The Institutions of Open Source Software: Examining the Debian Community[1]

Juan Mateos-Garcia*

CENTRIM, Brighton University
Freeman Centre
University of Sussex
Falmer Brighton East Sussex
United Kingdom BN1 9QE
E: mail: J.Mateos-Garcia@brighton.ac.uk

W. Edward Steinmueller

SPRU – Science and Technology Policy Research
Freeman Centre
University of Sussex
Falmer Brighton East Sussex
United Kingdom BN1 9QE
E: mail: w.e.steinmueller@sussex.ac.uk

Abstract
Free and open source software activities involve and, perhaps, evolve institutions (rules, norms and standards) that influence the formation, growth, and demise of communities. Community institutions are attractors for some individuals while discouraging other individuals from entering or continuing to participate. Their suitability may change as a community grows. This paper examines the institutions of the Debian community where issues of community identity, distribution of authority, and decentralisation have facilitated growth and development. These same institutions have also resulted in conflicts regarding community purposes and the quality and delivery of the community's output. We examine the institutional redesign undertaken to address these problems and derive implications for F/LOS communities and companies.

* Corresponding Author

Introduction

This paper employs the concepts of epistemic community, situated learning and legitimate peripheral participation to address the way in which communities of F/LOS (Free/Libre Open Source Software) developers define their purposes, co-ordinate their activities and recruit new members, as well as the processes by which these novices are socialised.[2] This framework is drawn from the literature on political science (Haas, 1992), scientific communities (Knorr Cetina, 1999) and communities of practice (Lave and Wenger, 1991), (Brown and Duguid, 1991). It was first employed to the study of F/LOS by Edwards (2001) as a means of focussing on F/LOS communities' social features and work practices. This focus is neglected in a large part of the existing literature, which primarily concentrates on individual motives for participation in F/LOS projects and the advantages of the F/LOS methodology when compared with traditional (closed) paradigms for software development.[3] These existing approaches are most often set in a context in which F/LOS projects are relatively successful and untroubled. Edwards' framework contributes to the opening of the "black box" of the F/LOS project, and we believe can be applied to contexts where we observe the presence of dysfunctional processes with a negative impact on performance and participation expectations, something that is quite common and also contrary to the sample of cases analysed by most researchers studying F/LOS communities.

In extending and applying this framework to the specific case of the Debian Project, we aim to illustrate its relevance for analysing the social structures of F/LOS communities and for diagnosing some of their possible ills and to draw some conclusions regarding what institutions (rules, norms, and standards) might best serve such communities.

---

[2] Throughout this paper we use the term "Free/Libre Open Source" software (F/LOS), to refer to our subject of analysis in order to acknowledge both the more politicised ("Free/Libre") faction of developers organised around the Free Software Foundation (http://www.fsf.org/), and the more pragmatic "Open" community which emerged with the launch of the Open Source Initiative (http://www.opensource.org/).

[3] For work focusing on the motivations of F/LOS developers see, for example, Raymond (1999), Lerner and Tirole (2000), Ghosh (1998), von Hippel (2002), Bezroukov (1999) or Himanen et al (2001). For some analyses of the comparative efficiencies of F/LOS and other paradigms for software development see Raymond (1999), von Hippel (2002) and Varian and Shapiro (2004).

The advantages of F/LOS projects are often depicted as a departure from a central tenet of existing theories of software development – 'Brooks law.' Brooks (1995) asserted that software development is subject to diminishing returns with the addition of greater number of developers. By contrast, F/LOS development is usually characterised by its advocates (and many scholars) as an activity that is subject to proportional or more than proportional increases in outputs (quantity and quality of the software code produced) as more developers become involved in a project. This deviation from conventional accounts of software development is explained by what Eric Raymond calls "Linus's Law," according to which "*Given enough eyeballs, all bugs are shallow*"[4]. This 'law' is focussed on one feature of the F/LOS development process – the problem of producing reliable code to implement clearly specified requirements, and on one consistent feature of software development – the un-intentioned introduction of error in code production. This is an important, but by no means, sole determinant of the success of the organisational innovation that F/LOS represents.

For example it does how contentions that may be offered by the people behind the eyeballs are validated or how F/LOS communities might seek to recruit individuals with the most acute and relevant vision to the tasks at hand. In this paper we examine how these issues of validation and recruitment play out in one setting, the Debian F/LOS community, which has faced and continues to face challenges that may affect its prospects for continued success.

We begin the paper by presenting a theoretical framework which, along the lines followed by Edwards, begins with the concepts of Epistemic Community, Situated Learning and Legitimate Peripheral Participation, and uses these concepts to pursue a more detailed analysis of the role and functions of a project's 'administrator' – one of several roles that emerge from the socio-technical and epistemological dynamics of the Debian project, and more broadly, F/LOSS communities. These dynamics also produce institutions (rules, norms and standards) that regulate participation inside a project as well as managing conflict and dissension which might diminish the quality of a project's outputs (and social processes). This discussion leads to the identification

---

[4] In an extended form, "*Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone*" (Raymond 1999), p. 41.

of some mileposts in the evolution of a project, each of which suggests the need for institutional strategies in order to cope with the growth of the technical and social complexity of the project. We then use this framework to consider the case of the Debian Linux Distribution, using primary data from its mailing lists archives, handbooks written to inform potential and actual community members, and previous analyses of institutional evolution and political conflict in this project.

1. F/LOS development as a communitarian process

Kasper Edwards (2001) builds a conceptual framework based upon analytical tools borrowed from the fields of political science and sociology in order to analyse the social organisation of F/LOS communities. Of particular interest in his analysis are the processes by which novices enter such communities and learn the technical skills and social norms that make their participation meaningful and valuable. In order to undertake this task he employs the concepts of "epistemic communities" and "situated learning," which we now briefly reprise.

Epistemic communities and Situated Learning

Haas (1992) introduces the concept of "epistemic communities" in the field of political science to identify groups of individuals who share a number of normative and causal beliefs and knowledge validity notions. Individuals in epistemic communities are in implicit or explicit agreement about the objectives of the community. In the case of Haas' domain of application, international policy, this is achieved through the accumulation of a body of knowledge, which is expected to influence policy makers' decisions in a desired direction. A central feature of Haas' approach is that a shared epistemological framework regulates the knowledge accumulation process by establishing which methodologies and criteria are taken as most useful for the production and validation of knowledge.

This framework seems particularly well suited for the analysis of the organisational arrangements adopted by participants in F/LOS projects because of its focus on collective work practices and their linkages to a specific epistemological framework, and the characterisation of a common "policy" enterprise as one of the essential

features of the community. This instrumental approach for the definition of social interactions makes it possible to encompass the "socially-light" and "intimacy-averse" environment in which a considerable share of F/LOS development takes place.[5] Edwards locates the emergence of socialisation and identity in relation to the *process* of open source working, and introduces Lave and Wenger's (1991) framework as the analytical method for comprehending this process.

The practices, values and beliefs that characterise F/LOS development and the interactions between those people who take part in it[6] create an environment that reduces the costs of re-negotiating purposes and facilitates decentralised collaboration between volunteers from heterogeneous backgrounds. The epistemic community approach provides little guidance, however, in answering the question: *how do outsiders start participation in the community, and through which processes do they learn and internalise the community's set of shared normative and causal beliefs, notions of validity and goals that substantiate their membership?*

To address this issue, Edwards utilises Lave and Wenger's (1991) concepts of "situated learning" and "legitimate peripheral participation." This leads him to identify a number of activities and structures through which novices start their participation in the social and technical spaces of a project and learn the norms and policies that regulate behaviour inside them[7]. Of particular note, Edwards controversially construes the "mentor" (a pivotal figure in Lave and Wenger's case studies) – the novice is 'guided' by what is embedded in the code, mailing lists and forum postings that the novice reads, adapts and contributes rather than an individual.[8]

---

[5] Respondents to FLOSS Survey (Ghosh et al., 2002) and the Boston Consulting Group Hacker Survey (Lakhani et al., 2002) emphasise acquisition of new skills and engagement in intellectually stimulating activities as the primary aspects for participation in F/LOS development, while social aspects remain secondary.

[6] Examples of which would be aversion to proprietary methods for software development, an emphasis on technical arguments during discussions and a disregard for "formal" qualifications and organisational, social or economic status. See Levy (1985) and Himanen (2001) for overviews of the moral precepts that constitute the basis of the F/LOS movement, also referred to as "the Hacker Ethic".

[7] See Von Krogh et al (2003) for an empirical analysis of some of the joining and specialisation patterns through which novices start participation in the Freenet project.

[8] Taking this view involves the supposition that the individual identities of insiders have effectively "coalesced" into these artefacts as part of a collective, decentralised development process, an assumption that many who are sympathetic to Lave and Wenger's framework will find problematic since it strips out and objectifies the interactive social processes involving the novice and the mentor into a relation between the novice and the 'text.' As we will see below, however, interactivity is re-introduced as the novice attempts to make contributions to the community effort.

Through this social process the novice may come to internalise the values of the community and learn how contributions are solicited and validated as well as the norms that regulate social interactions inside the community. Eventually, the novice becomes recognised by others as a member of the community and may influence the community through contributions to software code, documentation, technical discussions, Frequently Asked Question (FAQ) lists, etc.

2. Distributed authority inside an epistemic community and knowledge differentials as its organising principle

Edwards' framework is a useful starting point for the analysis of the organisational arrangements through which F/LOS communities carry out their activities, as it identifies mechanisms in projects that address some of the pitfalls of decentralised software development and, more generally, online interaction[9]. . Elliot and Scacchi's analysis of the GNUe documentation project concurs with this view, describing the way in which an alignment of participants' belief systems mitigates conflict.[10] We contend that these mechanisms are neither precise nor strong enough to effectively specify the goals and organise the efforts of F/LOS developers, nor necessarily sufficient to resolve conflicts that emerge on the day-to-day processes of F/LOS development. In these areas, the exercise of authority becomes a necessary source of stability and direction, which prompts the following discussion of authority in such communities.

---

[9] See Herbsleb et al (2000) for an appraisal of the problems faced by (commercial) software projects undertaken by dislocated teams.
[10] Elliot and Scacchi (2003).

The exercise of authority in F/LOS development work[11]

Academic accounts of F/LOS development often present a 'flat' hierarchical structure in which volunteers take the initiative of contributing software modules (fragments of a program) that are integrated with each other through standardised interfaces[12]. Our contention is that the process of integration is more complicated than strategies for integration of contributions play an essential role in determining the outcomes of a project. The organisational structure of most F/LOS projects has a vertical component which regulates its development and operation.

One figure common in F/LOS projects who embodies these integration functions is the administrator (elsewhere referred to as 'leader' or 'maintainer'). The administrator of a project is, in Eric Raymond's terms, its 'owner', that is, the individual who starts it, inherits it from a previous owner or recovers it from abandonment, and has the right to distribute its official version.[13] Raymond does not focus on the function preceding the exercise of this "right to release", this is, the right to decide *which contributions go inside the official version and which do not,* or the "integration function. Technocratic accounts of F/LOSS development would perhaps present this as a trivial undertaking that does not deserve special attention: the project administrator simply assembles the "puzzle" of the software program with the "pieces" contributed following the rules and principles that characterise the F/LOS epistemic community. If a software project is large and technically complex, this might be a difficult task, and because of the voluntary nature of F/LOS development work, some of the "spaces in the puzzle" (i.e. tedious tasks such as debugging and documentation) may not be completed in a timely way.[14] Nevertheless, by a technocratic account, there exists a "right" solution for the software problem, which the administrator will adopt, with guidance from the community when necessary. If she does not, she will eventually face a rebellion from other developers, who have the

---

[11] We do not enter on issues related to the definition of for example, authority versus influence and power (Bachrach and Baratz, 1963): in our framework, we use this concept to refer to attempts at influencing the actions of other participants in the project.
[12] This is, a Raymondian Bazaar (see Raymond, 1999).
[13] Raymond (1999).
[14] Ibid.

capacity to leave the project and to start a different branch (to "fork" the project) adopting the "right solution" that the administrator declined to adopt.[15]

Accounts such as this seem naïve to social process and incorporate a technologically deterministic view of software development which appears prevalent in ICT-savvy circles (and sometimes, amongst social analysts of ICT). Alternatively, we contend that the "software puzzle" we have described presents, except in its simplest configurations, a diversity of possible "solutions," each with merits and shortcomings. In a context of uncertainty and diversity (i.e. heterogeneous, sometimes conflicting individual goals and perspectives), there is space for choice about how to solve the puzzle, which arrangements of "software pieces" are more desirable, and also for differences between individuals about which solutions should be employed.[16] Our understanding of development processes in F/LOS projects adds a cautionary corollary to Linus' Laws; "*Given enough opinions, all development paths are contested*"

Decisions about which contributions and strategies are more useful for the solution of a problem, which issues should be considered problems and which should not, and the criteria according to which solutions to problems are to be judged are, in a context of uncertainty, made on the basis of values, beliefs and opinions. Since the rules, precepts and "shared goals" of a F/LOS epistemic community are not defined in a sufficiently rigorous or complete way so as to dispel the need to make frequent decisions with a degree of arbitrariness, exercises of authority cannot be completely based on rational and technical arguments. If these rules defined perfect protocols (in economic terms, complete contracts) for conflict resolution with no individual making an 'arbitrary decision', it would still be true that the selection of the goals and values embedded in them constitute, again, exercises of authority.

---

[15] This sort of explanation is implicit in Raymond (1999) who assumes that a meritocratic process of leader selection and the existence of 'checks and balances' provided by capacity to 'fork' (take existing code and develop it in a different way) guarantee the selection of optimal development paths. Similar explanations of F/LOS efficiency based on the allocation of stocks of information about user needs are present in von Hippel (2002).

[16] Discussions in developer mailing lists (i.e. http://www.lists.debian.org) and forums (such as Slashdot or Kuro5hin) demonstrate the extent of disagreements about the relative merits of different programs and techniques, and the desirability of achieving different goals permeate day-to-day development activities in the F/LOS community.

The source of authority in F/LOS projects: knowledge differentials between individuals

The previous discussion shows the importance of situating authority at the foreground of our understanding of F/LOS communities and their activities: determining its sources constitutes a promising approach to the understanding of the organisation of F/LOS projects.

We contend that individuals obtain the right to exercise authority over a project's development process by demonstrating superior knowledge about it, and that this knowledge is acquired through participation. By focussing on the social dynamics through which this knowledge is distributed throughout a project, it becomes possible to trace the emergence of the structure, usually hierarchical, by which development efforts inside are managed. Our analysis complements and extends previous research on the issue, much of which has concentrated on the motives for individuals' contributions and have highlighted reputation building.[17]

Inside our framework, "contributions" are not, *per se*, direct sources of reputation and authority in a project, but demonstrations of knowledge about its goals and its values, and of the skills necessary to fulfil a useful role inside its community. In a social milieu where sustaining a common purpose may be difficult and in which technical proficiency is given the utmost value, these demonstrations are rewarded with reputation, whose 'exchange value' is the right to exercise authority over the project and, if not its participants, then at least their contributions.[18] This line of reasoning is more fruitful for analysing the social dynamics inside F/LOS communities than the "Gift Culture" metaphor adduced by Raymond. We consider authority as the embodiment of a community's respect and trust for the dedication and technical proficiency of those individuals who, through their sustained involvement in the project, have demonstrated that they *know better*, not as some sort of deference to the reputation and property rights of more active or experienced developers.

---

[17] Raymond (1999), Lerner and Tirole (2000).
[18] It is important to emphasise that the knowledge we are referring to is both technical (that is, about software development and design), social (about the processes through which the community carries out its activities and the social norms that regulate interactions between participants) and "political" (about the goals of the community and preferred ways of achieving them)

If, as we argue, the essential source of authority in a project is knowledge of purpose and technique (acquired and demonstrated through participation), then the community will also be, to a degree, resistant to opportunistic behaviours aimed at maximising individual reputations or manipulating the direction of its project.

Distributed Authority

Our analysis has so far amounted to an identification of "institutional frameworks" (norms of an epistemic community), "membership acquisition processes" (through legitimate peripheral participation) and "governance" (the accumulation of decision-making authority). These three elements constitute the structure through which the efforts of a decentralised and heterogeneous community of volunteers are coordinated and focussed inside F/LOS projects. In the simplest instantiation of this model, all three of these elements are controlled directly or indirectly by the project administrator, whose authority arises from her (perceived) superior knowledge of the project. The authority to delegate, or to distribute authority, however, is an essential aspect of the project administrator's authority, and the source of the emergence of organisational hierarchies such as the star-shaped structure characterising Linux, or the team-based model followed by the BSD Unix project[19].

We contend that this authority delegation process is an inevitable consequence of the growth in the size and complexity of a project, and of the diversity of the population participating in it. As new developers enter a project and specialise in different areas, they implicitly appropriate these areas through their contributions, obtaining de facto responsibility over them. If a conflict emerges between the project administrator and an expert in an area where the latter is perceived to be more knowledgeable, the absence of clear criteria to determine who has the ultimate right to decide will result in confusion.[20] An administrator's delegation of authority to developers who have a

---

[19] Moody (2002) and Weber (2004) characterise the GNU/Linux organisational model. See Bezroukov (1999) for quotes from a former BSD Unix developer defending its centralised development model, as well as Jørgensen (2001).

[20] See Mateos-Garcia and Steinmueller (2002) and Moody (2002) for an account of this kind of conflict in the context of the GNU/Linux and Emacs projects.

history of high quality contributions is one way of addressing limitations in the 'span of control' in being well informed about the evolution of "her" project across a multitude of fronts. This authority is exercised by decisions about an individual's contributions to a project (either incorporating or rejecting them) which implies the acceptance or denial of that individual's access to the community. Persistent acceptance of contributions in an area eventually leads to the developer become a 'de facto' owner of it. This process will contribute to reducing the scope and intensity of controversies, inasmuch this delegation will be based on trust and experience. It also suggests that the administrator will delegate her authority on developers who are aligned with her goals and values, and with whom she is less likely to conflict. This is clearly a process of social structuration through accretion.

As Linus Torvalds, administrator of the GNU/Linux kernel development team explains in a mailing list:

> "I take stuff that I feel is good. Often that feeling of goodness
> comes from trusting the person who sends it to me, simply by past
> performance. At other times, it is because I think the feature is
> cool, or well done, or whatever. Hint: if you want stuff in my tree,
> make me trust you. Or work on things that I feel are innately
> interesting. Don't bother dragging me into your flame-wars and
> trying to convince me that I 'must' apply your patches."[21]

The conceptual framework we have elaborated in this section has important implications: F/LOS development comprises technological, social and institutional aspects, which in our framework could be respectively referred to as "design/modularity", "learning/reputation" and "governance/membership", and their interaction will be an essential determinant of project policy and success[22]. Although we acknowledge the existence of F/LOS projects that function according to other organisational structures not necessarily based on the delegation of authority from an

---

[21] http://www.lib.uaa.alaska.edu/linux-kernel/archive/2002-Week-44/0094.html
[22] Defining project success is a difficult task (see Crowston et al, 2003), although we would lean towards a definition that encompasses not only technical performance but also community sustainability.

administrator, we hypothesise that their emergence can be traced using the conceptual framework we have presented before.

In the following section we undertake a case study of one such project, Debian, with the aim of illuminating the relationships between the latter two spheres and substantiating the claims that we have introduced in the preceding analysis. It is one of our contentions that the absence of clear mechanisms for the exercise of authority in this project, including this delegation, that have become sources of conflict inside it.

3. Social Conflict and Organisational Structure in the Debian Community

In this section we develop a case study of the Debian GNU/Linux distribution to provide an empirical content to the preceding conceptual and theoretical discussion. The Debian community was chosen rather than some other F/LOS community because 1) the process of deciding to include or exclude a package in the distribution is less complex and this unit of analysis affords greater comparability than analysis of 'submits' of code fragments in a F/LOS community based upon coding, 2) the Debian community has grown in size and complexity over time sufficiently to trigger institutional evolution, and 3) the Debian community has devoted considerable (public) effort to a reflexive discussion of its purposes and methods.

The case study was developed, and is presented, in two stages. First, we investigate the stated aims and philosophy, organisational structure and development practices of the Debian community. Having done this, we examine its institutional evolution through a literature review and an analysis of available evidence including relevant news items and, when possible, primary evidence extracted from mailing lists, online forums and other public discussions. In this second stage, we focus on periods of instability that have led to the emergence of new organisational roles and structures to co-ordinate development efforts and regulate participation inside the project. This approach is inductive in that it works back from manifestations of change to their possible sources and consequences. By focussing on episodes in which organisational change occurs, we are attempting balance expressions of dissent (which may be more or less continuous within the community) with statements defending current practice because, at the points of our analysis, the possibility of change of these practices is under active discussion. Nonetheless, this methodology is not capable of producing systematic evidence about *continuity*, the periods in which dissent exists but is not mobilised into widespread or actively debated calls for or enactment of change. Finally, in the section entitle *Analysis*, we examine this historical account using as an analytical prism the concepts and analytical tools presented in the previous section.

Introducing Debian

The Debian project is a GNU/Linux distribution (a collection of F/LOS software programs integrated with an operating system) that was launched in August 1993. [23] Debian is developed by volunteers working in a decentralised fashion, and made available for free. Unlike several other distributions, Debian has not distributed a 'value added' package with user-friendly installer software, manuals, and CD or DVD images of the software.[24]

The Debian community has avowed its commitment to the F/LOS philosophy with the institution of the Debian Social Contract and the Debian Free Software Guidelines (DFSG) which respectively codify the moral agenda of the project (to develop a F/LOS Operating System that performs up to the highest standards of quality) and define a number of F/LOS criteria with which the license of all the software included in the distribution must comply.[25]

The organisation of Debian [26]

As we have mentioned, Debian is a distribution: its developers do not "code", but "maintain packages" (initially developed by "Upstream Authors" who are not, usually, part of the community), an activity that involves reviewing those packages and integrating them into a release. These packages (4,500 source packages as of the release of Debian 3.0 and over 18,000 in Debian 4.0) contain the software components that comprise an Operating System, as well as many other tools and applications (i.e. desktop environment, web-browser, database management, word processor etc.)[27]. Each maintainer is responsible for guaranteeing that her package

---

[23] See Murdock (1993).

[24] Debian does identify companies that serve some of these functions for users and notes that these companies make a financial contributions to the Debian effort

[25] The Debian Free Software Guidelines constitute the kernel of the Open Source Definition. See http://www.debian.org/social_contract (Debian Social Contract), http://www.debian.org/social_contract#guidelines (Debian Free Software Guidelines) and http://www.opensource.org/docs/definition_plain.html (Open Source Definition)..

[26] See Michlmayr (2003) for a more detailed overview of Debian's organisation.

[27] For more data about the size of different Debian distributions and its evolution, see Robles and González-Barahona (2003).

contains high quality software that fits coherently with the rest of the distribution, and is licensed under terms complying with the Debian Free Software Guidance (DFSG).

Maintainers upload packages to the Debian Archive, where they are checked for consistency using automated scripts, and reviewed by an *FTP Master* who assesses their basic quality and copyright status.[28]. The packages that are accepted are, depending on their level of reliability, incorporated to a file repository[29]: The project's *Release Manager* periodically "freezes" a sub-set of the packages in the most reliable repository, which are then subject to further testing and eventually released as the "stable Debian distribution".

Teams such as Security or Quality Assurance take care of specific issues when the need arises, such as when vulnerabilities in the code are detected or a package is 'orphaned' by its maintainer.[30].

A second institutional aspect that characterises Debian is the *New Maintainer Process* (NMP) through which individuals outside the project become maintainers.[31] The NMP begins with the verification of an applicant's identity and an assessment of her technical expertise and understanding and commitment to Debian's norms and values[32]. The NMP usually takes several months to complete and has its own organisational apparatus.

Another essential element of Debian's organisation is an "officership" in charge of decision-making and conflict-resolution, with a structure established in the Debian Constitution (first ratified in its version 1.0 on December 1998). [33]The **Debian Project Leader** (DPL) is chosen by Debian Developers in a yearly General Election;

---

[28] The FTP Masters are also in charge of maintaining the technical infrastructure that supports the archive. For a summary of their functions refer to http://lists.debian.org/debian-project/2005/02/msg00184.html.

[29] See Monga (2004) for further details..

[30] See http://www.infodrom.org/~joey/Vortraege/2005-06-23 and http://qa.debian.org/ for further information on these special teams.

[31] For a detailed description of the New Maintainer Process see http://www.debian.org/devel/join/newmaint and Wallach et al (2005)

[32] See O'Mahony and Ferraro (2004), and Coleman (2005) for further details regarding the principles according to which Debian's Web of Trust is organised, and examples of the procedures used to test an applicant's technical expertise and ideological commitment to Debian.

[33] See the Debian Constitution Version 1.2 (http://www.debian.org/devel/constitution) for a detailed description of each of the different positions and roles in the Debian organisational scheme.

she is the official representative of Debian, responsible for defining the project's vision, and has the right to make urgent decisions, appoint a Technical Committee in charge of conflict-resolution, and lend her authority to other developers (including the aforementioned release manager).

Debian History

Debian has often been described as an initiative that was launched with the aim of creating a Linux Distribution that would be "true" to the Free Software principles. An analysis of available documents and public discussions shows, however, that there was an important element of pragmatism in the project founders' motivations. According to one of them, the first reason for starting the project was discontent with the quality of existing Linux distributions.[34]

In contrast to the focus on 'Software Freedom' of these accounts, the initial Debian Manifesto emphasises "quality" and "responsiveness to the needs of the user community," without any references to the advancement of a specific political agenda. The Free Software Foundation (FSF) is only mentioned in the context of practical issues related to CD-ROM distribution. Although the FSF supported Debian with a grant, there was an eventual split between both groups as a consequence of what was perceived to be the latter group's political interference with the activities of the project[35].

Debian's association with FSF raised the profile of the project creating an influx of politically motivated developers. This, together with the growing interest of commercial firms and potential contributors not as knowledgeable or vocal about software freedom issues, generated disagreements and confusion, and led to the

---

[34] See the Debian Manifesto, which accompanied the Distribution's launch at
http://www.debian.org/doc/manuals/project-history/ap-manifesto.en.html
[35] According to a former project leader "*We had a little contretemps* (sic) *with FSF because we didn't want to be under their direction, and Richard Stallman said some things that got people on the net angry, but it's not like it was some kind of war". http://lists.debian.org/debian-devel/1996/10/msg01149.html*

eventual definition of the project's policy on this area[36]. As a developer recalls in an interview published in O'Mahony (2002):

> "A guy who is now with [open source firm] had suggested putting a particular package in [to the Debian distribution]. And I felt that the package was not free software. And I objected to it, […] I needed more than "I don't like it" to be my objection [….] I got this idea that there should be a social contract between Debian and the free software community that says how that Linux distribution should behave. So I decided that I could justify the choice of what goes in Debian and what does not, that we could have a Debian Social Contract."

The Debian Social Contract (DSC) and the Debian Free Software Guidelines (DFSG) made explicit the project's ideological commitment to the tenets of Free Software, codifying the beliefs and values that participants in the project were expected to uphold. This was associated with a shift from the initial pragmatism and focus on quality and usability towards a stronger emphasis on ideological and political issues. A developer, dissatisfied with this trend, states quite starkly "*Free software was a tail that eventually wagged the dog*"[37].

The next step in the institutional evolution of the project was the creation, in 1998, of the Debian Constitution, a formal framework to regulate the election of the DPL: this was made necessary by a number of conflicts between the extant leader (who had inherited his position from the project founder) and several developers. There were concerns about a perceived illegitimacy in the exercise of authority by the DPL, excessively heavy-handed and intrusive according to some developers[38]. Additionally, his focus on "bringing Debian to the masses" went against the wishes of an important

---

[36] See http://lists.debian.org/debian-devel/1996/06/msg00600.html for a discussion on this matter: When the Project leader suggests "*dropping the non-free directory* anditscontents *from Debian. We don't really need anything in there, and it would be nice to get closer to the free software ideal*" some developers reply that (most Debian users- and developers) "*are probably interested more in a useful system and less in political goals like having 100% (not 99%) free software*").
[37] http://lists.debian.org/debian-devel/2000/06/msg01194.html . This constitutes another illustration between the principles of 'free software' and 'freedom to choose whatever software does the job' present in many F/LOS communities (Elliot and Scacchi 92003)).
[38] See http://lists.debian.org/debian-devel/1997/09/msg00966.html for an example of a situation where the project leader was perceived to be interfering with the development of the package "Deity".

segment of the community (who preferred to focus on the technical needs of the "hacker" audience)[39].

> *"While now a large number of people think that* (second project leader) *was the best thing since sliced bread, lots of people were pissed off back then with him commanding people around. And what was the result? A Constitution that would ensure that no leader would ever have such power again"*[40]

Debian had reached the organisational milestone at which the size and degree of complexity where decision-making through protracted debate was unfeasible. In some cases it would be necessary for the leader to exercise her authority in order to settle controversial matters. In order to do this she would need political legitimacy, obtained through her success in the Debian General Elections. Concerns about a potential slide of Debian into "Authoritarianism" led to the inclusion in the Debian Constitution of safeguards to avoid an excessive concentration of power in the hands of the DPL and other officers.[41]

The launch, in 1998, of the Open Source Initiative and the creation of a legal "Open Source Definition" inspired by the DFSG raised Debian's profile even further, and led to a rapid surge in the number of new contributors, which resulted in what O'Mahony (2003) and Coleman (2005) refer to as the "membership crisis," a second milestone in Debian's organisational evolution. Veteran members of the project put in doubt the technical expertise and commitment of these "novices": It was argued that they uploaded low quality packages to the distribution and were responsible for a growth in the project's bug-count (O'Mahony (2004) p. 21) and that their ideological commitment to the project was doubtful. Concerns were also voiced about the risk of malicious new maintainers obtaining access to the project's code repositories and uploading security-compromised packages to the distribution.

---

[39] See his resignation notice at http://old.lwn.net/lwn/1998/0319/resign.html
[40] http://www.cyrius.com/journal/2006/03/09 (2003-2004 DPL's blog).
[41] Including the possibility of the community overriding the DPL in a 'general resolution', or restrictions of her rights including the expulsion of other developers.

Accordingly, previous channels for entrance in the project were closed and eventually replaced with the New Maintainer Process (NMP), a formal procedure for gaining membership in Debian including a verification of the new maintainer's identity and an examination of her commitment to Debian's philosophy, as well as her technical expertise[42].

Although the NMP constitutes the last step in Debian's institutional evolution as of the Spring of 2007, there have been further controversies between developers that have, for example, led some to propose further regulations in the project, such as a binding Code of Conduct for the mailing lists[43]. More interesting, for the purposes of this paper, is the polemic concerning the (arguably) excessive power wielded by a cadre of individuals in special positions such as Debian Account Manager, FTP Master and other special teams. This group (referred to as "the Cabal" by disenfranchised developers) has been accused of lack of responsiveness and accountability to the rest of the community and of trying to manipulate Debian's evolution secretly, making decisions about the project after private meetings outside of the community's scrutiny. The polemic proposal for Debian's future evolution elaborated by the release team (comprising release assistants and FTP Masters) during a private meeting held Vancouver in March 2005 constitutes an example of this. This proposal, which implied the discontinuation of the project's support for several hardware architectures, was interpreted as an attempt to hijack the control of Debian from the community, and led to intense arguments in the developer mailing lists.[44].

Controversies about the balance of power inside Debian came to the forefront again in March 2006, when the then Stable Release Manager resigned from his position with the following comment:

> "I'm sick of being left in the void. I'm sick of ftp masters not answering
> mails from the stable release manager to negotiate a timeline. I'm sick of
> ftp masters suddenly creating arbitrary preconditions for stable updates.

---

[42] See Wallach, Allan and Harries (2005) for a history and detailed description of the process.
[43] See http://www.gatago.com/linux/debian/vote/8051951.html for a debate about this issue in the context of 2006 Debian General Elections.
[44] See http://lists.debian.org/debian-devel/2005/12/msg00574.html for criticisms in respect to the first area. See Coleman (2005), as well as the original discussion at http://lists.debian.org/debian-devel/2005/03/thrd2.html regarding the Vancouver memo "flamefest".

I'm sick of having to ask again and again and being constantly blocked by them. I'm sick of this entire situation. It makes me ill, angry and utterly frustrated. It causes me being frustrated of Debian and unable to work on other issues, needing a rest more often than planned. I should do better with my limited life"*45*

This resignation led to requests for the dismissal of a particularly controversial FTP Master from his position. It was argued that according to the Debian Constitution, this individual could be relieved from his duties by the DPL. These calls have so far been ignored or rebutted. It has been argued that individuals in positions such as FTP Master "*might be better thought of as "infrastructure maintainers" instead* (of as project delegates)*, which implies a different relationship to the DPL.*"[46]. Some developers have showed their concern about a difficult transition for the project in case an aggressive course of action is taken. [47]

Analysis

We understand the history of Debian as a process in which project growth (both social and technical) has given rise to disruptive processes that have been addressed through the implementation of institutional strategies aimed at constraining diversity (by establishing the principles that regulate the functioning of the project, codifying its goals and values and promoting conformance by new participants) and reduce the co-ordination costs incurred as a consequence of increased complexity (by enforcing a degree of standardisation in development procedures and techniques and establishing a number of quality assurance measures). The values with which it is necessary to agree in order to become a member of the Debian 'epistemic community' have been clearly established in the Debian Social Contract, and the processes of legitimate peripheral participation through which this membership is acquired has been formalised into the New Maintainer Process. The Debian Constitution constitutes a third element legitimising the exercise of authority (the third 'leg' of our conceptual triad).

---

[45] http://lwn.net/Articles/174930/ . See
[46] Interview with 2005-2006 Debian Project Leader available at http://linux.slashdot.org/article.pl?sid=06/04/11/1931202
[47] http://lists.debian.org/debian-devel/2005/12/msg00633.html

This process is rife with controversy, as the creation and rearrangement of institutional structures has been perceived to clash with essential tenets of F/LOS development such as openness or freedom, and provoked shifts in the power (and knowledge) structure of the project. It is important to highlight that the establishment of goals, norms, participation criteria and standard work practices constitutes an exercise of authority by those individuals in positions of leadership inside the project, and has been opposed by participants who support different visions of its future. Their implementation has contributed to reduce instability, but has also led to the emergence of new problems such as an alleged slowing of the admissions process, a deficit in innovation and a sluggish cycle of releases. We now examine these issues.

A selection mechanism for constraining diversity is necessary in order to minimise the scope of conflict between participants with different beliefs about desirable development techniques, architectures and intermediate and final goals for the project. As we observed earlier, devising such a selection mechanism is not straightforward and it is nearly impossible if there is no clarity about *what the participants in the project are expected to believe or strive for.* In such cases, conflict will inevitably ensue with the potentially catastrophic selection mechanism of exit or forking taking hold as different constituencies inside the community pull in different directions. The potential for this becomes higher as the project grows. Debian's increasing popularity brought an influx of developers with different "visions" about the purposes of the project, and this made it necessary to set down policies and goals (that until then had remained tacit and vague) in order to avoid continuous discussion and controversy which would stall or disrupt the project.

The authority of the project leaders determined the result of this process: developers who were strongly committed to the ideological tenets of Free Software had reached, through continuous, high quality contributions, influential positions inside the project, from which they set their political and technical agenda. This agenda was implemented in an explicit (and permanent) manner once their divergences with the pragmatist faction (less strict about, for example, the inclusion of non-free software in the distribution) became evident. The DSC and the DSFG were thus erected as

explicit barriers to the entrance to the project of potential dissenters (sources of conflict), and as guides for current members. As O'Mahony (2006) argues:

> "The process of creating these two documents formalized the previously unarticulated values upon which the group was founded. It also helped ensure that a growing body of contributors new to the group's history and norms could share them. Sources both internal and external to the project suggest that gaining internal consensus of the group's mission and articulating a basis of authority was critical in preserving the project from commercial interests". (p. 261).

The Debian Constitution can be seen as a codification of the project's vision that has the goal of communicating to potential entrants what will be expected of them if they decide to join the project, and discouraging from participation those individuals whose values (e.g. more tolerant of proprietary software) conflict with the 'status quo'. Nevertheless, if thought of as a 'contract,' its non-binding and abstract nature would not have been sufficient to keep developers opposing or ill-informed about the tenets of F/LOS software (or devoid of the necessary technical skills) from entering the project. In a context of rapid population growth (which made monitoring of participant behaviour increasingly difficult), this led to the 'membership crisis'. This again provoked institutional change, that is, the implementation of a New Maintainer Process with the goal of guaranteeing that rules such as those codified in the DSC and DSFG would be understood and respected by new developers. The aim was to diminish the risk of malicious, unaware or rebellious individuals entering the project.

The Debian NMP, with its period of "internship", during which a candidate works in close collaboration with a sponsor (mentor) and an Application Manager formalises a legitimate peripheral participation process that had previously been embedded in the day-to-day activities of the community. It specifies a number of criteria that an applicant should fulfil in order to become full member of Debian. An applicant who has gone through the NMP can be expected to know the rules of the project, and to contribute according to them.

Also, by raising the barriers to participation in the project, the NMP discourages casual contributors whose inconsistent participation patterns ("wrong attitude") might negatively affect the quality of the distribution.

Some members of Debian have complained about the severity of the NMP requirements (especially in respect to its philosophical and political topics), and the length of time it takes to finalise it, which create important barriers to entrance into the project. In response to these criticisms, it has been argued that casual applicants to Debian membership need to be filtered out of the NMP as soon as possible in order to avoid wasting the time of volunteer Application Managers, and that the creation of an explicit process for the acquisition of membership makes Debian "more inclusive" than other projects such as the BSD Unix Distribution where, while the

> "…gurus accepted contributions from those who were not already participating on the project, it was difficult to pierce the inner circle of authority and become an actual member of the team. This (…) produced an unacceptable form of project participation, characterized by a degraded elitism that failed to equalize the terrain on which developers could prove their worth". (Coleman, 2005, pp. 12-13)

The NMP establishes clearly the requisites for membership reduces the diversity of participant viewpoints to a manageable level through the enforcement of a specific Debian "mindset" or "attitude" (especially in the area of "software freedom"). It also lessens the costs of co-ordinating decentralised development efforts (and ensuring quality) by guaranteeing that new entrants are conversant with a number of skills and procedures that regulate development inside the project.

Thus, the DSC, DSFG and NMP (and the technical policy manuals new applicants are expected to become familiar with as part of the latter) constitute institutional structures that promote specific goals and techniques inside the project and alignment in the understanding and compliance of new participants. However, this alignment does not preclude the need to make decisions about a multitude of issues such as determining the operational goals of a release, assessing the relative merits of different implementations of the same feature, or the stability of a package,

interpreting legal issues or appraising and punishing undesirable behaviours. The goal of the Debian Constitution is to legitimise decisions on these issues while avoiding excessive concentration of authority in the hands of an elite of developers. Debian achieves this legitimisation by using governance structure that is democratically elected (sustained high quality contributions- i.e. knowledge about the project- still constitute one of the main appeals that a candidate offers), and second, exercised in a limited way without stepping into the "areas of expertise" (packages) of other developers[48].

This emphasis on democratic selection of leaders is not present in technical decision-making, and using political tools such as Debian General Resolutions for these matters is clearly discouraged (Coleman, 2005). It is argued that technical controversies have a right answer that can be reached through argument and discussion, and that *"voting blocks the methodology of open debate"[49]*. Observation of the levels of technical bickering in Debian's mailing list underscores the excessive optimism of this belief: there are several controversial areas where the possibility of closing a debate are curtailed by technical uncertainty and the heterogeneity of developers' goals (at an operational, not necessarily political level); the unwillingness to exercise authority or 'political leadership', perceived to interfere with processes of technical debate has, according to a growing number of developers, impacted the project's release cycle negatively.[50] Discussions between candidates for the project leadership in recent years show dissatisfaction with the actual lack of action from past leaders. It is argued that Debian Officers need to exercise their authority more forcefully in order to "push the project forward".

> "I think that one of the biggest problems Debian is currently facing is the inability to make decisions. There are so many endless, completely futile (and repetitive) discussions going on. We need someone who comes in, tells people to shut up and makes a decision on behalf of the project. A decision people will follow, even if they personally disagree

---

[48] One of the main complaints about the second project leader was his perceived interference with the inner workings of one of Debian's packages, Deity.

[49] Coleman (2005), p. 20.

[50] See 2007's Debian Project Leader debate for evidence regarding the perceived problems that constant bickering and miscommunication have caused in the project (available at http://www.debian.org/vote/2007/suppl_001_debate)

with it. But seriously, do you think our culture would currently accept such a leader? I can tell you from experience that even people who have been asking for a "strong" leader won't actually follow a leader who tells them to take a certain course of action"[51]

This illustrates the strong barriers to modifying Debian's organisational structure to establish more decisive structures and speed up decision-making and conflict resolution processes. Such changes would clash with the main values of the project and would involve redistributions of power that were asymmetric with participants' expertise. The exercise of authority by a leader over another maintainer's package would be difficult to legitimise in this context. According to our conceptual framework, the political processes through which authority is legitimised in Debian are not sufficiently aligned with the project's informational architecture (which favours knowledge fragmentation, and henceforth, 'political decentralisation'). If, as we argue, F/LOS communities only tolerate an individual's exercise of authority over her areas of expertise, it seems that participants in Debian will be confined to decisive decision-making only inside the package they maintain.

The conflict between *de jure* authority enacted through Debian elections, and *de facto* authority acquired via sustained participation and the ensuing accumulation of expertise is also illustrated by the divergences between the Debian officership and what dissenters have called 'The Cabal' (FTP Masters and Account Management). Although there is, in principle, a "division of powers" between these two groups, with the former taking care of high level "vision" and "co-ordination" issues, and the latter of infrastructure and quality, it seems that, once again, goals and policy are not so easy to extricate from the technical substrate where they are embedded, and this causes conflict.

"Constant politicking" tolerated by a DPL without sufficient power to manage the community pro-actively have slowed Debian's development to an extent

---

[51] Post in the blog of a former DPL: http://www.cyrius.com/journal/2006/03/09#being-dpl

some fear might endanger its survival. As a developer argued in his 28th August 2006 resignation note from the project:

> "…over the past few weeks (months? Years?) it's become clear to me that Debian doesn't really seem to know who or what it's for. Arguments erupt over whether something is a deeply held principle or an accident of phrasing on the website; whether we should release more often or less often; whether free software is more important than our users having functional hardware.[52]"

Concerns about the growth of Ubuntu, a GNU/Linux distribution based on Debian but developed more rapidly by a smaller team of paid developers have led to calls for the enforcement of a periodical release rate in Debian, for a more streamlined process for the inclusion of advanced features in stable releases, and even for the creation of a fund-raising structure to pay core developers for their work[53]. Each of these examples aim to change the institutional setting of the project in order to modify its development process and outcomes, and illustrate the kind of debate about project goals (in this case innovation versus stability) that periodically takes place inside the community[54].

---

[52] http://mjg59.livejournal.com/66647.html
[53] http://www.linux.com/article.pl?sid=06/09/21/1623232
[54] See the following discussion at the news-site Slashdot for representations of both points of view: http://linux.slashdot.org/comments.pl?threshold=5&mode=thread&commentsort=0&op=Change&sid=192057

4. Conclusions

In this final section we outline the conclusions and limitations of our analysis and suggest avenues for further research.

A tale about the bazaar

This paper presents a cautionary tale about prevailing bazaar metaphor when applied to the analysis of F/LOS activities, and also when implemented in it[55]. We have focussed on a community where the conjugation of openness and growth has taken its toll in output delivery schedules and social stability. As we have shown, these problems have been tackled through institutional strategies that alter the balance between openness and stability, but that also lead to unintended consequences, such as development stagnation or decreases in participation. These observations suggest caution in employing an idealised model of the bazaar as an analytical metaphor: norms, values, membership processes and structures for governance appear to be essential in order to understand the productive and social dynamics of a project. To the extent that open source ways of working constitute an organisational innovation, it is an innovation with structural implications and contents.[56]

Having established the relevance of focussing on F/LOS institutions, we have illustrated how the epistemic community and legitimate peripheral concepts, and their extension to include the emergence of hierarchies and processes governing authority distribution constitute useful tools for analysis of the evolution of the Debian community. In the case of Debian, this latter process, the delegation of power based on trust, is constrained by the decentralisation of the structures through which development takes place. This explains the difficulties faced by the Debian project leader when trying to exercise any sort of meaningful authority

---

[55] Our focus has been on Eric Raymond's vision of F/LOS development, but our contention that decentralised accounts of the advantages of this model needs to be complemented with an analysis on the actual social and political institutions, and organisational structures through which work in F/LOS projects is accomplished are also relevant to the fruitful research program on User-Centric Innovations carried out by Eric von Hippel and his colleagues (see von Hippel, 2002).

[56] Further elaboration of these ideas may found in Mateos-Garcia and Steinmueller (forthcoming).

over their projects – an experience that stands in contrast with the GNU/Linux kernel or the BSD UNIX distribution. It also illuminates how the nature of technical platform for development (Debian's package management system) influences the social structure of the community. In this sense, our specification of the processes of institutional emergence constrained by the nature of a project's technical architecture complements the classification of F/LOS projects in different categories depending on their purposes, participation patterns and structures, (Nakakoji et al, 2002). It also appears linked to the description of the processes through which projects evolve (as for example, they move from initial stages of experimentation towards stability as more users start relying on them) offered in Nakakoji et al, 2002.

Implications

Our analysis has several implications relevant for F/LOS projects and companies intent on fostering F/LOS communities for support and co-development of their products:

- Institutional structures are essential determinants of a project's outcomes suggesting that special attention should be paid to their design and evolution including the explicit and implicit incentive structures that they create.
- Technical knowledge about a project area is likely to be the primary source of legitimacy for exercises of authority inside it. Therefore, the nature of the platform adopted for development (which determines the form in which knowledge is accumulated or dispersed throughout the community), constrains management actions inside a project.
- Decentralised platforms with components that interlink using standardised interfaces are easier to scale and employ to facilitate parallel development, but they also make it difficult and costly to oversee the quality of contributions, and to exercise authority pro-actively. This may hinder, for example, architectural innovation or, as we have shown in the case of Debian, slow down development speed.

- On the other hand, "vertical structures" where contributions are processed upstream through "gatekeeper" positions (such is the case of the GNU/Linux project) may be overloaded by growth.
- Establishing goals and norms early in the project life may avoid unproductive discussions or struggles for control over the project's direction when the project has accumulated a more diverse group of participants.
- Formalisation of institutional structures through the codification of project norms and values and joining procedures can simplify the negotiation involved in the legitimate participation processes, increase participants' feeling of "ownership" over the project and sharpen the focus of discussions. However, they do not guarantee the elimination of conflict and they may discourage participation as well as being costly in terms of resources.

Limitations and issues for further research

The qualitative, exploratory and situated approach followed in this paper limits the generalisation of its conclusions. Our results are based on evidence that should be interpreted with caution.

In particular, although our emphasis on controversies, disruptive processes and institutional reactions to them provides what we deem to be a healthy counterpoint to other, more optimistic, accounts of these projects' evolution, on the downside it might lead to excessive weight being placed on the opinions and perceptions of conflictive and disgruntled participants or minority groups. In the Spring of 2007, Debian released Debian 4.0 with four times as many packages as its prior release, demonstrating that despite the sharp conflicts that we consider in this paper, the community is still very much alive and successful in its main purpose.

The extent to which our results can be applied to other projects is limited, although preliminary research on other projects (Mateos-Garcia and Steinmueller, 2002) seems to support them. Nevertheless, further research focussing on other large projects such as for example Apache, BSD UNIX, Mozilla or Perl would be helpful. It would be particularly interesting to engage in a larger-scale quantitative

effort using the data available in F/LOS repositories such as SourceForge, Freshmeat or Savannah, in order to determine the extent to which our conclusions can be generalised to a broader collection of (medium size to large) F/LOS projects. This would require an operational definition of several of the variables we have defined in our analysis, such as "conflict", "organisation", "barriers to participation" or "project social performance"-- In terms of community sustainability -- a difficult but potentially rewarding effort.

References[57]

Bachrach, P. and Baratz, M., 1964. Decisions and nondecisions: An analytical framework. The American Political Science Review 57 (3), pp. 632-642.

Bezroukov, N., 1999. Open source software development as a special type of academic research (Critique of vulgar Raymondism). First Monday 4 (10). (http://www.firstmonday.dk/issues/issue4_10/bezroukov/index.html)

Brooks, F., 1982. The Mythical Man-Month : Essays on Software Engineering. Addison-Wesley, Reading, Massachusetts.

Coleman, E.G., 2005. Three Ethical Moments in Debian. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=805287

Crowston, K., Annabi, H. and Howison, J., 2003. Defining open source project success. International Conference on Information Systems (ICIS), 2003. http://floss.syr.edu/publications/icis2003success.pdf#search=%22crowston%20open%20source%20success%22

Edwards, K., 2001. Epistemic communities, situated learning and open source software development. Working paper: http://opensource.mit.edu/papers/kasperedwards-ec.pdf#search=%22kasper%20edwards%20epistemic%22

Elliot, M. and Scacchi, W. 2003. Free Software Developers as an Occupational Community: Resolving Conflicts and Fostering Collaboration. Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work, Santa Isabel Island, Florida.

Ghosh, R., 1998. Cooking pot markets: an economic model for the trade in free goods and services on the Internet. First Monday 3(3). http://www.firstmonday.org/issues/issue3_3/ghosh/index.html

Ghosh, R., Glott, R., Krieger, B., Robles, G., 2002. Free Libre and Open Source Software Survey and Study Final Report. International Institute of Infonomics, University of Maastricht and Berlecon Research GmbH. http://www.infonomics.nl/FLOSS/report/Final4.htm#_Toc13908258

Haas, P. M., 1992. Introduction: Epistemic communities and international policy coordination. International Organization **46** (1): 1-36

Herbsleb, J., Mockus, A., Finholt, T.A., Grinter, R.E., 2000. Distance, dependencies, and delay in a global collaboration. Proceedings of the 2000 ACM conference on Computer supported cooperative work, Philadelphia, Pennsylvania. http://portal.acm.org/citation.cfm?id=359003&coll=portal&dl=ACM

Himanen, P., Torvalds, L., Castells, M., 2001. The Hacker Ethic. New York, Vintage.

Jørgensen, N, 2001. Putting it all in the trunk: incremental software development in the FreeBSD open source project. Information System Journal 11 (4) 321-336.

Lakhani, K., Wolf, B., Bates, J., DiBona, C., 2002. The Boston Consulting Group Hacker Survey Release 0.73. http://www.ostg.com/bcg/bcg-0.73/BCGHackerSurveyv0-73.html

Lave, J., Wenger, E., 1991. Situated Learning: Legitimate Peripheral Participation, Cambridge University Press.

Lerner, J., Tirole, J., 2000. The simple economics of open source. National Bureau of Economic Research, Cambridge MA, Working Paper 7600.

---

[57] All electronically available documents last retrieved on the 28th September 2006.

Levy, S., 1985. Hackers. Dell, New York.

Mateos-Garcia, J., Steinmueller, W. E., 2002. The open source way of working: A new paradigm for the division of labour in software development?. SPRU Electronic Working Paper Series No. 92,
http://www.sussex.ac.uk/Units/spru/publications/imprint/sewps/sewp92/sewp92.pdf

Mateos-Garcia, J., Steinmueller, W. E., (forthcoming). Open, but how much? Growth, conflict and institutional evolution in open source communities. In. Amin, A., Roberts, J. Organising for creativity: Community, economy and space, Oxford University Press, Oxford.

Michlmayr, M., 2003. Managing Debian.
http://www.ukuug.org/events/linux2003/papers/michlmayr.pdf#search=%22managing%20debian%22

Monga, M., 2004. From Bazaar to Kibbutz: how freedom deals with coherence in the Debian project. Collaboration, Conflict and Control: The 4th Workshop on Open Source Software Engineering W8S Workshop - 26th International Conference on Software Engineering. Edinburgh, Scotland. http://homes.dico.unimi.it/~monga/lib/oss-icse04.pdf#search=%22monga%20bazaar%

Moody, G., 2002. Rebel Code: Linux and the Open Source Revolution. Penguin Books, New York.

Nakakoji, K., Yamamoto, Y., Nishinaka, Y., Kishida, K. and Ye, Y., 2002: Evolution Patterns of Open Source Systems and Communities. Proceedings of the International Workshop on Principles of Software Evolution, Orlando, Florida.

O'Mahony, S. and Ferraro, F., 2003. Managing the boundary of an 'open' project. Santa Fe Institute (SFI) Workshop on The Network Construction of Markets.
http://opensource.mit.edu/papers/omahonyferraro.pdf#search=%22o'mahony%202002%20debian%22

O'Mahony, S., (2006). Developing community software in a commodity world. in M. Fisher, M. and Downey, G., Frontiers of Capital: Ethnographic Reflections on the New Economy. Duke University Press, Durham, NC, pp. 237-266.

Robles, G., González-Barahona, J., 2003. From Toy Story to Toy History: a deep analysis of Debian GNU/Linux.
http://es.tldp.org/Presentaciones/200309hispalinux/20/20-en.pdf#search=%22robles%20barahona%20toy%20story%22

Raymond, E. S., 1999. The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. O'Reilly and Associates, Inc., Sebastopol, CA.

Varian, H., Shapiro, C., 2004. Linux adoption in the public sector: An economic analysis.
http://www.ischool.berkeley.edu/~hal/Papers/2004/linux-adoption-in-the-public-sector.pdf

von Hippel, E., 2002.Horizontal innovation networks - by and for users. Sloan School of Management, MIT, Working Paper No. 4366-02. June.

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

Wallach, H., Allan, M., Harries, D., 2005. The Debian New Maintainer Process: History and Aims.
http://www.srcf.ucam.org/~hmw26/publications/debconf5.pdf#search=%22Wallach%2C%20Allan%20and%20Harries%20%22

Weber, S., 2004. The Success of Open Source. Harvard University Press, Cambridge, Massachusetts.