



Paper to be presented at the
35th DRUID Celebration Conference 2013, Barcelona, Spain, June 17-19

Taming of 'Openness' in software Innovation Systems

Mehmet Gencer

Istanbul Bilgi University
Department of Computer Engineering
mehmetgencer@yahoo.com

Abstract

Open source software is associated with openness in terms of participation to the innovation process. In large-scale open source software innovation ecosystems, however, a variety of measures are taken to tame the potentially chaotic activities and align the contributions of various participants with the strategic priorities of the major stakeholders. Such taming rests on the dual desires to unleash the proven power of open source innovation and to drive it to a certain direction, and emerges in the form of a variety of governance and licensing practices, organizational activities and also encompasses certain common themes. By drawing on a sample of large scale open source software ecosystems, we disclose that methods employed for taming are isomorphic, and suggest that the software industry is settling on a strategic pattern for establishing systems of innovation, and related set of practices which retain virtues of open source innovation while allowing major stakeholders investing into the ecosystem to selectively support activities, and introduce top-down corporate discipline such as quality assurance measures. We argue that such innovation systems and legitimation of practices for their establishment cannot be understood with approaches such as open innovation, which focus on bilateral arrangements and favor isolated reasoning to a systemic and historical perspective. Instead they should be considered against the community's developed appreciation for collaborative innovation systems and practices that are proven to sustain it.

Taming of 'Openness' in Software Innovation Systems

Abstract

Open source software is associated with openness in terms of participation to the innovation process. In large-scale open source software innovation ecosystems, however, a variety of measures are taken to tame the potentially chaotic activities and align the contributions of various participants with the strategic priorities of the major stakeholders. Such taming rests on the dual desires to unleash the proven power of open source innovation and to drive it to a certain direction, and emerges in the form of a variety of governance and licensing practices, organizational activities and also encompasses certain common themes. By drawing on a sample of large scale open source software ecosystems, we disclose that methods employed for taming are isomorphic, and suggest that the software industry is settling on a strategic pattern for establishing systems of innovation, and related set of practices which retain virtues of open source innovation while allowing major stakeholders investing into the ecosystem to selectively support activities, and introduce top-down corporate discipline such as quality assurance measures. We argue that such innovation systems and legitimation of practices for their establishment cannot be understood with approaches such as open innovation, which focus on bilateral arrangements and favor isolated reasoning to a systemic and historical perspective. Instead they should be considered against the community's developed appreciation for collaborative innovation systems and practices that are proven to sustain it.

Keywords: Open source software; Innovation ecosystems; Strategy-as-practice

1 Introduction: the case for taming openness

Collaborative innovation is becoming increasingly commonplace in many industries, ranging from biotechnology to computer hardware/software. Many organizations are struggling with volatility in consumer demands and competition, and try to respond in a creative and agile manner. The consequences, as it appears, frequently involve seeking innovation beyond its traditional locus of internal R&D departments. Accordingly, various organizations pursue bilateral or multilateral arrangements to actively cultivate inter-organizational networks of collaborative innovation.

For firms in volatile markets, acknowledging the need to reach out to networks is a first move that open up new challenges. Most of these firms have established practices and strategies based on closed forms of innovation. Some have experience with what is recently named as open innovation, which involves exchanges with other entities in the environment (firms, research institutions, etc.), often using established forms of legally binding bilateral arrangements such as licensing. However networks defy existing bilateral forms to enable their participants to get organized with one another easily. Firms intending to benefit from networked innovation face various challenges regarding how to reason about, interact with, and profit from the networks they become part of. Networking is a chaotic forefront of innovation which is seeking organizational forms of its own, through ongoing trials and errors of its current adopters.

Among several other industries, software industry has been one in which attitude towards collaborative innovation strategies are changing rapidly. It seems that 'adaptation' is replacing 'planning' as the fundamental principle of innovation in software technology development. Within this changing mindset, Open Source Software (OSS), once a marginal movement, is finding its way into mainstream inter-organizational innovation practice in the software industry. OSS provides proven methodologies which stimulate software product evolution at the expense of process

predictability (as opposed to the established practices of software engineering), and promote product interoperability which facilitate a more efficient coordination medium (Behlendorf 1999; Benkler, 2005; Ritala, 2001). Leading players in the computing industry, such as IBM, Apple, and Google, actively cultivate open source innovation networks and/or use software technologies coming out of these networks as components in their key products or operations (West 2003).

However, despite the rhetoric of 'openness' surrounding these practices, considerable effort is vested in to tame the 'mess'. Within the multitude of often conflicting business agendas in a collaborative innovation system, each participant faces the dilemma of converting collaborative outcomes into competitive advantage. Accessing knowledge resources through networks overcomes rigidities of innovating in isolation (Jorde and Teece 1989; Ring and van de Ven 1994; Leonard-Barton 1992; von Hippel 2006). However, a firm expects some predictability, and would like to drive this distributed and emergent, rather than centralized and planned innovation process in a direction that makes sense for its own priorities. While 'openness' in OSS is associated with unleashing the bottom-up innovation, blending and aligning it with a business strategy unavoidably faces the dilemma of re-leashing this innovative power so that it is manageable in a 'top-down' way and becomes useful 'business-wise'.

In this exploratory study we examine how an institutional field (OSS) or a community organized as "collaborative innovation model" (von Hippel and von Krogh 2003) has changed towards a hybrid innovation model. Our focus is to lay out the elements of an emerging practice, how these elements address the problems of agents (firms) in collaborative software innovation, and consequently, how institutional acceptance about and suitability of these elements is verified. In doing so, we study the role of collective agency in selecting and implementing certain activities that trigger the initiation and maintenance of new practices. The major theoretical concern of the paper is to explore how incrementally introduced new practices lead to the transformation of an institutional field. In this vein, we investigate several cases of 'community-led' and 'business-led' collaborative innovation projects based on OSS, and attempt to identify common patterns in organizational structures and processes which address the duality of empowering bottom-up innovation while at the same time imposing requirements on its direction and outcomes in accordance with priorities of various stakeholders. More specifically, we explore how practices like quality assurance, licensing, governance, project development and developer promotion techniques were introduced to an innovation model and how they were accepted and established which in turn leads to the transformation of the institutional field.

One needs to look in vein to find a theoretical yardstick which can help to interpret emergence of open source innovation systems in the way they are. Open innovation framework has little to offer towards making sense of governance in such systems, for its take on strategy as essentially a firm centered activity and thus its lack of interest at the system level. Open source literature, on the other hand, offers a variety of findings for what motivates open source communities, but it is generally void of insights about why they are manifested through certain systemic elements rather than others.

Here, we develop a perspective based on strategy-as-practice approach, as it can address both strategic concerns of firms participating in open source software innovation systems, and why certain strategic practices (Whittington 2007) are available for mutual, multilateral acceptance, while others are not. The sample cases in our data indicate that recent corporate-led open source innovation systems legitimize themselves by drawing from practices developed in community-led systems. Once a pattern of practices are worked out -by incremental improvements- for the new strategic challenges in the new class of systems, these practices are imitated in new ones. Governance mechanisms and regimes developed in one innovation system are reproduced in other systems without the need for exogenous policies and policy-makers imposing them. In this sense, open source software innovation today appears to be an avant-garde class of inter-organizational systems which, by the virtue of owing its advantages to aggregation of actors and resources, well worth as a case in systems of innovation research on the one hand, but does not fit easily in the related literature which portrays such systems as entities which are shaped by exogenous

regulations and policies, in addition to weak interest in the research stream on virtual, geographically dispersed systems.

In section 2 we present a brief overview of literature related to collaborative innovation in general, and OSS in particular. Section 2 develops a perspective on OSS based on strategy-as-practice theory. Based on this theoretical backdrop, we present the research design and data, followed by our findings in sections 4 and 5, respectively. Building on well-known, large-scale OSS-based collaborative innovation projects, we identify common mechanisms and practices used for taming the ‘openness’ in terms of resource mobilization processes in relation to a proposed model of institutional change. In order to have a more refined analysis both success and failure cases are incorporated to the study. Finally, in section 6, we discuss how the institutional field in prospect is converging around a certain set of practices and organizational mechanisms for taming of openness, followed by conclusions.

2 Collaborative innovation and Open Source Software

In addressing how innovations are managed and how they create competitive advantage, Teece (1986; 1989) questions the role of in-house R&D as the driver of innovation; by placing design, rather than science, at the center of innovation process, Teece (1986; 1989) suggests that richness of collaborative linkages with other organizations (firms, universities) is the major driver of innovation capability. Corporate R&D work within firm boundaries complements this capability, often by introducing modest improvements on innovations which appeared elsewhere (Cohen and Levinthal 1990). Another stream of research focusing on inter-firm networks and their role in innovation conceptualizes network linkages and positioning as important factors in utilizing complementary assets for creating and sustaining competitive advantage in the presence of imitators. In this vein, inter-firm networks are ascribed roles ranging from being a prerequisite for commercializing innovations (Kogut, 2000) to itself being the locus of innovation (Powell et al., 1996). Within this networked view of the firm, various aspects of economic performance are subjected to empirical research: structure of firm network ties in relation to its innovation outcome (Ahuja, 2000), inter-organizational learning (Powell et al., 1996; Lane and Lubatkin, 1998; Nooteboom, 2008) and the role of network ties for firm survival (Oliver, 1994).

Despite the awareness about advantages of innovation networking, the business environment is at an infancy phase in terms of institutionalization of collaborative innovation networks (Whittington et al. 2011). Collaborative innovation is prone to a variety of problems; the initiation of inter-firm relations relies on a variety of antecedent factors (Oba and Semercioz, 2005), and development of relations and mutual trust that sustain them is hard to predict or control (Ring and van de Ven, 1994). Developing collaborative relations entail sharing critical knowledge (Gächter et al., 2010) and taking risks as committing to relation specific investments. Additionally, some research addresses the issue with an inter-organizational perspective, considering a range of problems such as the content and structure of inter-organizational networks of innovation and production in technological systems or districts (Langlois, 1990; Castilla et al., 2000; Antonelli, 2000), and adverse effects of protective regimes for realizing collaborative innovation in such systems (Heller and Eisenberg, 1998; Pisano, 2006; West and Gallagher, 2006). However, in spite of such problems, failure to establish collaborative innovation relations often results in the underutilization of innovations created inside the firm (Chesbrough, 2006; Chesbrough and Appleyard, 2007) and traditional business strategy of constructing barriers to prevent knowledge leaks also prevents knowledge intake and impedes combination of capabilities.

Designing inter-organizational innovation systems remains an experimental innovation task in itself; governance and coordination mechanisms, property regimes that enable such networks function in a way which reduces risks of participants and increases predictability of outcomes has to be developed. Furthermore, such an organizational design task is expected to engender a solution to a new problem but it must be recognizable, understandable and legitimate to all participants

(Krippendorff, 1989). In such a situation where things are uncertain but collective decisions are to be made, one must resort to proven solutions of the past that make immediate sense for everyone. Thus, organizational designs of collaborative innovation networks may have different starting points and evolutionary paths depending on the industry and its established practices, although they solve similar problems. Here we will only consider the case of software industry and open source software.

OSS is both widely cited in collaborative innovation literature, and at the same time treated specially. There are a variety of reasons for this special treatment. First, OSS has developed outside the business world, in communities spanning academia and young professionals (McKusick, 1999; Raymond, 1999). As it has found its way into business realm in the software industry, it did come with a baggage of historically developed methods and property regimes of its own. Thus, it exhibits a different history in its second life which involves experimentation of firms to adapt it into a commercially viable form. Second, the software field generally lacks patents which are the norm of intellectual property protection in most industries. Therefore software source code can no longer be protected against appropriation once it is disclosed. For this reason, OSS based collaborative innovation networks use commons based property regimes rather than bilateral or multilateral licensing arrangements common in similar networks in other industries. The so called open source licenses essentially give the right to use software to everyone. The historical license type of OSS was the 'public' license, which further mandated disclosure of any modifications to a public licensed software. However, years of experimentation with OSS in the software industry resulted in widespread use of more liberal licenses by firms which encourage collaboration while allowing usage in commercial products in the traditional sense. The model based on liberal licenses is commonly referred to as open source software, where the one based on more restrictive licenses is referred to as free software. In both models, the commons based property regimes combined with transparency of OSS software development method makes OSS an area where the openness in innovation is very high in terms of both the processes and the outcomes (Huizingh, 2010).

A considerable share of research on OSS focuses on community-led projects, which is the representative form of both its pre-business history and early OSS business models. These studies explored internal activities of OSS communities such as motivations of individuals for contributing to public software (Hertel, 2003; Lakhani and Wolf, 2005; Lakhani and von Hippel, 2003; von Hippel, 2005) and highlighted meritocratic leadership and community norms as the basis of coordination and control (Kogut and Metiu, 2001). Another stream of OSS research investigated recent business experimentation with the OSS model (West, 2003; Samuelson, 2006), business models underlying these experiments (Krishnamurthy, 2005), and implications of the rise of OSS model for software industry (von Hippel, 2001; Bonaccorsi et al., 2006; Economides and Katsamakas, 2006; Pykäläinen 2007). More recently, several researchers claimed a fundamental shift in the software industry as a result of OSS model's popularity, using various labels for the phenomenon such as 'OSS version 2' (Fitzgerald, 2006) and 'OSS generation 2' (Watson et al., 2008). Although limited in scope, some recent empirical findings (Hauge et al., 2008) seem to support Fitzgerald's (2006) claim that these developments signify the end of proprietary model and a paradigm shift in the software industry.

The research literature on OSS is limited in terms of understanding emerging forms and processes in OSS based collaborative innovation in relation to business strategies. Open innovation framework, for example, has an ontological basis which consists of the organizations, their relations and knowledge content of these relations, which allows exploring competitive and collaborative issues simultaneously. However, it is often the case that organizing collaborative innovation requires creation of practices and even formal entities for the governance of such practices, which are associated with the network itself, rather than particular actors or relations. What is highlighted by the community oriented approach of some OSS research becomes invisible under the actor oriented approach of open innovation framework, and vice versa. This may well be the reason behind recent preference of the term 'ecosystem' over 'network' in OSS research jargon which is

more conducive to speak about content and design of inter-firm collaborative innovation networks while blending relatively well with firm-centered open innovation thinking. OSS model provided the software industry with a design model which made sense to potential participants of innovation ecosystems that are to be established. Increasing numbers of such systems appearing since 2000s have mimicked the original model and its variations, while introducing incremental changes to make it more suitable to business environment. What is copied or modified includes property regimes, governance mechanisms, coordination processes, and even mutual expectations of the participants.

3 Innovation Practice and Open Source Innovation Systems

OSS community brings together geographically dispersed actors with diverse interests around a common issue: innovation in software. As a situated activity, innovation practice both shapes and is shaped by various activities of the OSS community and its situated actors (e.g. firms and developers) cannot be decoupled from the context in which they act. In this vein, the study of how innovation practice is constructed within the community must take into consideration the social situation that provides embedded norms of conduct and the micro interpretations of the same social situation by the agents (Jarzabowski 2005).

OSS practices have been shaped within various prominent projects, some of which are relatively well documented. These somewhat intertwined projects, and recently virtual communities exhibit basic properties of innovation systems as they sustain themselves due to aggregation benefits (Dodgson et al. 2008) they offer to their participants. In recent years, commercial firms have started to become part of OSS and initiated numerous projects of varying scales. While the benefits to commercial firms are not much different (e.g. sharing costs of innovation by the help of aggregation), they do have intrinsic criteria when it comes to capturing value created in OSS innovation systems, such as ensuring legal conditions (software licensing) for capturing value they have helped to create, or assuring quality of software partially built by others, but they will be liable for once it is sold. Therefore a fundamental shift in the requirements and expectations of the practitioners calls for an institutional change in OSS innovation.

This interplay between context (macro) and actors (micro) in the construction of innovation practice reflects the praxis which is the basis of such a reconstruction process and endorses a deeper understanding of the institutional change processes. Following Benson's (1977, p: 5) definition we characterize praxis as a "collective human action, situated in a given social and historical context but also driven by the inevitable by-products of that context-social contradictions."

In such collective action domains, institutional change is a complex social process which requires collective action of numerous actors. In the process of such a change, besides the purposive actions of multiple actors, structures and processes embedded within the collaboration network play a role in initiating new and disrupting existing practices (Wijen and Ansari, 2007). Thus, we will be utilizing a practice-based approach in order to explore how certain innovation related activities like recruitment of developers, development of new projects and decision making can trigger the transformation of the existing innovation practice and lead to the transformation of an ecosystem. Accordingly, we study innovation as a situated activity which is always constructed and reconstructed by the "practical-evaluative agency" (Emirbayer and Mische 1998).

This process of transforming the practice of innovation can follow various courses; change, recursiveness and stabilization (Jarzabowski 2005). This paper focuses on the change course of innovation practice and its construction among multiple, distributed actors (mainly developers and firms). Distributed actors while pursuing their own divergent interests reason, question, discuss and challenge the existing innovation practice; as active agents they exploit social contradictions, tensions, conflicts between the system they are part of and the society. This very process of

contestation makes the practice itself prone to fragmentation. Thus, various “structural” and “interpretive” actions must be generated to align the interests of distributed actors so that a collective action can be generated. This study focuses on these activities that lead to the transformation and change of the innovation practice in OSS community; a change which goes through breaks and reunions with institutional roots (Chia 1999). Under such conditions the innovation practice can be explained as the skilled ability to utilize, alter and direct resources for shaping the practice.

Such an approach to change deems the study of prevailing community (its structure and philosophy), goal-directed behavior of agents (firms), and the interaction between these two levels. In this framework agents are taken as practical-evaluative, future oriented and iterative. Their actions are mediated through structuring activities like governance, recruitment, quality assurance, and developing mechanisms for decision making which enable interaction between them and the community they are part of. These mediating activities also enable to coordinate distributed actions and contributions so that collective action is achieved. In this vein OSS is conceptualized as an activity system comprised of actors, their community and the goal directed innovation practice in which they are involved.

4 Research Design and Data

This exploratory study draws on a qualitative analysis of success and failure cases for explaining transformation in an institutionalized field-OSS. In so doing we aim to elaborate theoretical concerns related to the transformation of institutional fields and the role of various activities in this process. More specifically, our aim in this paper is to explore the transformation of inter-organizational open source innovation systems and their practices in software industry and to identify the elements of the emerging system. Research in the field has already identified how open source approach is increasingly influential in inter-organizational collaborations in software innovation (Fitzgerald, 2006, Hauge et al., 2008, Watson et al., 2008). In order to explore the transformation process we utilize the framework developed by Jarzabowski (2005) which is comprised of agents, community, their goal-directed behavior and mediating practices (Figure 1).

Insert figure 1 about here

As goal-directed activities we have studied staffing, recruitment, decision making, structuring and quality assurance because these activities are related to (i) understanding of inter-organizational sharing and coordination of responsibilities/outcomes, (ii) making common sense and appealing to the participants for collective decision making, and (iii) influencing the goals or outcomes of participant firms that promote the reconstruction of existing arrangements. Decision making in OSS projects is commonly understood in terms of either (i) setting the major design goals of software product by selecting among proposed features, or even creating sub-projects (i.e. bundles of features) when necessary, or (ii) promoting some contributors to ‘committer’ status who has the right to change or accept changes to master copy of software source code. The way decisions are made and their success/failure is related to human resources implementations like recruiting and staffing. Quality assurance in OSS projects evolves in the absence of a central authority to verify and evaluate the value of the products and services produced. Other activities we have included in our analysis also have a mediating role. In this vein we have identified property regime and formal governance mechanisms because they provide mechanisms that facilitate the transformation of the existing regime. Since license differences can introduce serious legal issues (de Laat, 2005), most projects use only one OSS license, or at best a few closely compatible ones.

As agents, we have identified six open source projects which formed our sample. These are all

large scale projects that sufficiently expose inter-organizational tensions, and due to their popularity, reliable first-hand and anecdotal evidence is available. We have included cases from two major categories of community-led and business-led OSS projects. With regard to OSS history, community-led projects in our sample predate business-led ones, and such mix further highlights the elements of OSS transformation. However, the categories are not always exclusive: some OSS projects start their lives as small non-commercial projects led by a small community of developers, but later meet corporate interest. Some other projects are the other way around: they are initiated by one or a group of firms, but later may enjoy support of individuals from professional communities.

Complimentary to the above, we have also selected a smaller set of cases of 'failure to embrace open source model'. Although limited in scope and extend, these cases are used to check for position of the OSS model within the software industry. This set consists of prominent firms in the computing industry which are strained, even failed, in the face of emergent OSS model. Case summaries presented in Tables 1 and 2 aim to capture brief historical background of competitive (dis)advantage in relation to OSS related developments in relevant market segments.

For both groups of cases (the six OSS projects and failure cases) , we have collected secondary data which incorporates published documents like public statements, web pages of related organizations, announcements, software release logs, developer guidelines and committer policies..

The data analysis of this study is comprised of three phases. Firstly, we analyzed cases to identify the innovation related activities in each and came up with a list of such elements and later these lists were redesigned to include only those activities that are common in all the cases comprising the sample. In the second phase we examined each element across all the cases, to identify how and why it has emerged and what were the subsequent outcomes. During this phase we also checked how these elements interacted with each other and how they differ between community-led and corporate-led projects. And finally in the third phase, we analyzed how these activities contributed to the transformation of the governance of innovation in particular and governance in general in OSS community. In this way we were able to identify the characteristics of the emerging governance structure and practices for organizing inter-organizational OSS innovation systems.

Insert table 1 about here

Insert Table 2 about here

5 Findings

In the first phase analysis of success cases we have identified six generic elements that are related to firm participation and commonly observed in all cases: licensing, governance mechanism, acceptance policy of new (sub)projects (goal setting), human resources, corporate leadership, and quality assurance activities (Figure 1). The emergence of these elements is summarized in Table 1 for each case.

The community-led and corporate-led groups indeed had common characteristics within group. The group characteristics and differences between the two groups are given in Table 3. Below, we first summarize similarities and differences between these groups, and then discuss the failure cases. Following this summary we discuss the emergence of characteristics in the more recent group of corporate-led OSS projects in order to identify how the taming of OSS proceeds.

Insert Table 3 about here

5.1 Similarities

One common element among the community and corporate-led projects is the *human resource activity* based on meritocratic promotion of developers. Meritocratic promotion is seen as an important basis of OSS model and seems to be fully adopted in corporate-led OSS projects as well. Besides the somewhat exceptional case of Android, both community-led and corporate-led projects adopt voting-based schemes for developer promotion. Contributors who demonstrate skills and adapt to community norms are voted into ranks by existing committers. Norms (re)articulated by existing meritocracy implementation thus becomes an important element of taming within corporate-led collaborative innovation systems.

In terms of *quality assurance*, OSS is commonly praised as hosting processes in which software quickly evolves into high quality products despite the lack of central control which empowers it. Research on the OSS quality assurance practices is rare (Zhao and Elbaum, 2003). The general quality assurance strategy in OSS has been to transfer testing and bug reporting activities to software user base. Many OSS projects make frequent releases of ‘testing’ versions of software, which in turn put into use by a large group of enthusiasts and those who are in need of new features not yet offered in ‘stable’ releases of software. Existing evidence (Zhao and Elbaum, 2003) indicate that frequent releases is often the case for OSS projects.

We have collected development guidelines from the case projects. These evidences suggest that the community OSS style of quality assurance is pretty much continued in all types of projects, however with certain differences related to use cases and organization types. All the cases followed the OSS pattern of releasing frequently. Community projects such as Linux and GCC, and to some degree Eclipse and Mozilla, follow the pattern of releases marked as ‘testing’ or ‘release candidate’. These releases are targeted on part of the user community who are willing to take the risk of using buggy software (in many cases for the sake of accessing new functionality) and provide feedback/bug-reports which in turn assist the developers to carry the software quality to a more mature level. In addition, both of these projects have clear (even formal) test plans and procedures. In the case of Apache, which is a collection of numerous sub-projects, the quality assurance activities exhibit certain variation. At the other extreme, in Android, the development of the core system is pretty much closed within Google and the quality assurance activities appear to be similar to closed source ones, i.e. they are shaped by the fact that the product goes live on millions of mobile devices of non-tester users. Although they make frequent releases much like OSS style, they work hard (behind closed doors) to ensure that these are stable software. On the other hand the Android ecosystem consists of thousands of other, higher level pieces of software (Android applications), which are developed by partners and are not subjected to such central quality assurance policies.

One noteworthy fact in quality assurance activities is that as the commercial orientation comes into the play, in corporate-led cases, quality assurance necessarily includes additional steps to ensure that contributions to software meet the license framework and legal requirements of the project related to commercialization of the software technology. This is the case with Apache and Eclipse. These projects ensure the code contributions go through several steps related to software quality assurance (automated tests, dependency checks, etc.) and also license sanitation. Overall the case of quality assurance in both firm-led and community-led projects seem to retain proven OSS methods (much like the case for promoting developers) and extend it with context specific requirements.

5.2 Differences

The **creation of new sub-projects** is handled in a more ‘transparent’ and ‘planned’ way in corporate-led projects where the process is given names as ‘incubation period/process’. In Apache and Eclipse examples, this period involves tempering of new project and its leadership to make

processes and software products meet certain quality standards. In community-led projects, on the other hand, all is decided by meritocratic leadership. There may even be cases where leadership makes decisions despite strong opposition (Gençer et al., 2006).

The difference in *property regimes* and corresponding licensing schemes reflect the difference of concerns with wider adoption (in community-led type) and suitability for use in commercial products (in corporate-led type). The experience seems to have shown that replacement of intellectual property hostile regime of free software movement with more liberal licenses is generally welcomed. Despite minor variations liberal licenses (with archetypes such as Apache and Mozilla licenses) seem to be the established norm for new corporate-led OSS projects.

In comparing the two types of innovation regimes, it is interesting to note that community led projects are less transparent in terms of *governance*. This situation may look somewhat counter-intuitive at first, but there are business strategic advantages of transparent governance in an inter-organizational setting. Such transparency serves well for directing community attention and setting the goals. Fitzgerald (2006), notes how such transparency has been utilitarian for avoiding the strategic planning vacuum of community-led projects, in which software development followed ‘an itch worth scratching’. Corporate-led projects aim to be inclusive and flexible in their planning process, but nevertheless such planning vacuum is avoided by careful outlining of targeted software features and development road-maps, and making them widely available. Hence, using transparency as an instrument has been valuable for directing community attention and setting goals effectively.

Similar to governance mechanisms, *corporate influence* is more transparent and works through formal bureaucracy in corporate-led projects, as in the example of Eclipse where each major participant dedicate certain amount of human resources (among financial resources) to the project and have seats in decision-making bodies. On the other hand corporate influence enters community-led projects by means of firms hiring lead developers or having seats in governance committees. Generally, as Watson (2008) notes it’s rather the amount of contribution rather than formal ownership that is the basis of control and influence in community led projects.

5.3 Failure cases and Position of OSS Model

It is hard to substantiate an argument that OSS model invalidates the previously prevalent proprietary software model, based on isolated cases. However, they do indicate significance of this claim voiced by various researchers (Fitzgerald, 2006; Watson et al., 2008). Among our sample, all are cases in which use of OSS based collaborative innovation strategy in the relevant industry segment created a vacuum which left ostensibly established players who do not adapt that strategy in a disadvantaged competitive position.

Our findings about cases of firm failure are summarized in Table 2. Among these is Sun Microsystems, which has been a major player in the computing industry for three decades. Java technology has been the flagship product of company recently. However, Sun’s Java based products and services have failed to compete with more innovative alternatives in the market, many of which came from competitors who adopted OSS based innovation strategies. Unlike Sun Microsystems which was sold to another company due to serious problems, Microsoft remains a very successful player in the industry. But its market position is seriously challenged by other players, many of whom embrace either open innovation or open platform approaches. In particular the company has failed to position itself in the growing mobile market which is now dominated by the Android technology that is led by Google and uses open source approach. Similarly Nokia, the leader in mobile appliance market, has found itself in serious trouble in the face of emerging Android dominance. The case of Nokia particularly highlights that collaborative innovation strategy based on non-OSS models fail to expand in the face of OSS-based alternatives, despite first-mover advantages.

6 *Emergent Practices for Taming Openness*

The findings of our study provide a picture of what is being tamed in business adoption of OSS, and in what direction the process heads. Most noticeable OSS element that is left intact in taming is meritocratic promotion of developers. Watson et al. (2008) note how OSS enables private businesses to access high quality talent base, and helps reduce hiring risks. Thus retaining the meritocratic promotion does not only keep program development quality high, but also provide access to talent that is relevant. In similar vein, adjustments to OSS licensing (see de Laat 2005 for a review) was kept at a point where it both attracts high quality and increasing volume of contributions while at the same time allowing them to be used in commercial products.

On the other hand, formal governance practice and goal setting activity are specific to corporatized era of OSS and appears to be still experimental. Some popular success stories like Eclipse are likely to set emerging norms in this respect. Furthermore, comparison of the Android case to others like Eclipse and Apache suggest that expanding the support base of collaborative innovation is associated with increasing transparency in governance structures and practices. Such transparency not only improves trust among partners, but also makes the process more predictable. In addition it allows participants to cast a direction to goals and outcomes of the collective process, thus contributing to the taming of OSS. Increasing use of formal governance bodies and practices in OSS-based collaborative innovation introduces an organizational innovation into the software industry. In a business environment where there are established bilateral forms such as joint ventures, such multilateral forms are rather new. This phenomenon further highlights the need for ontological commitment to network in inter-organizational collaboration research, since the networks start to become concrete entities with some sort of controlling bodies and norms of their own.

Studies on collaborative innovation mainly focus on the ability of a firm to attract ideas and resources from external actors and appropriate benefits of innovation. As indicated by Langlois (1990) firms are pursuing ways to improve innovation performance beyond the hierarchical structures and thus, mix of new organizational forms are emerging. Studies such as van der Linden et al. (2009) present economic frameworks within which the OSS innovation model is being employed by companies. During the last decade, research reports such as Linden et al. and Fitzgerald (2006) appeared in professional and academic media that appeal to mainstream practice, unlike the preceding decade where OSS-related literature was rather marginal. Such literature offers a retrospective account of the open source experience in economic terms such as value creation chain and differentiation. While we lack any extensive evidence at the global level, there are regional studies whose results support that adoption of OSS model is on the rise (Hauge et al., 2008; Nikula and Jantunen, 2005).

Accordingly, OSS is an appropriate case to study how (due to endogenous and exogenous pressures) innovation regimes can incorporate some elements of other innovation regimes. In the background of these developments is the increasing speed of innovation, driven by variety and volume of demand, high connectivity of devices, and high degree of transferability of information. This makes it hard for even the biggest players to control or supply complex products on their own. When a complex system such as the software industry is on the verge of a fundamental transformation, various marginal practices can become common as they fit well into the new environment. Furthermore, these market pressures triggered the software ecosystem where different types of innovation and governance regimes had to cooperate in order to cope with the emergent changes. Innovation regimes build on in-house software production and revenue generation by patented products gradually has adopted elements of the OSS system. Traditionally, the major concern of firms operating in this innovation regime was whether to buy or make the software and recently this problem has been transformed to collaborate or not; and if the decision is to collaborate then how the relations with the partners should be governed so that efficiency and appropriation

would be achieved. The consequences of such a transformation unavoidably lead to transformation of the OSS innovation regime as well. Some authors have already pointed to fundamental shifts in OSS processes, and noted that these indicate alteration of ground rules in both the OSS movement and the software industry (Fitzgerald, 2006). Fitzgerald also explains how ‘OSS 2.0’ balances “a commercial profit value-for-money proposition while still adhering to acceptable open source community” (2006 p: 588).

Our findings extend this argument and present the organizational elements which are associated with the shifts in the open source process towards more corporate involvement. There appears to be an emergent, stable pattern of proven mechanisms, strategies, and processes. Thus, the transformation indicated by Fitzgerald's OSS2.0 is institutionalized into common workable governance mechanisms that emerged from the daily activities and enabled participant market actors to set-up collaborative systems. Furthermore, emergence of such innovation and governance practices seems to be isomorphic; since they make common sense in most situations and practical, actors do not seek other ways of doing things, and avoid to introduce disturbance of the respective systems. We suggest that emergence of the transformation mentioned by Fitzgerald (2006) has already been institutionalized and legitimated, hence will soon lead to faster adoption of OSS strategies.

We call for due attention to the state of the change and transformation in the software industry. Using a term like OSS 2.0 indicates a fundamental change in the OSS, but diverts attention from the business-wise strategic roots of the overall change it intends to convey. Furthermore, it potentially oversees phases of emergence of patterns of inter-organizational design for collaborative innovation, following disruption introduced by OSS (such as legitimation and stabilization of practices). The new hybrid innovation regime incorporates activities and mediating practices of previously existing innovation regimes. Such a new design has developed from existing patterns and practices, in incremental steps such that each modification is recognizable and makes sense to stakeholders. Yet, the emergent attractor is radically different from its dual origins (of OSS and proprietary models). Its elements are different from those common in corporate hierarchies; rather than targeting a top-down control or planning in the strict sense, they are instruments of taming. It is through these elements that an OSS ecosystem continues to be an emergent system which retain advantages of bottom-up innovation, while at the same time its major stake-holders can “invite it to emerge”(Jelinek, 2004) in certain ways rather than others.

Greater acknowledgment of the need for collaborative innovation is one part of the disruption in the software industry, and OSS seems to be the organizational model of choice. Its emergence is probably triggered by the success of popular community products such as Linux and Apache, but its true admission to business strategy practice went through a series of experiments; and it went all well. As far as this emergent combination of solutions work, it can be expected to be the first choice of collaborative innovation ecosystems which will be established in software industry in near future. Nokia's Symbian project might as well continue to be as innovative as Android, but the players of the industry seem to relate much more easily to OSS model at the moment. Phrasing the current situation rather dramatically, OSS makes a lot of sense, and makes all others look like nonsense.

The state of affairs in software innovation indicates that free software and open source software threads reaching a settlement in which their focus shifts to co-endowment rather than arguing over their differences. The community now enjoys more interest and more resources, while the corporate movement keeps finding ways to turn community experience to viable business practices.

7 Conclusion

In this study we have focused on emergence and role of practices and governance structures that are associated with taming of collaborative innovation networks based on OSS model. We have used several cases of community led and corporate led OSS ecosystems, and attempted to identify common elements in organizational structures and processes We contend that community led and

corporate led OSS ecosystems differ, where (1) practices and formal arrangements for taming is more explicit and transparent in the latter, (2) meritocratic basis of developer promotion is retained in both, (3) means of asserting influence by corporations is different in both, but considered legitimate as far as the meritocratic basis is retained. By also referring to cases of failure to embrace OSS innovation and its negative consequences, we further suggest that the computing industry is converging upon OSS innovation.

References

- Ahuja, G.(2000). 'Collaboration networks, structural holes, and innovation: A longitudinal study', *Administrative Science Quarterly*, 45, 425–455.
- Antonelli, C. (2000). 'Collective knowledge communication and innovation: The evidence of technological districts'. *Regional Studies: The Journal of the Regional Studies Association*, August, 34, 535–547.
- Behlendorf, B. (1999). 'Open Source as a Business Strategy', in DiBona, C., Ockman, S., Stone, M. (Eds.) *Open Sources: Voices from the Open Source Revolution*, pp 149-170. O'Reilly.
- Benkler, Y. (2005). 'Coase's penguin, or, linux and the nature of the firm', in R. A. Ghosh (Ed.), *CODE : Collaborative Ownership and the Digital Economy*, pp 169–206. Cambridge: The MIT Press.
- Benson, J.K. (1977). 'Organizations: A dialectical view', *Administrative Science Quarterly*. 22(1), 1-21.
- Bonaccorsi, A., Giannangeli, S., Rossi, C. (2006). 'Entry strategies under competing standards: Hybrid business models in the open source software industry'. *Management Science*. 52(7), 1085–1098.
- Castilla, E. J., Hwang, H., Granovetter, E. , Granovetter, M. (2000). 'Social Networks in Silicon Valley', in Lee, C.M., Miller, W.F., Hancock, M.G., Rowen, H. (Eds.), *The Silicon Valley Edge: a habitat for innovation and entrepreneurship*, pp 218–424. Stanford University Press.
- Chesbrough, H. (2006). 'Open Innovation: A new paradigm for understanding industrial innovation', in Chesbrough, H., Vanhaverbeke, W. (Eds.), *Open Innovation: Reaching a new paradigm*, pp. 1-14. Oxford University Press.
- Chesbrough, H. W., Appleyard, M. M. (2007). 'Open innovation and strategy', *California Management Review*. 50(1), 57–76.
- Chia, R. (1999). "A 'Rhizomic' Model of Organizational Change and Transformation: Perspective from a Metaphysics of Change", *British Journal of management*, 10, pp 209-227
- Cohen, W. M., Levinthal, D.A. (1990). 'Absorptive Capacity: A New Perspective on Learning and Innovation', *Administrative Science Quarterly*. 35(1), 128-152
- de Laat, P. B., 2005. Copyright or copyleft?: An analysis of property regimes for software development. *Research Policy*. 34(10), 1511–1532.
- DiBona, C.; Ockman, S. & Stone, M. (1999), *Open Sources: Voices from the Open Source Revolution*, in *Open Sources: Voices from the Open Source Revolution* , DiBona, C.; Ockman, S. & Stone, M. (Eds.), O'Reilly.
- Dodgson, M.; Gann, D. M. & Salter, A. The Management of Technological Innovation: Strategy and Practice Oxford University Press, USA, 2008*
- Economides, N., Katsamakas, E. (2006). 'Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry', *Management Science*. 52(7), 1057–1071.
- Emirbayer, M., Mische, A. (1998). 'What is agency?', *The American Journal of Sociology*. 103,

962-1023.

Fitzgerald, B. (2006). 'The transformation of open source software', *MIS Quarterly*. 30(3), 587 – 598.

Gächter, S., von Krogh, G., Haefliger, S. (2010). 'Initiating private-collective innovation: The fragility of knowledge sharing', *Research Policy*. 39, 893-906.

Gençer, M., Oba, B., Özel, B., Tunalioglu, V.S. (2006). 'Forking: The GPL coherent technology for flexible organizing in foss development', *European Group of Organizational Studies 2006 Colloquium in Bergen, Norway*.

Hauge, O., Sorensen, C.F., Conradi, R. (2008). 'Adoption of open source in the software industry', in Russo, B., Damiani, E., Hissam, S., Lundell, B., Succi, G. (Eds.), *Open Source Development, Communities and Quality*, Volume 275 of IFIP International Federation for Information Processing, pp 211–221. Springer Boston.

Heller, M. A., Eisenberg, R. S. (1998). 'Can patents deter innovation? the anticommons in biomedical research', *Science*. 280(5364), 698–701.

Hertel, G. (2003). 'Motivation of software developers in open source projects: an internet-based survey of contributors to the linux kernel', *Research Policy*. 32(7), 1159–1177.

Huizingh, E. K. R. E. (2010). 'Open innovation: State of the art and future perspectives', *Technovation*. 31, 2-9.

Jarzabkowski, P. (2005). *Strategy as practice: An activity-based approach*. SAGE.

Jelinek, M. (2004). 'Managing Design, Designing Management', in Boland, R., Callopy, F. (Eds.), *Managing as Designing*. Stanford.

Jorde, T. M., Teece, D. J. (1989). 'Competition and cooperation: Striking the right balance', *California Management Review*. 31, 25–37.

Kogut, B. (2000). 'The network as knowledge: Generative rules and the emergence of structure', *Strategic Management Journal*. 21, 405–425.

Kogut, B., Metiu, A. (2001). 'Open-source software development and distributed innovation', *Oxford Review of Economic Policy*. 17(2), 248–264.

Krippendorff, K. (1989). 'On the essential contexts of artifacts or on the proposition that "Design is making sense (of things)"'. *Design Issues* 5(2), 9–39.

Krishnamurthy, S. (2005). 'An analysis of open source business models', in Feller, J., Fitzgerald, B, Hissam, S.S., Lakhani, K.A. (Eds.), *Perspectives on Free and Open Source Software*. MIT Press.

Lakhani, K. R., von Hippel, E. (2003). 'How open source software works: "free" user-to-user assistance', *Research Policy*. 32(6), 923–943.

Lakhani, K. R., Wolf, R. G. (2005). 'Why hackers do what they do? Understanding motivation and effort in Free/Open Source Software Projects', in Feller, J., Fitzgerald, B, Hissam, S.S., Lakhani, K.A. (Eds.), *Perspectives on Free and Open Source Software*. MIT Press.

Lane, P. J., Lubatkin, M. (1998). 'Relative absorptive capacity and interorganizational learning', *Strategic Management Journal*. 19, 461–477.

Langlois, R. N. (1990). 'Creating external capabilities: Innovation and vertical disintegration in the microcomputer industry', *Business and Economic History*. 19, 93–102.

Leonard-Barton, D. (1992). 'Core capabilities and core rigidities: A paradox in managing new product development', *Strategic Management Journal*. 13, 111–125.

van der Linden, F., Lundell, B., Marttiin, P. (2009). 'Commodification of industrial software: A case for open source', *Software, IEEE*. 26, 77-83.

- McKusick, M. K. (1999). 'Twenty years of Berkeley Unix: From AT&T-owned to Freely Redistributable', in DiBona, C., Ockman, S., Stone, M. (Eds.) *Open Sources: Voices from the Open Source Revolution*. O'Reilly.
- Nikula, U., Jantunen, S. (2005). 'Quantifying the interest in open source systems: Case south-east Finland', in Scotto, M., Succi, G. (Eds.), *Proceedings of the First International Conference on Open Source Systems*, pp. 192–195.
- Nooteboom, B. (2008). 'Learning and innovation in inter-organizational relationships', in Cropper, S., Ebers, M., Huxham, C, Ring, P.S. (Eds.), *The Oxford Handbook of Inter-organizational Relations*, pp 607-634. Oxford University Press.
- Oba, B., Semercioz, F. (2005). 'Antecedents of trust in industrial districts: an empirical analysis of inter-firm relations in a turkish industrial district', *Entrepreneurship and Regional Development*. 17(3), 163–182.
- Oliver, A. L. (1994). 'In between markets and hierarchies - networking through the life cycle of new biotechnology firms', Institute for Social Science Research, Working Paper Series issr-1005, Institute for Social Science Research, UCLA. available at <http://ideas.repec.org/p/cdl/issres/issr-1005.html>.
- Pisano, G. (2006). 'Profiting from innovation and the intellectual property revolution', *Research Policy*. 35(8), 1122–1130.
- Powell, W. W., Koput, K. W., Smith-Doerr, L. (1996). 'Interorganizational collaboration and the locus of innovation: networks of learning in biotechnology', *Administrative Science Quarterly*. 41(1), 116–145.
- Pykäläinen, T. (2007). 'Model for profiting from software innovations in the new era in computing', *Technovation*, 27, pp179-193
- Raymond, E. S. (1999). 'A Brief History of Hackerdom', in DiBona, C., Ockman, S., Stone, M. (Eds.), *Open Sources: Voices from the Open Source Revolution*, pp 19-30. O'Reilly.
- Ring, P. S., van de Ven, A. H. (1994). 'Developmental processes of cooperative interorganizational relationships', *The Academy of Management Review*. 19(1), 90–118.
- Ritala, P. (2001). 'Coopetition Strategy - When is it Successful? Empirical Evidence on Innovation and market performance', *British Journal of management*, early view 2011
- Samuelson, P. (2006). 'IBM's pragmatic embrace of open source'. *Commun. ACM*. 49(10), 21–25.
- Teece, D. (1986). 'Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy', *Research Policy*. 15(6), 285–305.
- Teece, D. J. (1989). 'Inter-organizational requirements of the innovation process'. *Managerial and Decision Economics*. 10, 35–42.
- van der Linden, Lundell, F., B., Marttiin, P. (2009). 'Commodification of industrial software: A case for open source', *Software, IEEE*. 26(4), 77 –83.
- von Hippel, E. (2001). 'Learning from open source software', *MIT Sloan Management Review*. Summer, 82–86.
- von Hippel, E. (2005). "Open source software projects as 'user innovation networks'", in Feller, J., Fitzgerald, B, Hissam, S.S., Lakhani, K.A. (Eds.), *Perspectives on Free and Open Source Software*. MIT Press.
- von Hippel, E. (2006). *Democratizing Innovation*. MIT Press.
- von Hippel, E., von Krogh, G. (2003). 'Open source software and the "private-collective"

innovation model: issues for organization science', *Organization Science*. 14, 209-223.

Watson, R. T., Boudreau, M.C., York, P. T., Greiner, M.E., Wynn, Jr., D. (2008). 'The business of open source', *Commun. ACM*. 51, 41–46.

West, J. (2003). How open is open enough? melding proprietary and open source platform strategies. *Research Policy*. 32, 1259–1285.

West, J., Gallagher, S. (2006). 'Patterns of Open Innovation' in Open Source Software, in Chesbrough, H., Vanhaverbeke, W. (Eds.), *Open Innovation: Reaching a new paradigm*, pp 82-108. Oxford University Press.

Whittington, R. (2007) Strategy Practice and Strategy Process: Family Differences and the Sociological Eye. *Organization Studies*, 28, p1575-1586

Whittington, R., Caillaet, L. and Yakis-Douglas, B. (2011). 'Opening Strategy: Evolution of a Precarious Profession', *British Journal of Management*, 22: 531–544.

Wijen, F., Ansari, S., 2007. Overcoming inaction through collective institutional entrepreneurship: Insights from regime theory. *Organization Studies*. 28, 1079-1100.

Zhao, L., Elbaum, S. (2003). 'Quality assurance under the open source development model', *Journal of Systems and Software*. 66(1), 65 – 75.

Figure 1: A framework for analyzing community transformation

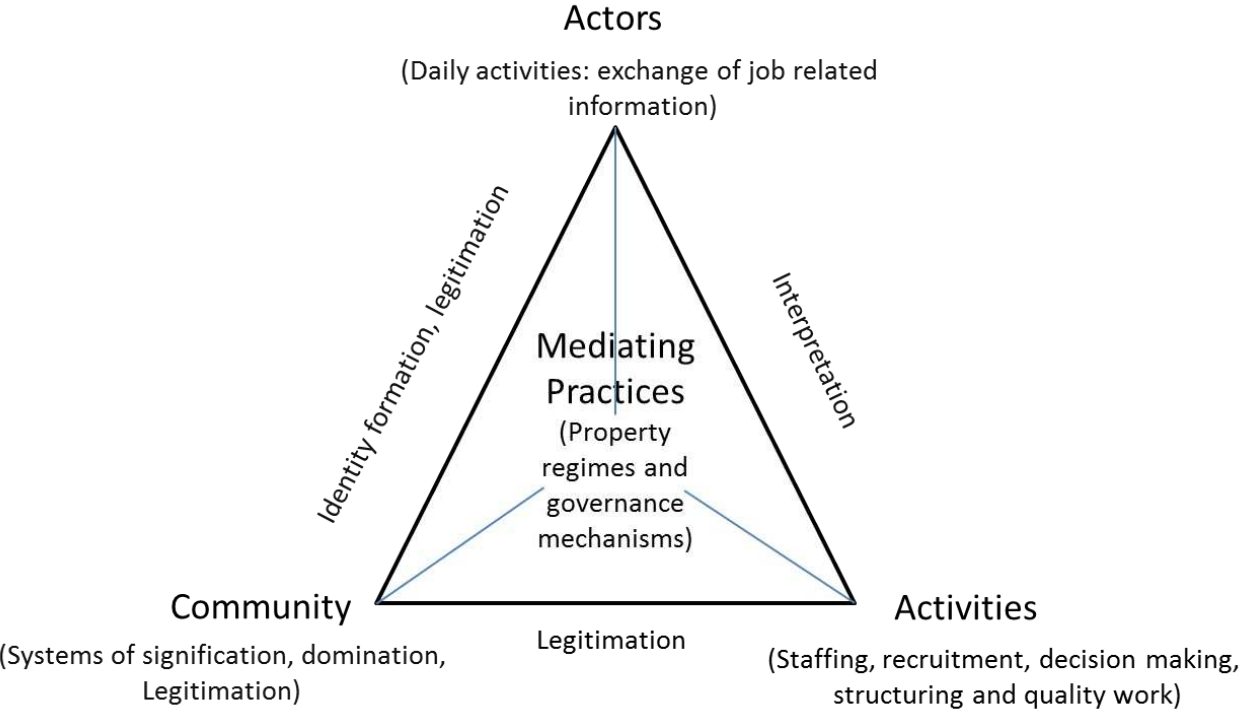


Table 1: Cases of success examined in this study

Organization	Overview
Apache (community-led)	A web server software which has a liberal license allowing use in commercial products. Apache is a community project since its start in 1999 but enjoyed a lot of corporate support early on from firms like IBM who use the technology in some of their products. Apache is governed by the Apache Group which later became a foundation. New activities are put through an incubation process for taming “providing guidance and support to help each new product engender their own collaborative community, educating new developers in the philosophy and guidelines for collaborative development as defined by the members of the Foundation” ⁱ
Linux (community-led)	An operating system software whose development is led by a vast global community, with a public license (General Public License, GPL) limiting commercialization. But it is used as infrastructure element by many firms successfully. Some key developers were later hired by leading companies in the industry. It lacks any formal governance body, and led by its originating leader Linus Torvalds since 1991, and has a closed and small leadership team. Yet the accelerating corporate contributions have prompted creation of a foundation in 2007 dedicated to fostering the technology, and hosts several events where influential community members are brought together.
Eclipse (corporate-led)	A software development platform initiated by IBM in 2001. The alliance has later become a foundation which oversees the development ecosystem in a more transparent way. Foundation members are leading firms of the computing industry who pay dues and dedicate developers to the project. Contributing individuals are promoted to the committer status by existing committers, provided that they demonstrate ‘discipline and good judgment’ ⁱⁱ . New projects are put through an incubation process, similar to Apache case. The liberal license of Eclipse software was later refined by the foundation and made compatible with GPL to facilitate reuse in appropriation, along with changes in code acceptance policies.
Mozilla (corporate-led)	A project with web browser and e-mail software. It was initiated by the firm Netscape upon its failure to compete with Microsoft’s Internet Explorer. Its license has gone through several changes from one that explicitly favored Netscape, to a more standard liberal license compatible with the broader open source ecosystem (de Laat, 2005). Mozilla considers itself as a “hybrid organization, combining non-profit and market strategies to ensure the Internet remains a shared public resource” ⁱⁱⁱ . Mozilla uses an incubation process similar to Apache for taming new projects.
GCC (community-led)	The GNU Compiler Collection is a piece of software which is used for compiling many other software for a variety of platforms, and is licensed with GPL. Its development, started in 1987, was confined within the GNU team led by Richard Stallman, a prominent figure of the open source movement. As it became a fundamental technology for many firms targeting the UNIX platform, a steering committee was formed in 1999 with representatives from leading firms and universities, but the project leadership emphasizes that the committee members represent communities, not their employers.
Android (corporate-led)	An open software platform for mobile devices had an impact in the market in only three years following its announcement in November 2007. Various device producers switched to Android, further strengthening its acceptance and contributing features. The innovation ecosystem is led by Google, and formalized as the Open Handset Alliance with 81 member organizations from various segments of the mobile industry. The software is licensed using the liberal OSS license used in the Apache project. Although promotion policies are transparent, these are less meritocratic compared to more mature projects like Eclipse; for example it is only the project leaders (who happen to be Google employees) who promote others, etc. Therefore corporate influence is also through this hierarchy. It is unclear how new projects are approved.

Table 2: Cases of failure examined in this study

Organization	Overview
Sun	<p>The company's technology, Java, has been distributed freely to promote adoption. But although Sun used its so called 'community process' to incorporate industry demanded features into their Java technology, the company essentially maintained a hard handed approach for controlling it. While companies like IBM enjoyed considerable increase in the revenues from Java technology, Sun, despite its ownership of the Java brand, trademarks, and patents, has failed to turn this advantage to revenues. Although the company finally decided to adopt open source strategy in the face of competition, this move came rather late and the company was finally sold to Oracle following its decline over the years.</p>
Microsoft	<p>The company has made several moves towards embracing open source strategy from 2004 on. However, these moves appear mainly a boundary spanning activity-mainly public relations- rather than a business strategy; as noted by the company officials these moves did not involve any core product^{iv}. Google trends show that the excitement around Microsoft's open source move is fading out^v. Despite its firm position in the PC software market, the company's position seems to be under threat. In May 2010 stock market value of Apple has surpassed Microsoft for the first time^{vi}. Although Apple does not have a big share in the PC market it has been successful in offering innovative products in the mobile segment, and adopted open innovation strategy for many years at the very core of its technologies (West, 2003). Similarly the open source Android technology by Google has been very successful in the mobile market, further limiting Microsoft's future prospects for expanding beyond the shrinking PC market. The shifts in relative positions of these players suggest the advantage of open source strategy in the new market landscape.</p>
Nokia	<p>Having maintained a strong market leadership for so long, the company's market share and profits are on the fall with entry of Apple and Android into the mobile market. Nokia has a history of collaborating with industry partners, such as for developing the Symbian operating system used in smartphones. However, the current situation shows that the Android alliance is quickly sweeping out the Symbian from mobile devices. Nokia's recent reaction to these developments was to change the troubled mobile OS, by announcing partnership with Microsoft^{vii} (instead of, for example, being a late participant in Android alliance).</p>

Table 3: Mechanisms and practices in corporate led and community led OSS projects

	Community led	Corporate led
<i>Licensing</i>	GPL, targets largest adoption	liberal, balances adoption and appropriation
<i>Governance mechanisms</i>	not formal, meritocratic	foundation/bureaucratic, transparent
<i>Acceptance of new projects (goal setting)</i>	not transparent, left to leadership	well defined incubation processes
<i>Human resources practices</i>	meritocratic promotion of developers	meritocratic promotion of developers
<i>Corporate leadership</i>	Imposed by hiring lead developers, or through committees	Imposed through formal bureaucracy
<i>Quality assurance</i>	community powered	community powered, with additional measures for legal issues

iApache incubator policy at http://incubator.apache.org/incubation/Incubation_Policy.html,
retrieved Nov.2011

iiEclipse committer policy at <http://www.eclipse.org>, retrieved Nov. 2011

iiiMozilla organization description at <http://www.mozilla.org/about/organizations.html>, retrieved
Nov. 2011

ivTech business news site at http://www.seattlepi.com/business/179256_msftopen25.html,
retrieved Nov. 2011

vGoogle trends <http://www.google.com/trends?q=microsoft+open+source>, retrieved Nov. 2011

viTech business news site at <http://techcrunch.com/2010/05/26/apple-microsoft-market-cap-2/>,
retrieved Nov. 2011

viiBBC business news <http://www.bbc.co.uk/news/business-13154794>, retrieved Nov. 2011