1. Introduction

Open source projects and communities are one of the most significant and interesting products of the Internet. With the possibility of connecting and collaborating on a historically unprecedented scale, people have created an astonishing technologies and stores of human knowledge through voluntary contribution. Two of the most significant achievements of the open source phenomenon are the GNU/Linux operating system and Wikipedia, the free encyclopedia that anyone can edit.

The remarkable success of the open source movement raises important questions for the traditional corporate business that has been the dominant model of industry since the Industrial revolution. The most critical questions for corporations are in the areas of motivation, innovation and sustainability. Open source communities have displayed remarkable success on these fronts. Increasingly, new innovations in web technology are emerging out of open source software projects. Members of these projects are deeply passionate about their work, devoting significant portions of their lives to them without expecting profit or intellectual property rights. Finally, these projects appear to sustain themselves without a strong central control, despite members arriving and leaving. These characteristics of open source projects make them very attractive to corporations seeking to learn from their strengths.

Some corporations have acquired open source projects in attempts to fund and foster the open source community and to use open source products and technologies for profit. This *hybrid strategy* aims to generate synergy and market value by combining the best of the open source and traditional corporate model. Popular examples of this strategy are Novell's acquisition of SUSE Linux and more recently Oracle's acquisition of MySQL and OpenOffice.org from Sun. This strategy still remains largely unexplored and unproven and understanding its viability requires understanding the social and organizational factors that make open source communities unique. Then we may be able to answer the big question - "What are the effects on the motivation of members and organizational structure of an open source community when their project is acquired by a corporation?"

Altruism and the desire to create and learn are the main driving forces for open source contributors. Open source members are primarily driven by strong intrinsic motivations and trust and recognition within the community built on past performance plays a critical role in sustaining motivation. On the larger scale, a constant struggle between conflict and self-organization allows the community from falling into global control and allows it to exist in a state of dynamic equilibrium. These ideas will be developed in greater detail in this paper and supported by existing qualitative and quantitative research in this field. In the light of these concepts it will become evident that open source communities operate on social dynamics that are distinct and incompatible with the traditional business model. I will argue from existing evidence and established social scientific concepts of motivation and network effects that the hybrid strategy will initially be met with dissent from the open source community and will inevitably lead to a division of the project into a corporate sponsored commercial branch and a new *forked* open source branch.

The migration of members to the commercial project will result in a shift of motivation towards the extrinsic end of the spectrum and may eventually crowd out intrinsic motivation. However, a majority of the core members of the original project will continue working on the open source branch demonstrating strong intrinsic motivation, strong ties within the existing network and a commitment to open source values.

Therefore, hybrid models attempting to leverage open source communities to drive corporate growth will fail. The open source model represents a socio-technical niche and cannot be fully transferred and merged with traditional profit maximizing models. The paper will conclude with an analysis of Oracle's acquisition of Sun and its attempts to create a hybrid strategy for using the open source MySQL and OpenOffice.org projects for commercial growth. This analysis will highlight the concepts developed throughout the thesis and explain how motivation, network effects and self-organization in the OSS community will influence the outcome of a hybrid strategy.

2. Background

Free/open source software (F/OSS) has its roots from near the beginning of computing and is typically free while providing users with source code that is usually shared via the internet and can be adjusted for users' own needs. In the

1960s, while using computers for their work, researchers had to share software code because commercial software was not available (Moon & Sproull, 2002).[1]

Recent advances in networking technology enable worldwide communication that support social interaction, cooperation, and collaboration for learning and knowledge building (Friedman, 2005). Ubiquitous networking has fostered the development and support of communities that form quickly in support of a particular cause and then disband (Shirky, 2008). Sometimes, however, join- ing an online community involves being committed to participation and incorporates formal knowledge integrated with informal practice. This frequently happens in communities that develop and support open source software (Lakhani & von Hippel, 2003).[2]

3. The Individual

3.1 Theories of motivation

Most of the literature on OSS is split between two schools of thought. The first is the more economic, self-interested, and market-based school of thought (Iannacci, 2002; Lerner and Tirole, 2006). This school of thought believes that the OSS phenomenon can be analyzed in the logic of traditional neoclassical economics, and the model of private reputation development; a type of extrinsic motivation similar to rewards. Participating individuals in the development project can increase their potential wages and income in the future due to development of

[1] Hoda Baytiyeh and Jay Pfaffman, "Open source software: A community of altruists.," *Computers in Human Behavior* 26, no. 6 (November 2010): 1345-1354.

[2] Ibid.

market signals (Spence, 1974) and individual reputation. The second is the more social and anthropological school of thought based on reciprocity, kinship, and gift economies (Zeitlyn, 2003). This model is based on the anthropological literature on gifts (Mauss, 1955; Strathern, 1992; Titmuss, 1970). Kinship, trust, and reciprocity drive this social anthropological approach (Schwimmer, 1995) to OSS development and is closer to a model of ''intrinsic motivation'' (Frey and Stutzer, 2007).[3]

A third model that attempts to synthesize existing theories of reciprocity and motivation is suggested by Chong Ju Choi et al. (2009). This model is based on the idea of a psychological contract, a term introduced by Schein (1965) and Levinson et al. (1962) in the 1960s. OSS programmers have a psychological contract within the global community of software development programmers, linked by trust with shared values and professional backgrounds. The collective, public provision of OSS is possible, because an individual and a commercial behavior would be seen as violating the psychological contract to the global open source community.[4]

For OSS, the global social community has an anthropomorphic identity (Morrison and Robinson, 1997; Rousseau, 1989), thus similar to the loyalty of employees to an organization.[5] This similarity in how individuals in both OSS and traditional corporations anthropomorphize their organizations and display loyalty towards their community also creates a significant challenge for corporations

[3] Chong Ju Choi, Sae Won Kim, and Shui Yu, "Global Ethics of Collective Internet Governance: Intrinsic Motivation and Open Source Software," *Journal of Business Ethics* 90, no. 4 (December 1, 2009): 523-531.

[4] Ibid.

[5] Ibid.

acquiring OSS projects. Individuals loyal to OSS and deeply invested in psychological contracts in that community will be unable to accept the corporate strategy. This is one reason why corporate takeovers invariably result in a backlash from the OSS community; not only from the members of the project but from the worldwide OSS community at large which sees the corporation as disrupting the vision of the OSS movement.

Torvalds (1998), who published the source code of the Linux Kernel, claimed that one of his main personal motives was the – fun to program – and he believed that his co-developers had the same incentive. Conversely, he declared that the success of Linux is related to the reputation and status that might provide the developers with career opportunities prospects (Torvalds & Diamond, 2001). Lakhani and Wolf (2005) found that although financial incentives are important for contributors, work enjoyment is a key intrinsic motivation for such devotion where creativity to improve programming skills and enjoyment were revealed to be the main factors that stimulate contributors' work for free.[6]

Other researchers have showed that contributors' objectives are to reveal their technical capabilities to obtain better job opportunities for future prospects and the main incentives behind the volunteer participation are for extrinsic benefits (Lerner & Tirole, 2000, 2002). Also, Riehle (2007) claimed that software developers

[6] Baytiyeh and Pfaffman, "Open source software."

strive to become contributors in open source projects to acquire more recognition, independence, and therefore to guarantee better future as well as better careers.[7]

Another approach to understanding motivations is proposed by Linderberg (2001) who separates intrinsic motivation into two components: enjoyment and obligation to the community. He assumed that people possess a diversity of objectives while achieving their activities. A frame is created around the main objective with the related compatible objectives. After the main objective is achieved, the other goals still remain in the person's background intentions. Therefore, an individual could have an extrinsic incentive (e.g., monetary rewards) as a main objective along with an intrinsic incentive (e.g., self-enjoyment) as a related objective and vice versa. Individuals can have the two types of motivations that balance one another for a single activity.[8]

3.2 Types of motivation[9]

3.2.1 Motivation to learn

One potential motivator for the open source community membership is the desire to learn. Dewey (1915) argued that humans possess an innate desire to learn. Since adults are able to identify their needs (Knowles, 1980; Wentzel, 1994), they may engage in learning situations to meet a specific goal (Wlodkowski, 1989). One type of learning that may occur in the contribution process to F/OSS is to discover the strategies and methods involved in the process of participation.

[7] Ibid.
[8] Ibid.
[9] Ibid.

3.2.2 Motivation to create

Another motivational factor is creating and sharing an artifact. Constructionism, or ‘‘learning by making”, is based on this deep-based desire of creation and innovation (Harel & Papert, 1991). Open source software developers create and refine software tools for others to use. The creation itself might provide satisfaction: from the initial stages to the completion of the project in order to witness the end of the course of action.

3.2.3 Social motivators

Another motivator for contributing to an OSS project may be to be part of a community. Individuals may be motivated partly to be- long to a community, one of the fundamental human needs (Deci, Vallerand, Pelletier, & Ryan, 1991; Maslow, 1987; Ryan & Deci, 2000). Another social factor is sharing knowledge where the main purpose is the benefit of the whole community by helping others (Lindenberg, 2001).

3.2.4 Extrinsic motivators

There may be extrinsic motivators to contribute to F/OSS projects. The most obvious one is that people develop F/OSS tools to meet their own needs or accomplish work. Von Hippel (1988) found that participants have strong incentives to create solutions to their particular needs. Also, Lerner and Triole (2000) identified two types of payoff for contributions, an immediate payoff (e.g., ability to use the product) and a delayed payoff (e.g., potential future rewards in terms of recognition and reputation). Likewise, one's social stature within the F/OSS community can be related to the performance in the group's activity.

3.2.5 Flow motivators

Another motivator may be the desire to be in a state of flow (Nakamura & Csikszentmihalyi, 2003). Aspects of flow like loosing track of time when people are completely engaged in an activity or having ones' ability and level of challenge in balance. Being com- mitted to F/OSS may be challenging and therefore a flow state might be attained by contributors.

3.2.6 Altruism motivators

One more motivator indicated in a study of Wikipedia is altruism (Baytiyeh & Pfaffman, 2009). Also, Oreg and Nov (2008) emphasized on altruistic motives when exploring the reasons for contribution to OSS. In lieu of tangible rewards, givers receive psychological benefits such as the satisfaction of helping or living up to some commitment (Ross-Ackerman, 1998). Being altruistic provides rewards such as boosting one's ego, enjoyment, and community identification. Altruism is a natural part of human nature and is exhibited in some manner by everyone (Ozinga, 1999). In such a gifted setting, given the abundance of resources, social status is not determined by what one has but by what one gives away, such is the case in the OSS community (Raymond, 1999).

3.3 Getting paid for work – effects on motivation

3.3.1 *Contributors' intrinsic motivations to participate in OSS projects are negatively related to being paid to participate.*

The literature in psychology suggests that other extrinsic motivations, even those that are internalized (and thus not strictly extrinsic) could crowd out intrinsic motivations. With respect to internalized extrinsic motivations that are based on the identification of values, Ryan and Deci (2002) observe that such identification is often compartmentalized and separated from one's other beliefs and values, and is characterized by a reduced self- determination. In her review of the literature on OSS motivations, Rossi (2004) states that use-value motivations can provide a powerful explanation for why people do tasks that may be uninteresting and mundane, and that are not "appreciated per se, for the intrinsic plea- sure and enjoyment a programmer may derive from them" (p. 5).[10]

3.3.2 *Being paid to participate in OSS projects is positively related to contributors' use-value motivations to participate.*

Similarly, individuals who are being paid to con- tribute to OSS projects are likely to have higher status motivations because an interest in attaining status motivates

[10] Jeffrey A. Roberts, Il-Horn Hann, and Sandra A. Slaughter, "Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects," *Management Science* 52, no. 7 (July 1, 2006): 984-999.

individuals to demonstrate competence for which they will be financially rewarded.[11]

3.3.3 *Being paid to participate in OSS projects is positively related to contributors' status motivations to participate.*

According to the classic literature on operant conditioning (Skinner 1953), behavior that is rewarded with positive reinforcement is more likely to be repeated in the future. In Western societies, monetary com- pensation is the ultimate positive reinforcer to regu- late economic activities in organizations. Hence, we expect that contributors who are paid to participate in Apache projects would participate more intensely than those who are not paid. This is because their wages act as a constant positive reinforcer of their participation behavior.[12]

3.3.4 *Being paid to participate in OSS projects is positively related to contributors' level of participation.*

We also expect that individuals with higher levels of use-value motivations would exhibit higher levels of participation. In terms of use-value motivations, one of the most often cited drivers of OSS participation is the opportunity to create code that meets the specific needs of a developer (Raymond 1999). High levels of use-value motivation therefore suggest a high level of participation.[13]

3.4 Sustainability – Why do members continue to contribute?

[11] Ibid.

[12] Ibid.

[13] Ibid.

3.4.1 Building a Utopian community

Computer mediated communication can obscure race, ethnicity, and social class (Friedman, 2005). The OSS communities seem to welcome any member, and offer considerable mobility among roles and positions, since no one distributes the tasks. This community is inclusive worldwide. The open nature of OSS communities resemble are consistent with Welton's (2005) definition of a 'Utopian' community with the recognition that human beings have the capacity of self-determination and self-expression. The concept of 'Utopia' generally circles around ideas of the good society or the perfect society. Many theorists treat Utopia as the motivation for change (e.g., Mannheim, 1936) or the obstacle to it (e.g., Marx & Engels, 1968). H. Baytiyeh and J. Pfaffman (2010) use a broad definition, where 'Utopia' is understood as the expression of the desire for a better way of living, a place and time where equality and freedom converge to liberate human creativity (Levitas, 2004).

In a Utopian community, another motivator is the commitment to give back to the community (Hoffman, 1981). When asked why they continue to participate, respondents consistently indicated that they valued community support, or ''the community spirit." A typical comment was ''I used it, needed help, went to the community to get the help. Once I understood more, I contributed back by helping new users coming on board."[14]

[14] Baytiyeh and Pfaffman, "Open source software."

3.4.2 Embracing liberty

Another reason for participating in F/OSS projects and remaining in the community was the freedom. They expressed two senses of freedom that they valued. The first was the freedom to contribute their efforts however and whenever they choose. Raymond (1999) documented the importance of creative communities free of the power relationships of the workplace that can inhibit open communication and creativity. The second, more altruistic, sense of freedom was that F/OSS can help to free others from the tyranny of proprietary software and the companies that profit from it. Open source communities are multi- faceted; the network is not just the Internet, but contributors form a distributed, loosely coupled, peer-to-peer network (Raymond, 1999). This structure provides support from experts around the world. Participants value being a member of a free community with no boundaries; they love F/OSS because it is open.[15]

3.4.3 Sharing intelligence

In contrast to developing proprietary software in a context that requires non-disclosure, OSS allows one's code and the procedures that are used to have a broader benefit to society. It may be surprising to learn that F/OSS participants are driven mostly by altruistic values where they are giving code, information, and expertise away, while also helping outsiders and new arrivals to come on board or to solve F/OSS technical problems. These values though may be indicative of a new generation's values. This is a significant point of departure from the "closed source"

[15] Ibid.

thinking of the corporate business model and is another reason why hybrid models may fail to integrate proprietary thinking with the altruistic values of open source.

4. The Social Network

4.1 Cryptohierarchies and self-organization

Athina Karatzogianni and George Michaeldes suggest that in communities that exist at the interface between order and randomness (at the edge of chaos), conflict and crisis can act as a catalyst or a defence mechanism towards establishing governance structures or, failing that, disintegration. n order and randomness (at the edge of chaos), conflict and crisis can act as a catalyst or a defence mechanism towards establishing governance structures or, failing that, disintegration. Conflict is a catalyst in the sense of enabling the morphosis of cryptohierarchies, and a defence mechanism in the sense of forcing communities to separate.[16] These concepts may explain why most OSS projects inevitably separate into new communities when corporations acquire them (e.g. Novell and Open SUSE).

Conflict and crisis can result in different outcomes. For example, through negotiation and soft control, communities such as peer-to-peer networks can develop new structures in order to cope with conflict, creating core and periphery groups and cryptohierarchies. In another scenario, due to extreme group polarization, the community is unable to create new structures, but branches out and uses conflict as a defense mechanism to avoid centralization. Or in the worst-

[16] Athina Karatzogianni and George Michaelides, "Cyberconflict at the edge of chaos: Cryptohierarchies and self-organisation in the open-source movement.," *Capital & Class*, no. 97 (Spring2009 2009): 143-157.

case scenario, the community separates into two (forking the code), and there is no collaboration between original and fork, in which case conflict can be constructive or destructive depending on the evolution of the communities and groups involved.[17]

These dynamics are forcing open-source communities and, more often than not, networked communities to exist at the edge of chaos, and to constantly engage in lines of flight and resistance from the system of global control, while ignoring current capitalist practices and 'growing their own' models of self-organizing knowledge creation and exchange (meta-cyberconflict).[18]

4.1.1 Conflicts

Cyberconflicts within the open source communities can be categorized in the following way. First, as *ultra-creative*, *intra-communal* conflicts between individuals in an open- source community. This can lead to much more diverse knowledge creation or, in the worst-case scenario, to code *forking*. Forking, where the code is replicated and continued by another team of developers is different from code branching. For instance, in the proprietary software Unix, different projects incompatible to each other exist (forking). In OSS Linux, official and experimental versions of software exist (branching). NeoOffice is a fork of OpenOffice,org, with an incompatible license (GPL rather than LGPL) due to disagreements about licensing and about the best method with which to port OpenOffice.org to Mac OS X. Forking

[17] Ibid.

[18] Ibid.

is considered to be a bad thing because it implies a lot of wasted effort in the future and tend to be accompanied by a great deal of strife and acrimony between the successor groups over issues of legitimacy, succession, and design direction. Major forks are rare enough that they are remembered individually in hacker folklore (such as the Gnu-Emacs/XEmacs split).[19]

In the bigger picture, there is a general conflict between the open-source community and aligned proprietary software companies supporting open-source initiatives against the Microsoft monopoly and its allies. Here, macro-organizational structures and the dynamics of the IT industry are important, as are questions of identity, strategy (framing) and structure (hierarchy vs. network or hybrid, such as in the Linux case, when Torvalds started rerouting submissions to lieutenants). Within this bigger picture, a meta- conflict occurs synchronously bringing all these different levels together and posing them in direct and intense contact and contrast to the current global system of capitalist accumulation.[20]

4.1.2 Self-organization and the edge of chaos

The 'edge of chaos' is defined as being the state of a system in which the system is undergoing a phase transition: i.e. its behavior is shifting from one state to another. In social systems, 'edge of chaos' refers to the conceptual region between order and chaos, and refers to a system which is at a 'self-organized' state. In open-

[19] "Fork (software development) - Wikipedia, the free encyclopedia", n.d., https://secure.wikimedia.org/wikipedia/en/wiki/Fork_(software_development)#Forking_free_and_open_source_software.

[20] Karatzogianni and Michaelides, "Cyberconflict at the edge of chaos."

source communities and possibly in other network structures, the edge of chaos is captured in two ways in which the system can self-organize. First, open-source communities exhibit power law distribution (see e.g. Healy & Schussman, 2003; Madey, Freeh & Tyran, 2005); and second, every successful community tends to be organized into a two-tier structure with a core and a periphery group (see Lee & Cole, 2003; Michaelides, 2006). Karatzogianni and Michaelides think that these these two forms of self-organisation not only unavoidable, but also a necessary component to the success of the community. First, networks that follow power law distributions tend to be more robust and are more adaptable to environmental disturbances (see e.g. Barabasi, 2002). Second, the fact that communities tend to separate into core and periphery groups enables them to effectively exploit and integrate knowledge from diverse sources (Michaelides, 2006).[21]

4.1.3 Emergence of core and periphery

As the community evolves it reaches a point where it is no longer manageable as a single tier. Because of interconnectivity among developers, the coordination overheads scale exponentially, necessitating the creation of a second governing tier. By modeling this process as a cusp catastrophe model (see Thom, 1975), Michaelides showed that two parameters govern the question of who becomes a core developer: social interconnectivity and knowledge sharing. The amount of knowledge shared governs the asymmetry between core and periphery,

[21] Ibid.

while the level of social interconnectivity governs the rate of change through which a developer progresses from being a peripheral to a core member.

The community separates into core and periphery in order to reduce the coordination overheads; but more importantly, it separates into two different roles: those of exploration and exploitation. While there is a need to reduce the overheads that result from increased connections in the network, there is still a need for knowledge sharing. To this extent, the role of the periphery is to explore knowledge while the role of the core is to exploit it by selecting the best ideas and code (see also Lee and Cole, 2003).[22] This is another key distinction between OSS communities and proprietary project teams. Traditional business models involve teams with well-defined roles and rarely follow the natural evolution into core and peripheral groups that is found in OSS communities. This is another reason why corporate acquisition of OSS projects may fail if there is an artificial imposition of traditional structure onto the twin group structure of OSS.

4.2 Polarization and leadership

Group polarization occurs when 'members of a deliberating group move toward a more extreme point in whatever direction is indicated by the members' predeliberation tendency' (Sunstein, 1999). Polarization increases when the group defines itself by contrast to another group: when there is some sense of identity reinforcing group consensus, rather than complicating things, e.g. in the XFree86 fork X.org. On the other hand, depolarization can occur due to external shock: new

[22] Ibid.

members, new arguments, new information. The lesson from group polarization is that social homogeneity can be damaging to good deliberation, something proven by better knowledge exchange in communities in which conflict actually occurs.

Evidently, a strong hierarchical component is vital to successful OSS (see Jordan Hubbard & FreeBSD; Lee & Cole, 2003; Mockus, Fielding & Herbsled, 2002, 2005-). Core developers are very well organized: 'not a formal organizational chart, but rather a status-based pecking order which is known to project participants and serves as a way of policing members', OSS 'as virtual organizations' rely on mechanisms of social control and self-control, not on trust per se (Gallivan, 2001). Large-scale OSS projects are most often staffed by professional software developers (though not always: see Netscape). Still, questions remain. *How will the increased participation of commercial players influence the ideological issue of cryptohierarchies?* The answer might be in the symbiosis of competitors, or in further forking and disintegration. **For open-source communities to succeed, commercial players need to play with the same rules as everyone else. Only then can there be symbiotic relationships between individuals and/or organizations with different or competing interests.**[23]

4.3 Community models

4.3.1 Practice

One model that these F/OSS communities may follow is the community of practice (CoP) model (Lave & Wenger, 1991). Communities of practice are groups who

[23] Ibid.

‘‘share a concern, a set of problems, or a passion about a topic, and who deepen their knowledge and expertise in this area by interacting on an ongoing basis” (Wenger, McDermott, & Snyder, 2002, p. 4). Group members in CoP interact regularly together to share information, insight, and advice. They accumulate knowledge and become informally bound by the value of the shared learning. Three essential elements are central to a CoP: the domain of knowledge, which enables members to recognize the importance of the community and inspires them to participate; the community, which creates relationships among members based on mutual respect and the willingness to share ideas and experience; and the practice, a set of tools, terms, activities, and documents shared by the members.[24]

4.3.2 Altrusists

Investigations into the motivations of Wikipedia contributors (Baytiyeh & Pfaffman, 2009) suggested another type of community: a community of altruists. A com- mon theme found was the desire to build something that will ben- efit others. Unlike CoP, in which members are working in a trade or to learn one, data suggested that Wikipedians contribute largely for the benefit of others.[25]

5. Corporations and OSS

5.1 Hybrid Strategies

Corporations that acquire open source projects need to define and control property rights for value appropriation. Unfortunately, activities that permit value

[24] Baytiyeh and Pfaffman, “Open source software.”
[25] Ibid.

appropriation by the firm are sometimes detrimental to value creation within the community.[26] Two broad sets of governance mechanisms for a hybrid strategy are discussed here.

5.1.1 Decision-Making Rights.

Code Control. In the gated source community, only the corporate sponsor is allowed to alter the source code. This strict control over the code affects both need-driven and hobbyist participants. Need-driven participants worry that their voices will be drowned out by the needs of the firm and its customers when software-related decisions are made. Such control limits the ability of hobbyists to work and contribute in self-defined ways. In addition, the volume of feed- back and overall activity is likely to decline due to both decreased participation and tighter control over what is committed to the source code and, therefore, used by others.[27]

5.1.2 Property Rights.

Private Ownership of Source Code. Private ownership of the code acts to dismantle the collective development process in a variety of ways. Most noticeably, ownership by the firm creates the possibility that the developer will not have access to the code at a later date. Participants value the results of their efforts and expect to continue using the software well into the future. The open source project gave them this right, but the gated source project did not make this guarantee. Private ownership also appears to inhibit reciprocity: if the firm is not donating the code to the community,

[26] Sonali K. Shah, "Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development," *Management Science* 52, no. 7 (July 1, 2006): 1000-1014.
[27] Ibid.

why should the developer take additional time and effort to donate code to the firm?[28]

These restrictions can be thought of as limiting the value available to the individual developer, i.e., the developer can only use the code for certain purposes, modifications made and deployed must meet community standards (rather than his own preferences), and the code may only be shared with others willing to abide by the community's governance arrangements, thereby decreasing the volume of subsequent improvements and feedback that many developers relish. On the other hand, these restrictions might create value for the company.

Governance practices can create shortcomings in hybrid forms, although some shortcomings can be ameliorated. For example, the firm sponsoring the gated source community has significant resources at its disposal and employed a team of developers and marketers to work on the gated source project. These employees took care of many of the tasks—such as assisting participants, incorporating suggestions and code, and maintaining the overall architecture of the code—that would have been fulfilled by volunteer hobbyists in the open source community.[29]

[28] Ibid.
[29] Ibid.

5.2 Oracle, MySQL and OpenOffice.org

5.2.1 Oracle and MySQL

In early 2008, MySQL AB, one of the most iconic open source groups and developer of the world's fastest growing open source database (MySQL), was acquired by Sun Microsystems. This move was expected to drive the adoption of MySQL in traditional enterprise applications. MySQL is the "M" in "LAMP" – a software platform that is viewed as the foundation of the Internet.[30] Sun was itself a major player in the open source industry with its OpenSolaris operating system for servers and this acquisition was seen as a positive move by the larger OSS community. It should be noted that this acquisition was not a hybrid strategy in the strictest sense in which a corporation has clear property and control rights and well defined profit targets. Sun contributed to the funding and organization of the MySQL community and obeyed the GPL licensing in the future development of the product.

Oracle acquired Sun in 2009, elbowing aside IBM.[31] Oracle had already made the MySQL community nervous when it acquired Innobase in 2005. Innobase provided the InnoDB Storage Engine to MySQL that enables advanced features for enterprise applications. However, Oracle was bound by the terms of the GPL license to release InnoDB to be used with MySQL. MySQL and the OSS community was not very welcoming of Oracle's acquisition of Sun and there was skepticism about the

[30] "MySQL :: Sun to Acquire MySQL", n.d., http://www.mysql.com/news-and-events/sun-to-acquire-mysql.html.

[31] "Oracle Snatches Sun, Foiling IBM - WSJ.com", n.d., http://online.wsj.com/article/SB124022726514434703.html.

future of MySQL. The MySQL community continued to fight against Oracle even a year after the acquisition. A campaign headed by MySQL founder Monty Widenius created an online petition with the following choice of solutions for the EU commission, which was handling the merger.

1. "MySQL must be divested to a suitable third party that can continue to develop it under the GPL."
2. "Oracle must commit to a linking exception for applications that use MySQL with the client libraries (for all programming languages), for plugins and libmysqld. MySQL itself remains licensed under the GPL."
3. "Oracle must release all past and future versions of MySQL (until December 2012) under the Apache Software License 2.0 or similar permissive license so that developers of applications and derived versions (forks) have flexibility concerning the code."[32]

By late 2010, Oracle responded to the growing fears of the MySQL community with 10 commitments[33] to continue supporting the GPL version of MySQL with improvements[34] and the merger was eventually finalized.

Throughout these series of events there was a sense of crisis within the MySQL community, with one or more forks of the project seeming inevitable (e.g. MariaDB).

[32] "MySQL Community Still Fighting Against Oracle - Softpedia", n.d., http://news.softpedia.com/news/MySQL-Community-Still-Fighting-Against-Oracle-131904.shtml.
[33] "Oracle Makes Commitments to Customers, Developers and Users of MySQL", n.d., http://www.oracle.com/us/corporate/press/042364.
[34] "Oracle pledges MySQL community love • Channel Register", n.d., http://www.channelregister.co.uk/2010/09/20/mysql_oracle_community_pledge/.

An Open Database Alliance was created to coordinate a vendor neutral environment for continuing MySQL development. These events show the pattern of a failing hybrid strategy that is outlined in this paper. Oracle eventually compromised and agreed to support the free community version of MySQL along more OSS friendly terms, shifting away from a more corporate controlled model since it was not viable. The effects of psychological contracts in the OSS community, the vision of liberty and a utopian learning community, the principles of reciprocity and altruism, and the loyalty towards an anthropomorphic idea of OSS all add up to a great force against creation of hybrid strategies. Only companies that demonstrate genuine goodwill and understanding of the core values of the OSS ethos are likely to forge a mutually beneficial contract with an OSS project.

5.2.2 Oracle and OpenOffice.org

Oracle had even worse luck with OpenOffice.org. Oracle initially intended to support the Open Document Format (ODF) that is used by OpenOffice.org but its intentions about the future of the project were unclear. Last month (Apr 2011), the OSS community forked the project and creating an alternative called LibreOffice. They founded a nonprofit organization called The Document Foundation (TDF) in order to create a truly vendor-neutral governance body for the software. LibreOffice is based on the OOo source code, but it also incorporates a large number of other improvements driven by its own developer community Most of the major companies that have historically been involved in OOo development have moved to stand behind TDF and LibreOffice, including Red Hat, Novell, Google, and Canonical.

The ecosystem-wide shift in favor of LibreOffice has left Oracle as the only major party still developing OOo, forcing the company to compete against the broader community. The community defections eventually made OOo financially untenable for Oracle, which is why the company has finally thrown in the towel. Oracle says that it is ready to hand over control of the project to the community, but doing so at this point would be little more than a symbolic gesture; the community has already moved on of its own accord. The LibreOffice escape from Oracle is a powerful demonstration of how open source forking can be used to protect community autonomy and lock out exploitative stakeholders. Several other Oracle open source projects are also declaring independence from the database giant. Oracle's current approach to dealing with the communities that participate in its own open source software projects is clearly not sustainable, and is arguably becoming detrimental to some aspects of the company's long-term business agenda.[35]

This interesting turn of events again validates the concept of cryptohierarchies within OSS communities that exist on the edge of chaos. This unique phenomenon allows the community to easily polarize in the face of an external conflict and resolve the crisis through forking the project. This story is also demonstrates the idea of the OSS community as a decentralized system that can still make critical decisions headed by a strong core group of leaders within each project.

[35] "Oracle gives up on OpenOffice after community forks the project", n.d., http://arstechnica.com/open-source/news/2011/04/oracle-gives-up-on-ooo-after-community-forks-the-project.ars.

References

Bagozzi, Richard P., and Utpal M. Dholakia. "Open Source Software User Communities: A Study of Participation in Linux User Groups." *Management Science* 52, no. 7 (July 1, 2006): 1099-1115.

Baytiyeh, Hoda, and Jay Pfaffman. "Open source software: A community of altruists." *Computers in Human Behavior* 26, no. 6 (November 2010): 1345-1354.

Choi, Chong Ju, Sae Won Kim, and Shui Yu. "Global Ethics of Collective Internet Governance: Intrinsic Motivation and Open Source Software." *Journal of Business Ethics* 90, no. 4 (December 1, 2009): 523-531.

"Fork (software development) - Wikipedia, the free encyclopedia", n.d. https://secure.wikimedia.org/wikipedia/en/wiki/Fork_(software_development)#Forking_free_and_open_source_software.

Karatzogianni, Athina, and George Michaelides. "Cyberconflict at the edge of chaos: Cryptohierarchies and self-organisation in the open-source movement." *Capital & Class*, no. 97 (Spring2009 2009): 143-157.

Lee, Jyh-An. "ORGANIZING THE UNORGANIZED: THE ROLE OF NONPROFIT ORGANIZATIONS IN THE COMMONS COMMUNITIES." *Jurimetrics: The Journal of Law, Science & Technology* 50, no. 3 (Spring2010 2010): 275-327.

"MySQL :: Sun to Acquire MySQL", n.d. http://www.mysql.com/news-and-events/sun-to-acquire-mysql.html.

"MySQL Community Still Fighting Against Oracle - Softpedia", n.d. http://news.softpedia.com/news/MySQL-Community-Still-Fighting-Against-Oracle-131904.shtml.

Oh, Wonseok, and Sangyong Jeon. "Membership Herding and Network Stability in the Open Source Community: The Ising Perspective." *Management Science* 53, no. 7 (July 1, 2007): 1086-1101.

"Oracle gives up on OpenOffice after community forks the project", n.d. http://arstechnica.com/open-source/news/2011/04/oracle-gives-up-on-ooo-after-community-forks-the-project.ars.

"Oracle Makes Commitments to Customers, Developers and Users of MySQL", n.d. http://www.oracle.com/us/corporate/press/042364.

"Oracle pledges MySQL community love • Channel Register", n.d. http://www.channelregister.co.uk/2010/09/20/mysql_oracle_community_pledge/.

"Oracle Snatches Sun, Foiling IBM - WSJ.com", n.d. http://online.wsj.com/article/SB124022726514434703.html.

Roberts, Jeffrey A., Il-Horn Hann, and Sandra A. Slaughter. "Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects." *Management Science* 52, no. 7 (July 1, 2006): 984-999.

Shah, Sonali K. “Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development.” *Management Science* 52, no. 7 (July 1, 2006): 1000-1014.