects are not lead with formal authority (Johnson, 2006). Instead, these projects exhibit specific community norms that direct the community behavior (Müller-Seitz & Reger, 2010; O'Mahony & Ferraro, 2005). Therefore various OSS initiatives have developed different modes of governance such as "benevolent dictatorship" (Linux), "rotating leadership" (Perl) and "voting committee" (Apache) (O'Reilly, 1999; Rossi, 2004).

In terms of OSS structural organization, three main forms emerged:

- Onion-like model of organization (Crowston & Howison, 2005; Gabriel & Goldman, 2002; Glott et al., 2007; O'Reilly, 1999), represents an onion with several layers, accordingly community users are organized in layers and progress to the core as shown in figure 13.

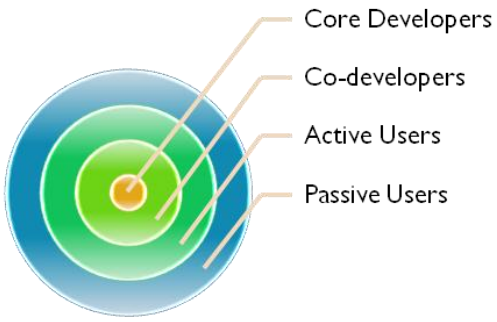

Figure 13: Onion-like organizational structure of OSS communities

- Two-tier structure (Lee & Cole, 2003), symbolizes distributes users in a large periphery and a small amount of core developers as shown in figure 14.

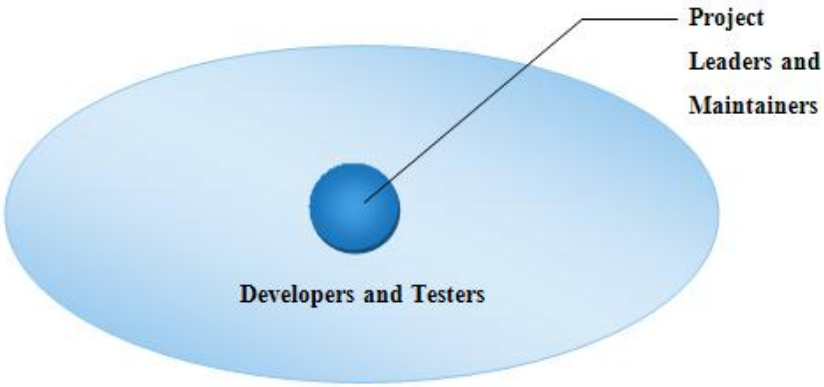

Figure 14: Two-tier organizational structure of OSS communities

- Collection of cores (Ghosh, 2005), displays the community organized into a collection of cores, each having their own smaller peripheries, which overlap each other as shown in figure 15.

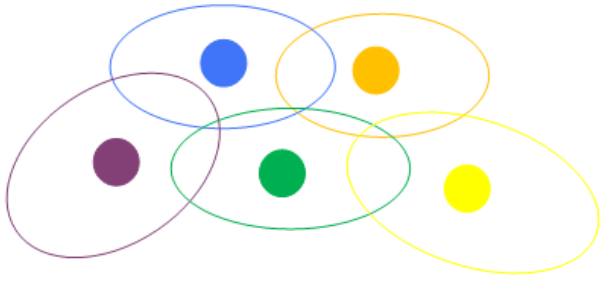

Figure 15: Multi-core organizational structure of OSS projects[13]

[13] Source: Self-construction, modeled after Ghosh (2005), P.16

Wikipedia is the only example that is studied for its mode of governance within the OC stream. Wikipedia is built upon “wiki” pages, which are anarchic in nature (Rettberg, 2005). However, as Müller-Seitz & Reger (2010) note Wikipedia involves various bureaucratic organizational mechanisms such as different types of permission, rules and institutionalization (see figure 17). Hence the Wikipedia community governs itself following a hybrid model of bureaucratic collective norms and bodies such as the ‘Arbitration Committee’, the ’Association of Members’ Advocates’ or the ‘Wikimedia Foundation’ (Bryant et al., 2005; Müller-Seitz & Reger, 2010).

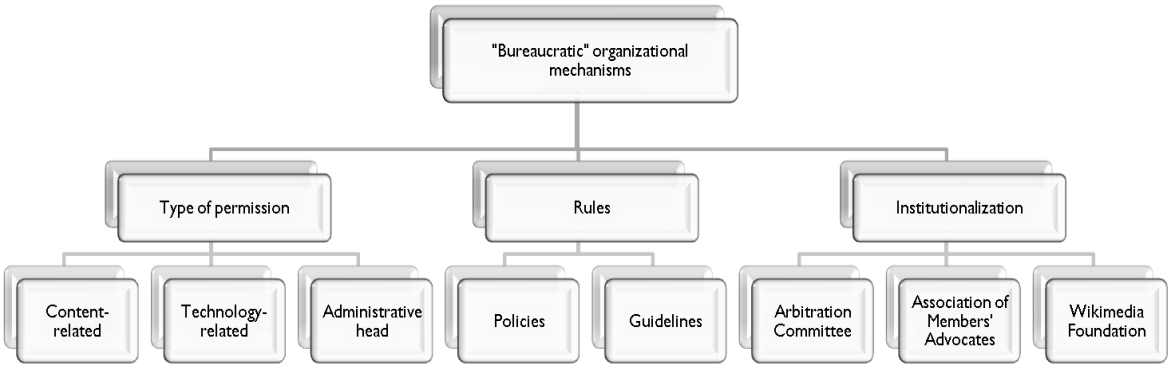

Figure 16: Bureaucratic organizational mechanisms of Wikipedia[14]

In terms of structural organization mode, Wikipedia represents a two-tier structure, with two main types of users: anonymous and registered contributors. A second example exhibiting the two-tier structure of the OC communities is “Take the Bus” (Lioubareva & Feledziak, 2007). In this case, the core of the project consists of the project initiator who provides the basis for potential contributors in the periphery.

4.4. Final Product

This section describes the final outcome of the OSS/OC project. The characteristics analyzed in this comparison cluster in regards to the final product are quality and the license applied as highlighted in figure 17 linked to its authors.

[14] Source: Müller-Seitz & Reger (2010), P.469

Product			
Source	**Domain**	**Product Quality**	**Licenses**
Benkler & Nissenbaum (2006)	OC	✓	
Cheliotis (2009)	OC		✓
Chesney (2006)	OC	✓	
Glott, Schmidt, & Ghosh (2010)	OC	✓	
Creative Commons (2011)	OC		✓
OECD (2007)	OC		✓
Greeinstein & Zhu (2012)	OC	✓	
Okoli (2009)	OC	✓	
Pfaffenberger (2001)	OC		✓
Berlecon (2002)	OSS		✓
Bonaccorsi & Rossi (2003b)	OSS		✓
Anthony, Smith, & Williamson (2007)	OSS	✓	
Bonaccorsi & Rossi (2003a)	OSS	✓	
Johnson (2006)	OSS	✓	
Comino, Manenti, & Parisi (2007)	OSS	✓	✓
Kogut & Metiu (2001)	OSS	✓	
Lerner & Tirole (2005b)	OSS		✓
Lee & Cole (2003)	OSS	✓	
Open Source Initiative (2011)	OSS		✓
David et al. (2003)	OSS		✓
Lakhani & Panetta (2007)	OSS	✓	
Osterloh, Rota, & Kuster (2002)	OSS	✓	
Raymond (1999)	OSS	✓	
von Hippel (2001)	OSS	✓	
Weiss (2005)	OSS		✓
Oreg & Nov (2008)	OSS,OC	✓	

Figure 17: Product Literature

Quality of Product

Quality describes the degree to which the product fulfils the desired need. Hence quality is perceived differently by each person and the measurement criteria are diverse. Studies reflecting the OSS development argue that this way of production is not only efficient (Kogut & Metiu, 2001), but also of superior quality compared to software produced from traditional firms (Johnson, 2006; von Hippel, 2001). This quality level in software is achieved from a review process. A large audience tests the program, identifies and reports the bugs, debugs it and provides improvement suggestions (Lee & Cole, 2003; Osterloh et al., 2002). Therefore, the critical peer-review process is the central mechanism in the OSS projects to maintain high quality (Johnson, 2006; Lee & Cole, 2003; Oreg & Nov, 2008). The quality measurements vary from aesthetic issues such as elegance of the code to functional issues such as bug absence (Comino et al., 2007). According to Bonaccorsi & Rossi (2003a) a piece of high-quality code

exhibits high performance, but at the same time is simple, clear and logic. Additionally, three steps for a piece of code to be considered of sufficient quality and to be integrated in the official software release is presented by Lee & Cole (2003): (1) appear correct to the project leader (2) receive approval from code maintainer(s) and (3) be well tested by the other developers.

Whereas in software development, several times executing the code can validate its functionality, it is difficult to conduct such "one-click tests" in content creation. Even if there is an "running" file (e.g. a text web page, song file, etc.), its quality is subordinately judged based on "source code" functionality criteria, but primarily on subjective output aspects (Okoli, 2009). Subjective quality has been operationalized in Wikipedia as a high rate of accuracy (Chesney, 2006), sometimes comparable to the Encyclopedia Britannica (Anthony et al., 2007; Lakhani & Panetta, 2007), and its level of being [less] biased in its context (Greeinstein & Zhu, 2012).

Unlike OSS, quality in content is created without official peer-review process[15]; any page can be edited by anyone, and the contribution will become immediately part of the end-product (Oreg & Nov, 2008). Raymond (2002) also highlights the difference between open source software and content in terms of review: "Music and most books are not like software, because they don't generally need to be debugged or maintained" (Raymond, 2002, p 192). Also Nov and Oreg (2008) note, that content initiatives generally do not need a review process. Anthony et al. (2007) explain the high quality articles without formal review with the type of contributors to Wikipedia: anonymous users (Good Samaritans) who are likely to be experts in their fields author high-quality articles (or correct mistakes they see) and registered users (Zealots) who have a high level of participation, being interested in reputation within the community, and are generally concerned about the quality of articles. Glott et al. (2010) also show that less experienced users have a high barrier for contribution and hence only knowledgeable users contribute.

However, Schroer and Hertel (2009, p.113) see "clear structural similarities such as peer-review processes" between software and content, represented by Wikipedia, and draw their argument on feedback processes as important factor for writing articles. Additionally Wikipedia articles are described as never completed but ever-evolving following a continuous editing process (Chesney, 2006; Rettberg, 2005). Furthermore a different OC example using reviews to maintain quality and meet the users' acceptance is the technology news platform "Slashdot". In Slashdot community members assure the quality and relevance for the posts by rating and commenting to these posts. This is viewed by Benkler & Nissenbaum (2006) as a strict peer-review system and hence similar to the software environment. Slashdot hence is seen as example the review process is critical to achieve the desired accepted quality level.

Summing up, both domains utilize Linus' law "Given enough eyeballs, all bugs are shallow" (Raymond, 1999), but debugging access and sustaining high quality can be managed differently: Either by "cherry picking" by administrators or purely trusting the "wisdom of the crowds".

IP Regime

Copyright laws have governed for years the various types of creative works, with the intention to protect the copyright holders from the illegal duplication and distribution of their works for a certain amount of time (Bonaccorsi & Rossi, 2003b; Pfaffenberger, 2001). However, even though this mechanism is aimed to trigger the creativity of many authors, in the view of Pfaffenberger (2001) is it destructing the public

[15] Indeed, Wikipedia started a content classification system to indicate article comprehensiveness and encourage users for quality enhancement.

domain and excludes further innovators. Moreover, recent tendencies to open up innovation processes promote broader access to information (Chesborough, 2003) and question the established "closed" paradigm to commercialize innovation (Teece, 1986).

The first open source [software] license was introduced by Stallman in the mid-1980s, namely the GNU Public License (GPL) (Lerner & Tirole, 2005b). Even today GPL is the most widely used OSS License (Berlecon, 2002; David et al., 2003; Lerner & Tirole, 2005b; Weiss, 2005), even many more licenses can be applied to OSS. According to the Open Source Initiative, more than 40 schemes are defined as OSS licenses (Comino et al., 2007; Open Source Initiative, 2011). The most popular ones and their key characteristics are summarized in figure 18:

License	Available at no cost	Source code freely available	No usage restrictions	Distribution allowed	Modification allowed	Copyleft [17]	Commercial exploitation allowed [18]
Public Domain	✓	✓	✓	✓	✓		✓
GNU Public License (GPL)	✓	✓	✓	✓	✓	✓	
GNU Lesser General Public License (LGPL)	✓	✓	✓	✓	✓	✓	✓
Mozilla Public License (MPL)	✓	✓	✓	✓	✓	✓	✓
Berkeley Software Distribution (BSD) License	✓	✓	✓	✓	✓		✓
MIT License	✓	✓	✓	✓	✓		✓
Apache License	✓	✓	✓	✓	✓		✓

Figure 18: OSS Licenses

Within OC the most commonly known and used licenses with more than over 400 million works distributed are the Creative Commons (CC) licenses (Creative Commons, 2011). The CC licenses were proposed by Lawrence Lessig in 2002, as a mean to promote the growth of free culture (Cheliotis, 2009) and to maximize digital creativity, sharing and innovation (Creative Commons, 2011). In the terminology of CC licenses, the main CC Licenses and their characteristics are summarized in figure 19 (Creative Commons, 2011; OECD, 2007):

License	No usage restrictions	Accredit the copyright holder	Distribution allowed	Modification allowed	Share alike	Commercial use allowed
CC0- No Rights Reserved (Public Domain)	✓		✓	✓		✓
Attribution CC BY	✓	✓	✓	✓		✓
Attribution-ShareAlike CC BY-SA	✓	✓	✓	✓	✓	✓
Attribution-NoDerivs CC BY-ND	✓	✓	✓			✓
Attribution-NonCommercial CC BY-NC	✓	✓	✓	✓		
Attribution-NonCommercial-ShareAlike CC BY-NC-SA	✓	✓	✓	✓	✓	
Attribution-NonCommercial-NoDerivs CC BY-NC-ND	✓	✓	✓			

Figure 19: OC Licenses

5. Discussion of Results and Research Implications

Open Source Software (OSS) has already been established as a successful model of production and organizational structure. The model is expanding in the collaborative production of other digital goods such as text, music, video etc. named Open Content (OC). Our comparison reveals differences within the four framework clusters Actors, Task, Process and Product as shown in figure 20 and raises questions for further research. Especially the clusters Actor and Product differ whereas Task and Process reveal no strong differences.

Comparative Overview		
Criteria	**OSS**	**OC**
Product		
Type of Product	Executable file	Body of knowledge
Quality of Product	Comparable or higher than proprietary software	Comparable or more recent than proprietary products
Quality of Product/ Peer-review	Formalized peer-review process and finally acceptance by administrators necessary	Different approaches of peer review and content acceptance available
IP Regime	Various licenses with several characteristics available	Various licenses with several characteristics available
Actor		
Numbers	Wide range strongly depending on project, following power law distribution and cave model	Wide range strongly depending on project, following power law distribution and cave model
Type	Individual contributor, organizations (Firms, NGO)	Individual contributor, organizations (Firms, NGO)
Diversity and educational Background	Very strong male shift, but all educational levels present	Strong male shift, but all educational levels present
Professional Background and Skills	Highly skilled programmers often with professional software industry experience	Highly motivated individuals with projected skills towards project content
Motives for Participation	Started mainly intrinsic, but progresses towards (commercial) benefit usage	Currently mainly intrinsic (ideology, fun), commercial interest nascent or not observed
Task		
Modularity of Task	Modularity identified as key success factor and software architecture thereafter aligned	Modularity observed and often naturally included in product
Effort Intensity	11.4-14.3 h/week (but high variance)	8.27-15.5 h/week (but high variance)
Barriers to Participation	Certain threshold of content and coding expertise in order to pass the review process, commitment status depending on socialization process	Certain threshold of content expertise, commitment status depending on socialization process depending on project
Process		
Project Duration and Evaluation Phases	Evolutionary process	Evolutionary process
Supporting Technological Tools	Product repository, development tools, and additional project discussion tools	Product repository, development tools, and additional project discussion tools. Often direct product editing possible.
Governance	Bureaucratic organizational mechanisms, socialization allows progressing towards centre	Bureaucratic organizational mechanisms, socialization allows progressing towards centre

Figure 20: Main differences OSS and OC

Task and Process, referring to an organizational view, are two categories that can be decided upon start and are little depending on the contributors' involvement. Comparing OSS to OC, we assume that each group learns from each other to choose the most suitable organizational set-up and hence profits from the high transparency in open source. Putting these findings on the research agenda, we raise the questions which of the several conditions are chosen under which circumstances and which are the more sustaining ones. Is for example a dictator or a team leadership more powerful, and how is the influence of firms? How should a governance structure be established and adapted due to growing community size? Linking the set-up to the contextual factors is therefore a fruitful area for further research for theory and practice. Additionally several research lacks in OC were identified, being already touched in OSS studies:

- Task related research questions: What is the effort intensity for the OC task and the resources spent to create digital content? What are the barriers to participate in such initiatives?
- Process oriented research questions: What are the phases that an OC project goes through? How are OC projects governed? What organizational structures, policies and rules do they follow?

The category Actors reveals strong participants heterogeneity, indicated by the contributors' Type, Motivation, Employment, Skill Level and Diversity. However seen from a lighthouse point of view, besides diversity[16], professionalism seems to be a key influence factor. In OSS, there are more corporate participants than in OC, motivation is more extrinsically than intrinsically driven, more people are commercially employed and their skill level is assumed higher in their respective domain. However, this would contradict the original intention of OSS and especially the findings in early OSS research regarding user characteristics. It is rather likely, that OSS grew out of it childhood with mainly altruistic and fun motivated contributors to a more serious phenomenon which attracts the participation of firms with concrete commercial interests and users with higher signaling intensions. This proposition is also supported by research of "Men on the Inside" phenomenon (Dahlander & Wallin, 2006; Lee, 2011) showing that firms strategically influence OSS communities. In contrast, OC is still developing and commercial interest has just started. Hence OC should be compared to OSS in its infanthood days, where participants desired to connect with peers and follow their leisure interests where equally prevailing. A time series analyzes could shed light in this perspective comparing of early analysis of OSS with today's settings and additional to OC. Did something change? How does the research field develop? Is OC now in the situation software has been some years ago? However, sensitivity and curiosity should be raised when comparing the actor's attributes within OS and OSS, and even between early and late studies. These, additionally, should furthermore drive research to link fundamental heterogeneity to project specifica to obtain central relationships (e.g. particular motivations to governance aspects).

Furthermore, OC research should progress in several contributors related research aspects:

- Diversity and skills of the actors in OC communities; Are the "missing" technical skills reducing the barriers for participation? How is firm involvement and potential commercial exploitation present?

[16] Reasons for that are unsure, however one motive can be the general low participation of female contributors in both OSS and OC. Additionally, OC projects may attract more female participants due to more appealing topics.

The Final Product category reveals quality as crucial aspect. A high quality (of software or content) refers to two aspects: (1) is the project free of errors and bugs addressing the review process and (2) has the project reached the desired (end-) status. Due to the nature of OSS and OC these quality aspects are seen differently, and especially when it comes to aesthetic (beauty of code or writing) or belief (convictions and creeds) matters, OSS has the advantage of presenting a stable version or not that reaches the desired (objective) outcome (e.g. calculating pounds in kilo), whereas the status in OC is difficult to evaluate. Hence, research about participation according to the process development and contributors' quality conflicts would bring insights into contributors' staying or leaving in the long run. Furthermore while, the OSS projects use formalized peer-review and established techniques to maintain the quality of the code, in the case of OC initiatives, it is difficult to generalize the techniques used, based on only two examples. Research about the accepted degree of participation restriction in trade of improved quality could open an intense discussion and deepen the openness vs. control discourse (West & O'Mahony, 2008). Additionally the consequences of the product nature are unclear. Whereby OSS requires access to the source code to modify the product, OC initiates outcome often united the source code and the human readable product. Easy to use tools and little technical knowledge in Wikipedia for example lowering participation barriers and facilitate direct product modification. Hence it opens product improvement to a many people without detailed technical skills but expert content knowledge and novel ideas. Understanding these mechanisms would allow integrating even more people in the production process and foster innovation.

Also several research gaps resulted from the framework application:

- How is quality defined, maintained and improved in OC projects? How are concrete governmental rules in OC? What is the optimal level of openness for product innovation, should everybody allowed to participate or are access right for own use satisfying?

5.1. Research Limitation

Besides the research implication, there are several imitations of this study to be borne in mind. OSS research area is quite established, but OC is still a nascent research phenomenon. This is evident in the process of finding reliable and relevant scholar work on OC projects. This lack of research affects the incompleteness of the comparisons between OSS and OC. Studies shedding light in the OC communities focusing mainly on Wikipedia. Even though, the results might not be applicable to other OC communities, judging the popularity and the success of Wikipedia (Raasch et al., 2008), it is still a valid source of information but in order to acquire a complete picture of the OC phenomena more instances from this area should be studied.

A second limitation concerns the scope of study of OC and OSS projects. Up to now mainly the most successful, and iconic, projects (Linux, Apache, Mozilla, and Wikipedia) are taken into account. Smaller, medium, and little successful or younger projects are little researched but are necessary for a full picture of the open source landscape.

A third limitation lies in the mainly qualitative comparison. Many studies are single case based and of qualitative nature. Large scale empirical evidence and also more quantitative research would bring further concrete insights and research to a more representative and comparable level. Unfortunately there are few studies available that provide quantitative insight further than an effect direction or qualitative description that could have been used for comparison.

5.2. Conclusion

Open source started with software and has spread to adjacent areas like books, music, encyclopedia called Open Content. The objective of this paper was to find uniting and differencing aspects of Open Source Software and Open Content. This study hence contributes to the literature and supports practice in a multifold way. First, the study unveils the main concepts of OSS and OC, clustered in a systematic framework. It provides insight about transferability of research findings and shows that OSS and OSC differ in essential aspects (e.g. participating actors) but on the other hand have many similarities (e.g. organizational setting). Second it provides a structured comparison of existing literature especially within the OC domain. Thirdly, most of the research is concentrated on OSS, even though OC projects are well established, our analysis broadly indicates research gaps and stimulates further studies in the OC[17] stream. By studying further the OC creation, a full picture of this phenomenon can emerge and helps to develop a comprehensive theory about this collaborative model also in accordance to the umbrella phenomenon Open Source Innovation. It hence helps to understand implications of this phenomenon for the contributors, users, hosts, and firms of this new organizational model and nexus of innovation.

[17] We also believe this applies to Open Source Hardware also called “Open Design“

References

Anthony, D., Smith, S. W., & Williamson, T. (2007). The Quality of Open Source Production: Zealots and Good Samaritans in the Case of Wikipedia. Quality. Hanover NH.

Baldwin, C. Y., & Hippel, E. von. (2011). Modeling a Paradigm Shift: From Producer Innovation to User and Open Collaborative Innovation. Organization Science, 22(6), 1399-1417.

Benkler, Y. (2002). Coase's Penguin, or, Linux and" The Nature of the Firm." Yale Law Journal, 112, 369–446.

Benkler, Y., & Nissenbaum, H. (2006). Commons-based Peer Production and Virtue. Journal of Political Philosophy, 14(4), 394-419.

Benussi, L. (2006). The evolution of free / libre open source software. Retrieved February 2012, from http://flosshub.org/358

Berlecon. (2002). Free/Libre Open Source Software: Survey and Study. 2002. Berlin.

Bonaccorsi, A., & Rossi, C. (2003a). Why Open Source software can succeed. Research Policy, 32(7), 1243-1258.

Bonaccorsi, A., & Rossi, C. (2003b). Licensing Schemes in the Production and Distribution of Open Source Software: An Empirical Investigation. SSRN Electronic Journal. Pisa, Italy.

Bryant, S. L., Forte, A., & Bruckman, A. (2005). Becoming Wikipedian: transformation of participation in a collaborative online encyclopedia. Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work (pp. 1–10).

Cedergren, M. (2003). Open content and value creation. First Monday, 8(8). Retrieved February 2012, from http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1071/991

Cheliotis, G. (2009). From open source to open content: Organization, licensing and decision processes in open cultural production. Decision Support Systems, 47(3), 229-244.

Chesbrough, H.W. (2003). Open Innovation: The new imperative for creating and profiting from technology. Boston: Harvard Business School Press

Chesney, T. (2006). An empirical examination of Wikipedia's credibility. First Monday, 11(11). Retrieved February 2012, from http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1413/1331

Clarke, R. (2004). Open Source Software And Open Content As Models For eBusiness. 17th Bled eCommerce Conference, eGlobal. Bled, Slovenia.

Comino, S., Manenti, F., & Parisi, M. (2007). From planning to mature: On the success of open source projects. Research Policy, 36(10), 1575-1586.

Creative Commons. (2011). Creative Commons Licenses. Retrieved August 2011, from http://creativecommons.org/licenses/

Creswell, J. W. (2009). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. Research Design. California, USA: Sage Publications.

Crowston, Kevin, & Howison, J. (2005). The social structure of free and open source software development. First Monday, 10(2). Retrieved February 2012, from http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1478/1393

Crowston, K., Annabi, H., & Howison, J. (2003). Defining open source software project success. Proceedings of the 24th International Conference on Information Systems (icis 2003) (Vol. 1, pp. 327–340).

Crowston, Kevin, Wei, K., Howison, J., & Wiggins, A. (2010). Free / Libre Open Source Software Development: What We Know and What We Do Not Know. ACM Computing Surveys. Forthcoming.

Dahlander, L., & Wallin, M. W. (2006). A man on the inside: Unlocking communities as complementary assets. Research Policy, 35(8), 1243–1259.

Dalle, J.-M., & David, P. (2005). Allocation of Software Development Resources in Open Source Production Mode.

David, P. A., Waterman, A., & Arora, S. (2003). THE FREE/LIBRE/OPEN SOURCE SOFTWARE SURVEY FOR 2003. Source. CALIFORNIA, USA.

Free Software Foundation. (2010). The Free Software Definition. Retrieved June 9, 2011, from http://www.gnu.org

Gabriel, R. P., & Goldman, R. (2002). Open Source: Beyond the Fairytales. Retrieved February 2012, from http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1071/991

Ghosh, R. (2007). Sustainable Openness. Washington DC: Workshop on Cyberinfrastructure and the Social Sciences, National Science Foundation. Retrieved February 2012, from http://www.flossproject.org/papers/20070129/DCI-NSF-rishabghosh.pdf

Ghosh, Rishab A., Glott, R., Krieger, B., & Robles, G. (2002). Free/Libre and Open Source Software: Survey and Study.Part IV: Survey of Developers. Retrieved August 2012, from http://www.flossproject.org/report/Final4.htm

Ghosh, Rishab Aiyer. (2005). Understanding Free Software Developers : Findings from the FLOSS Study 1. In J. Feller (Ed.), Perspectives on free and open source software (pp. 23-46). MIT Press.

Ghosh, Rishab Aiyer. (2006). Who is behind open source? Gartner Open Source Summit. Barcelona.

Ghosh, Rishab Aiyer, & Prakash, V. V. (2000). The Orbiten Free Software Survey. First Monday, 5(7). Retrieved February 2012, from http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/769/678

Glott, Ruediger, Schmidt, P., & Ghosh, R. (2010). Wikipedia Survey – Overview of Results. Verbatim (Vol. 2010).

Glott, Rüdiger, Meiszner, A., & Sowe, S. K. (2007). Report on the Learning Environment of FLOSS Communities.

Greeinstein, S., & Zhu, F. (2012). Is Wikipedia Biased? American Economic Association.Forthcoming.

Hao, X., Zhengang, Z., Chunpei, L., & Zhuo, D. (2008). The Study on Innovation Mechanism of Open Source Software Community. 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, 1-5.

Hars, A., & Ou, S. (2001). Working for free? Motivations of participating in open source projects. Proceedings of the 34th Annual Hawaii International Conference on System Sciences.

Henkel, J. (2004). Patterns of Free Revealing – Balancing Code Sharing and Protection in Commercial Open Source Development. Technology Management. Institute for Innovation Research, Technology Management and Entrepreneurship, University of Munich.

Huff, A. S. (2008). Designing research for publication. Design (pp. 85-107). California, USA: SAGE Publications.

Johnson, J. P. (2006). Collaboration, peer review and open source software. Information Economics and Policy, 18(4), 477-497.

Ke, W., & Zhang, P. (2008). Motivations in OSS Communities: The Mediating Role of Effort Intensity and Goal Commitment. International Journal of Electronic Commerce, forthcomin.

Koch, S., & Schneider, G. (2002). Effort, co-operation and co-ordination in an open source software project: GNOME. Information Systems Journal, 12(1), 27-42.

Kogut, B., & Metiu, A. (2001). Open-Source Software Development and Distributed Innovation. Oxford Review of Economic Policy, 17(2), 248-264.

Krishnamurthy, S. (2002). Cave or Community? An empirical examination of 100 Mature Open Source Projects. First Monday, 7(6). Retrieved February 2012, from http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/960/881

Lakhani, K. R., & Panetta, J. a. (2007). The Principles of Distributed Innovation. Innovations: Technology, Governance, Globalization, 2(3), 97-112.

Lakhani, K. R., & Wolf, R. G. (2005). Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. In J. Feller, B. Fitzgerald, S. Hissam, & K. R. Lakhani (Eds.), Perspectives on Free and Open Source Software (pp. 3-21). MIT Press.

Lakhani, K., & Hippel, E. von. (2003). How open source software works: “free” user-to-user assistance. Research Policy, 32(6), 923-943.

Lee, G. K., & Cole, R. E. (2003). From a Firm-Based to a Community-Based Model of Knowledge Creation: The Case of the Linux Kernel Development. Organization Science, 14(6), 633-649.

Lee, V. (2011). How Firms Can Strategically Influence Open Source Communities: The Employment of “men on the Inside.” Springer.

Lerner, J. (2001). The open source movement: Key research questions. European Economic Review, 45(4-6), 819-826.

Lerner, J., & Tirole, J. (2002). Some Simple Economics of Open Source. The Journal of Industrial Economics, 50(2), 197-234.

Lerner, J., & Tirole, J. (2005). The Economics of Technology Sharing: Open Source and Beyond. Journal of Economic Perspectives, 19(2), 99-120.

Liang, L. (2007). Free / Open Source Software Open Content. Development. Thailand: United Nations Development Programme – Asia-Pacific Development Information Programme.

Lioubareva, I., & Feledziak, B. (2007). Authority, Labour Division, and Conflict Resolution in Commons-based Peer Production.The cases of Debian and Take the Bus Project. Proceedings of The diffusion of Free/Libre Open Source Software (Vol. 54).

Madey, G., Freeh, V., & Tynan, R. (2002). THE OPEN SOURCE SOFTWARE DEVELOPMENT PHENOMENON: AN ANALYSIS BASED ON SOCIAL NETWORK THEORY. Eighth Americas Conference on Information Systems (pp. 1806-1813).

Markus, M. lynne, Manville, B., & Agres, C. E. (2000). What Makes a Virtual Organization Work? Sloan Management Review, 42(1), 13-26.

Markus, M. L. (2007). The governance of free/open source software projects: monolithic, multidimensional, or configurational? JOURNAL OF MANAGEMENT AND GOVERNANCE, 11(2), 151-163.

Moon, J. Y., & Sproull, L. (2008). Essence of Distributed Work: The Case of the Linux Kernel. First Monday, 5(11). Retrieved February 2012, from http://www.firstmonday.org/issues/issue5_11/moon/index.html

Möller, E. (2008). Definition of Free Cultural Works. Retrieved July 9, 2011, from http://freedomdefined.org/Definition

Müller-Seitz, G., & Reger, G. (2010). " Wikipedia , the Free Encyclopedia " as a role model? Lessons for open innovation from an exploratory examination of the supposedly democratic-anarchic nature of Wikipedia. International Journal of Technology Management, 52(3/4), 457-476.

Nov, O, & Kuk, G. (2008). Open source content contributors' response to free-riding: The effect of personality and context. Computers in Human Behavior, 24(6), 2848-2861.

Nov, Oded. (2007). What motivates Wikipedians? Communications of the ACM, 50(11), 60-64.

OECD. (2007). Giving Knowledge for Free: The Emergence of Open Educational Resources. Knowledge Creation Diffusion Utilization.

Okoli, C. (2009). Beyond Open Source Software: An Introduction to Researching Open Content and Wikipedia. Working Papers on Information Systems, 9(64).

Open Source Initiative. (2010). The Open Source Definition. Retrieved June 9, 2011, from http://www.opensource.org

Oreg, S., & Nov, O. (2008). Exploring motivations for contributing to open source initiatives: The roles of contribution context and personal values. Computers in Human Behavior, 24(5), 2055-2073.

Ortega, F., Gonzalez-Barahona, J. M., & Robles, G. (2008). On the Inequality of Contributions to Wikipedia. Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008) (pp. 304-304).

Osterloh, M, & Rota, S. (2007). Open source software development—Just another case of collective invention? Research Policy, 36(2), 157-171.

Osterloh, Margit, Rota, S., & Kuster, B. (2002). Open Source Software Production: Climbing on the Shoulders of Giants.

O'Mahony, S., & Ferraro, F. (2005). Managing the Boundary of an "Open" Project. In J. Padgett & W. Powell (Eds.), Market Emergence and Transformation.

O'Reilly, T. (1999). Lessons from Open-Source Software. Communications of the ACM, 42(3), 33-37.

Pfaffenberger, B. (2001). Why open content matters. Knowledge, Technology & Policy, 14(1), 93–102. Springer.

Preston, M., & Zurer, R. (2003). Open Content & The Search for a Better Intellectual Property Model : Lessons from Open Source.

Pénin, J. (2008). More open than open innovation? Rethinking the concept of openness in innovation studies.Strasbourg.

Pénin, J. (2011). Open source innovation: Towards a generalization of the open source model beyond software. DIME Final Conference. Maastricht: DIME.

Raasch, C., Balka, K., Herstatt, C., & Abdelkafi, N. (2009). THE OPEN SOURCE MODEL – BEYOND SOFTWARE. EURAM Conference. Liverpool.

Raasch, C., Herstatt, C., & Abdelkafi, N. (2008). Creating Open Source Innovation: Outside the software industry. PICMET '08 - 2008 Portland International Conference on Management of Engineering & Technology (pp. 433-442). Cape Town, South Africa.

Raymond, E. (1999). The cathedral and the bazaar. Knowledge, Technology & Policy, 12(3), 23-49.

Raymond, E. (2002). Afterword: Beyond Software? Retrieved August 2011, from http://catb.org/~esr/writings/homesteading/afterword/

Rettberg, S. (2005). All Together Now : Collective Knowledge , Collective Narratives , and Architectures of Participation. Digital Arts and Culture Conference. The Richard Stockton College of New Jersey Arts and Humanities.

Roberts, J., Hann, I.-H., & Slaughter, S. (2006). Understanding the Motivations, Participation and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects. Social Science Research.

Rossi, M. A. (2004). Decoding the " Free / Open Source (F / OSS) Software Puzzle " a survey of theoretical and empirical contributions.

Rosenzweig R. (2006). Can History Be Open Source? Wikipedia and the Future of the Past. The Journal of American History , 93(1), 117-146

Ryan, R., & Deci, E. (2000). Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. Contemporary educational psychology, 25(1), 54-67.

Schroer, J., & Hertel, G. (2009). Voluntary Engagement in an Open Web-Based Encyclopedia: Wikipedians and Why They Do It. Media Psychology, 12(1), 96-120.

Shah, S. K. (2006). Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. Management Science, 52(7), 1000-1014.

SourceForge. (2011). SourceForge. Retrieved August 2011, from http://sourceforge.net/

Teece, D.J.(1986), Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy, Research Policy,15(6, 285-305

von Hippel, E. (2001). Innovation by user communities: Learning from Open-Source Software. MIT Sloan Management Review, 42(4), 82-86.

von Hippel, E., & von Krogh, G. (2003). Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. Organization Science, 14(2), 209-223.

von Krogh, G., & von Hippel, E. (2003). Special issue on open source software development. Research Policy, 32(7), 1149-1157.

von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, joining, and specialization in open source software innovation: a case study. Research Policy, 32(7), 1217-1241.

von Krogh, G., & von Hippel, E. (2006). The Promise of Research on Open Source Software. Management Science, 52(7), 975-983.

von Krogh, G., Haefliger S, Spaeth, S., Wallin M.W. (2012). Carrots and Rainbows: Motivation and Social Practice in Open Source Software Development. MIS Quarterly, forthcoming.

Weiss, D. (2005). A Large Crawl and Quantitative Analysis of Open Source Projects Hosted on SourceForge. Poland.

West, J., & O'Mahony, S. (2005). Contrasting Community Building in Sponsored and Community Founded Open Source Projects. Proceedings of the 38th Annual Hawaii International Conference on System Sciences.

West, Joel, & O'Mahony, S. (2008). The Role of Participation Architecture in Growing Sponsored Open Source Communities. Industry and Innovation, 15(2), 145-168.

Wikipedia. (2011). Wikipedia. Wikimedia Foundation, Inc. Retrieved August 2011, from www.wikipedia.org

Wunsch-Vincent, S., & Vickery, G. (2007). Participative Web: User-created content.

Zhang, X., & Zhu, F. (2006). Intrinsic Motivation of Open Content Contributors : the Case of Wikipedia. Workshop on Information Systems and Economics. Evanston, IL.

Zwass, V. (2010). Co-Creation: Toward a Taxonomy and an Integrated Research Perspective. International Journal of Electronic Commerce, 15(1), 11-48.