Eindhoven Centre for Innovation Studies

The end of communities of practice in open source projects? Evidence from the Debian case.

B.M. Sadowski & G. Rasters

Eindhoven Centre for Innovation Studies, The Netherlands

Working Paper 05.09

Department of Technology Management

Technische Universiteit Eindhoven, The Netherlands

July 2005

The End of Communities of Practice in Open Source Projects? Evidence from the Debian case

Keywords: Open Source Software community, Community of Practice, Team-based Projects, Debian

Abstract

The paper explores the notion of team-based work that has recently been introduced in the discussion within Open Source Community (OSS) projects. In contrast to open boundaries within a community of practice, project-based teams refer to a relatively small group of collaborating people who operate within clear and relatively stable team boundaries and fit particular functions, roles, and norms. The history of Debian, in particular the discussion in recent leadership elections, has shown that there is an interest in team-based work. After characterizing the stages in the evolution of the Debian project, team-based work is considered as providing novel solutions to the problem of release management and growth problems within the Debian community.

1. Introduction

Open source communities face a dilemma as they have to be organized in a way that allows the creation of new knowledge while at the same time facilitating development of new stable releases of a particular 'free' software package. Open source software (OSS) communities have been characterized as being open, i.e. anyone who wants to join might do so as long as some general behavior rules are followed, involve usually a large number of people and share some (software) source code among its members. Open means that the source code is kept open and available to anyone with an Internet access. The boundaries of OSS are relatively flexible, allowing a rather frequent change of collaborators. The creation of new knowledge in these communities requires, on the one hand, a set of organizational rules and structures that allow critical evaluation of existing knowledge, innovation and rapid elimination of error (Kogut 2000). On the other hand, the growing needs of the open software community reduces the time available for the introduction of new releases while requesting a high quality of new releases (Michlmayr 2004). Due to the dilemma, the organizational forms to coordinate and govern the collaborative work in OSS have to be flexible and adapt easily to changing conditions.

In the literature, OSS have recently been addressed as community of practice (CoP), i.e. a "group informally bound together by shared expertise and passion for joint enterprise" (Wenger and Snyder 2000). The characteristics of this group are related to joint strategy development, the generation of new lines of business, common problem-solving activities, an endorsement to spread best practices and the joint development of professional skills (Wenger and Snyder 2000). These characteristics are distinct from project-based teams, which refer to a relatively small group of collaborating people who operate within clear and relatively stable team boundaries and fit particular functions, roles, and norms (Hertel et al. 2002). Interestingly, the emergence of CoP has been a reaction to overcome the shortcomings of firm based models of knowledge creation (Lee and Cole 2003). The need to coordinate and govern collaborative work that requires the development of creative knowledge within OSS while adapting to the needs of the growing open source community seems to have made team-based project work methods more attractive to its members. Team-based project work has certain advantages as in these projects clear targets can be set, the boundaries and rules are clear which guarantees continuity in the management of new releases.

In this context, we link the call for more team-based projects within the OSS to perceived problems of release management. Paradoxically, these calls are part of the drive within OSS to evolve to more professional knowledge management practices within OSS in order to deal with the needs of the open software community.

In the following we briefly characterize the appropriateness of the concept of community of practices (CoP) to describe the evolution of OSS. Afterwards, we focus on the emergence of CoP and recent calls for new teams based project organization within the Debian community. We conclude with a brief discussion of our findings.

2. The emergence of community of practice in the evolution of OSS

In the evolution of OSS communities, three major stages have been distinguished which require different organizational forms of collaborative project work: the project initiation stage, the process of going "open" and the project growth, stability or decline stage (Schweik and Semenov 2003; Rasters 2004). In the project initiation stage, OSS projects commence because one or more people realize that there is a computing-related problem or challenge left unfilled, and for one or more reasons, they decide to take it on (Godfrey and Tu 2000). Here the "itching problem" described by E. Raymond comes into play: "every good work of software starts by scratching a developer's personal itch." (Raymond 1998). At that point it is important to reach programmers who think along with this new initiative. Motivation, "the kernel," and a modular design are three important components of this stage of an OS project (Schweik and Semenov 2003). Even if there is an increasing number of studies that have focused on the motivation of programmers to take part in OSS communities (Hertel, Niedner et al. 2003), the motivations of the initiators to start up a new project have just recently received some attention in the literature. Schweik and Semenov (2003) have characterized these motivations as related to meet some personal need, to work on the leading edge of some technology; to address some software crisis; and/or to provide intellectual stimulation. Furthermore, there might be some socio-political motivations related to the sheer enjoyment to do the work and/or an interest in taking on a technical rival. Even economic considerations like skill-building and low opportunity costs (e.g., nothing to lose by undertaking the project) have been likely motivations for initiators to start a programming project (Schweik and Semenov 2003). The second component of the initial stage is related to importance of an initial product for others to build upon — what has been called the project core, or the kernel. The initial project kernel has to show some promise, in order for other virtual members to join in. The third critical component is a good design and the concept of modularity. Modularity allows programmers to work in parallel. This modularity also enables the project leader to keep better control over the project when the work progresses (in complexity)(Rasters 2004).

In order to enter the going open stage, OSS projects face certain challenges such as achieving project and product credibility, developing adequate communication mechanisms, creating effective recruitment strategies as well as developing appropriate institutional and governance design. To achieve project and product credibility, the project needs to obtain support from a number of enthusiastic "core developers", to show some "plausible promise" (i.e., a high development potential of the kernel in conjunction with an existing enthusiastic programmer community of high reputation), to attract interest from programmers due to its innovativeness, to have some importance while allowing a (future) large number of developers to participate, and to demonstrate that the right amount of the problem has already been solved before the project becomes "open." (Schweik and Semenov 2003). In order to develop appropriate communication channels different internet based forms of communication are exploited ranging “free form” discussions (e.g. mailinglists, IRC channels), to strongly structured discussions (e.g. bug tracking systems or trouble ticketing at helpdesks), to knowledge based discussions (e.g. wiki platform). To create effective recruitment strategies, the initiator has to choose a platform for announcing the project that has the potential of reaching as many readers as possible. Nowadays this function is integrated in central websites for project hosting. The development of appropriate institutional and governance designs has been critical in the success and failure of open source projects. Markus et al. (2000) outlined four interrelated coordination mechanisms that decide over the faith of open source projects. The way these projects manage their membership, enforce rules and institutions, the mechanisms of monitoring and sanctioning, and, in particular, creation of reputation. Due to the interaction of these governance mechanisms, open source projects can successfully evolve despite their inherent potential for chaos (Markus, Brook et al. 2000).

At the going open stage, OSS projects resemble a community of practice (CoP), i.e. a group of programmers that informally interact with each other based on shared expertise and enthusiasm for the common goal of the project. (Wenger and Snyder 2000). The characteristics of the CoP are related to specific processes of knowledge creation and possession in which a 'collective' good is accumulated that is owned by members of the OSS; open membership as long as the members contribute on continuous basis to the creation of knowledge to the development of new software packages; the voluntarily motivation to engage in software development; a knowledge distribution that extends beyond the boundaries of the firm and a mainly technology mediated communication among the members of the OSS (Lee and Cole 2003; van Hippel and von Krogh 2003; Rasters 2004).

In reaching the third stage (project growth, stability or decline) these characteristics of CoP have become the center of debate as performance goals, i.e. the introduction of new releases, are getting increasingly difficult to achieve. The dilemma within OSS at this stage is to adapt to the growing needs of the open source community by providing more rapidly new software releases while assuring a high level of quality. In order to manage these emergent challenges there have been calls within OSS for more team-based oriented approaches. In the following we characterize this evolution by describing the history of the Debian project and the current discussion on Debian mailing lists.

3. Evolution towards team-based management approaches? The case of Debian

More than 900 volunteer package maintainers are currently working on over 8250 packages and improving Debian GNU/Linux. Debian is a free Operating System (OS). Debian uses the Linux kernel (the core of an operating system), but most of the basic OS tools come from the GNU project (GNU is a recursive acronym for "GNU's Not Unix"); hence the name GNU/Linux. Debian is being developed cooperatively by an increasing number of developers mainly through the Internet. In the following, we briefly characterize the first two stages of the Debian project while focusing on the current third stage and the discussion within the Debian community around team-based projects (Rasters 2004).

3.1. The Project Initiation Stage

"Like any area of endeavour, Open source projects are initiated because one or more people realize that there is a computing-related problem or challenge left unfilled, and for one or more reasons, they decide to take it on." (Schweik and Semenov 2003). Ian Murdock started the Debian project from scratch after being dissatisfied with the SLS release. Ian Murdock wanted to "draw a few people out the woodwork", and had put down a request for comments, suggestions and advice. He made clear that he was developing an initial product for others to build upon. In 1993, when Ian Murdock decided to start an Open Source distribution that would always be free, he found a group of like-minded people to work with him. The stated goal was to create a complete operating system that would be 'commercial grade' but not, in itself, commercial. Ian Murdock posted his intentions to the Usenet in August of 1993 and immediately found outside interest, including that of the Free Software Foundation, the creators of much of the core software of all Linux-based systems. Murdock credits this early interest as being pivotal to the acceptance of Debian into the free software world. Murdock posted his announcement in order to try and reach out for a small group of motivated individuals who had ideas for the project. Or as Varghese puts it: "In 1993, when Ian Murdock decided to start an Open Source distribution that would always be free, he found a group of like-minded people to work with him. The question of freedom was important to Murdock (...). It started as a small, tightly-knit group of free software hackers, and gradually grew to become a large, well-organised community of developers and users (Varghese 2003).

3.2. The Going Open Stage

When Ian Murdock felt that Debian was ready to be shared, he made the official announcement on the Internet, and encouraged others to help him to improve it. On 2 September Murdock officially announced the Debian project. This announcement was made on the same Linux newsgroup (c.o.l.a = comp.os.linux.development newsgroup) he also re-posted his two earlier postings about Debian. However in this official posting he released the name of the Debian mailinglist which should be used for the project.

Ian Murdock decided to follow the Open Source Developers licensing principles; he made the decision to follow the GNU and receive a General Public

License (GPL). Debian GNU/Linux is a strong supporter of free software. Since many different licenses are used for software, a set of guidelines, the Debian Free Software Guidelines (DFSG) were developed to come up with a reasonable definition of what constitutes free software. Only software that complies with the DFSG is allowed in the main distribution of Debian. The Debian developers of the Debian GNU/Linux system have also created the Debian Social Contract. The DFSG is part of the contract. Initially designed as a set of commitments that they agreed to obey, they have been adopted by the free software community as the basis of the Open Source Definition.

The Debian 0.91 release gave a first glimpse of the Debian philosophy. By this time, a dozen or so people were involved in development, though Ian Murdock was still largely packaging and integrating the releases himself. After this first public release of Debian, attention was turned toward developing the package system called dpkg. A rudimentary dpkg existed in Debian 0.91, but at that time this was mostly used for manipulating packages once they were installed, rather than as a general packaging utility. By the summer of 1994, early versions of dpkg were becoming usable, and other people besides Ian began joining the packaging and integration process by following guidelines that explained how to construct packages that were modular and integrated into the system without causing problems. By fall 1994, an overloaded Ian Murdock, now coordinating the efforts of dozens of people in addition to his own development work, transferred responsibility of the package system to Ian Jackson, who proceeded to make many valuable enhancements, and shaped it into the current system. After months of hard work and organization, the Debian Project finally made its first distributed release in March of 1995, Debian 0.93 Release 5. Debian 0.92 had never been released, and Release 1 through Release 4 of Debian 0.93 had been development releases made throughout the fall and winter of 1994. Table 1 provides an overview of Debian releases and major events during this second phase.

Insert Table 1 about here

By this time, the Debian Project, as it became known, had grown to include over sixty people. In summer 1995, Ian Murdock transferred responsibility of the base system, the core set of Debian packages, to Bruce Perens, giving him time to devote to the management of the growing Debian Project. Work continued throughout the summer and fall, a final all-out binary format release, Debian 0.93 Release 6, was made in November 1995 before attention turned to converting the system to the ELF binary format. Ian Murdock left the Debian Project in March 1996 and Bruce Perens assumed the leadership role, guiding the Project through its first Buzz release, Debian 1.1, in June 1996. In the period between 1996 and 1999 there were three more stable releases.

In 1999 Debian entered the phase in which the community became really concerned about the quality of maintainers joining the project. There was even a freeze on accepting new maintainers. A crisis occurred when the project no longer felt that it could adequately protect its boundaries and closed its doors to new potential members. As the acting project leader W. Akkerman at that time observed: "I have to acknowledge that Debian has reached the point where it has grown too much and cannot continue as before. At the moment we already have chaos all over with no proper leadership. Only very few people are taking care of general management tasks. Remember this is an association of >500 people. There is still no proper management. Guess what would have happened if it were a company..."

This led to the constitution of the New Maintainer Process and the articulation of membership criteria and a process, thereby institutionalising the openness of the Debian project. The Debian New Maintainer process is a series of required proceedings to become a Debian developer. This new maintainer approach has been a way of keeping Debian open, but at the same time a way to manage its boundaries. It was also a way to finetune a new social structure that could help ensure that new member's skills, goals, and ideology were in line with that of the collective. (O'Mahoney and Ferraro 2003). From 1999 onwards there were three other releases, however there was a gap of three years between the 3.0 release in 2002 and the last Sarge release in 2005.

3.3. The project growth phase: Towards Stabilization and Adaptation to the Need of the OSS

As Schweik and Semenov (2003) observe, open source projects can grow at this stage based on new membership. They can remain stable relying on the same number of participants as in the going open stage, or they gradually might decline due to a lack of interest of participants (Schweik and Semenov 2003). The willingness of participants to continue their cooperation in a particular project is related to past progress in areas such as project and product credibility, the development of adequate communication mechanisms, the creation of effective recruitment strategies as well as the development of an appropriate institutional and governance design. As has been shown in Table 1, from its initiation phase to the growth phase the Debian project was developing rapidly from only a few developers into a large community. During this growth the community found ways to cope with this expansion, mainly by streamlining and coordinating the communication. Their communication processes were streamlined in different ways, for instance by using mailinglists as well as providing procedures and rules for communication. Structured communication is needed in such a complex environment. The Debian project had to deal with an increasing number of packages, which has to be coordinated through the implementation of different procedures. However, not only the package management has been a complex issue, but also the management of the open source community required some effort. As the 'real' programming work was undertaken in small groups, it became important that programmers would still feel as part of the Debian community as a whole. Or as Lameter (2002) puts it: “Being one among 1000 developers also makes the individual rather anonymous. The attraction in the past for many developers was the personal relationships that develop in the project. We need to reorganize the project into smaller groups where these significant relationships can develop" (Lameter 2002).

Another issue that is often brought up in the discussion of the management of the community was the slow release cycle of Debian. The Debian project had often to defend itself on this matter. The Debian community has always been proud of the fact that it will not release buggy software, and will release only when the software is stable. For a commercial company this would be hard to achieve. As the introduction of a new release is commercially driven. However in the Debian community, work is done by volunteers, mostly in their spare time. Their rule is: "we release when it is ready!" There are no deadlines. "Of course, it takes us a long time before we release, however, when we release, our users know they can trust the software." As Ian Murdock puts it in an interview: "I agree that the slow release cycle is a problem. The Debian folks recognize it as a problem too and are taking steps to address it. Release management is very hard, especially when you're dealing with hundreds and hundreds of people, many of whom have never met and most of whom work on the thing purely as a hobby. It's far easier when you have a company and people are all in the same place and getting paid. So, this is a common problem among free software projects, and Debian has to deal with it on a scale larger than most projects have had to deal with it. And they're getting there."

There is evidence that shifts in leadership approaches are related to the slow release management of the last stable versions, as well as the problems with the growth of the community. As Table 2 shows, since 1993 the Debian project has been headed by a number of leaders with very different leadership styles. There have been experiments in leadership style. In the beginning when there were only a few people involved in the Debian project, a strong leadership was accepted. However, with the growth of the Debian community this style of leadership did not work anymore and the project was running into managerial problems. This was the point when leadership elections were arranged. The ways in which elections were organized also grew over time, from simple plain text mission statements on personal election platforms to election debates on IRC channels.

Insert Table 2 about here

Ian Jackson led the Debian project from January 1998 until December 1998. This was the point in time when the project leaders became elected. The enormous growth of the community prohibited informal ways of transferring leadership. Jackson tried, together with the community to "fit the governance structure" to the size of the community and to the feelings of freedom that lived in the community. Ian Jackson had major influence on how the Debian project become structured with respect to writing the constitution, election methods and the description of leadership models.

In 2000, the leadership debate and a speech of the opponents were introduced in the election. The debate was held on Tuesday, February 15, 2000 at 1900 UTC, at the irc.Debian.org on channel #Debian-debate. This is an a-synchronous chat channel, where everyone could log in. The format of the election was as follows: 24 hours before the debate each of the candidates e-mailed an 'opening speech' to the debate organizer, Jason Gunthorpe. They were then placed on this page. Everything was added at the same time to ensure fairness. The actual debate had two parts. First, a strongly moderated traditional debate: The moderator asked a candidate a question. The candidate then had a reasonable period to answer. After the answer each of the other candidates responded in turn. The first candidate was allowed to make closing remarks on the question. The order of the candidates was rotated for each question. The second part of the debate was more freestyle. Questions submitted by the audience and developers were asked. Each candidate got a short period to respond. After the debate a log of the debate was posted, so voters could read everything at their own pace. In the leadership elections of the year 2005 a major difference with previous leadership elections emerged.

3.4. The call for team-based project approaches

The year 2005 has been a very interesting one in the evolution of the Debian community. The Debian GNU/Linux version 3.1 codenamed "Sarge" was released after nearly three years of constant development. Within the open source community, criticism increasingly mounted against the slow release management cycle of the project. Within the leadership elections, the slow release management and the growth of the user community were considered as "hot" items among candidates running for election even if this issue had already intensely been discussed in previous elections. Interestingly, candidates running for election presented this time new solutions to these critical issues. They suggested a whole new approach towards leading the Debian project. The election platforms of two running candidates Branden Robinson and Andreas Schuldei suggested forming a small formal team of Debian developers aimed at supporting the project leader. This team, nicknamed "Project SCUD", was organized in the last few weeks of 2004. The team consists of six developers. The basic idea behind the team-based approach was to better distribute the workload and to have the most appropriate person being responsible for a particular area of expertise. We will focus now, in more detail, on the statements made by the two candidates (Andreas Schuldei and Branden Robinson) running for the elections, which are part of the SCUD team. Afterwards we will characterize the discussion within the community as a whole as a response to these statements.

According to the SCUD members having a Debian project leader (DPL) team would allow them to distribute the workload, avoid burnouts and problems related to real-world unavailability of individual developers. In previous election platforms it became obvious that candidates running for election favored specific tasks more than others even if they were related to the function of a Debian project leader. While being part of the DPL team it is possible to micro-delegate tasks to the most appropriate person, which makes certain tasks more enjoyable for them than for others and lets them to deal with them more efficiently.

The SCUD team identified small teams (up to seven people) as probably the single most important unit(s) for the Debian project to grow in a healthy way. If the team functions well it can solve more problems than individual developers. The SCUD team strives for low fluctuation/turnover and stability within the team. The team should be able to provide a *smooth entry point* for new developers to gain proficiency and develop skills. Furthermore, teams should be the place where developers can get to know each other quickest and best (due to the small number of people in the group). Another advantage is that people can form a *knowledge pool* when cooperating on package maintenance, infrastructural or organizational tasks, and it is less likely that such pool would get lost compared to the knowledge and skills lost if a single developer is departing. This would make Debian more resilient against unmaintained packages or head hunters. Because these teams can grow and divide on their own, they are *self-organizing* and provide for *very good scalability* in numerical growth. An example of team-based work being organized in the Debian project is provided by Andreas Schuldei who argues that the Debian project needs more frequent, regular releases since the present delays cause frustration and a decline in morale in the Debian community. To pave the way for a smoother development cycle and release process he took the initiative to organize a team-based meeting of the release team and FTP-masters.

While the members of the SCUD team have been enthusiastic about their new ideas, there has been some controversy within the Debian community about the Project SCUD, which has also been referred to as a self-appointed group of advisors to the Debian Project Leader. The SCUD proposal has been a source of some concern, especially how it would integrate within the constitution and the existing organisational structure. We will briefly summarize a few of the arguments against the idea of small teams within the Debian community.

The discussion on the mailinglist shows that members of the Debian community got confused by the DPL team idea. They argue that the DPL can always delegate tasks to other members of the project and therefore the argument of SCUD members that it is impossible for a single DPL to have time to do everything is not valid. “Why can't the DPL simply immerse in the developer community and consult with individual developers, or all of us, depending on the challenge at hand? Why the need for a closed council, which will surely employ closed means of communication among its members? Why not consult in public so we all know how our project is actually being led?” Other members have become more concerned about the constitutional implications of the SCUD team, since the Debian Constitution does not define the DPL's function as a team; it only defines the DPL's function, that of the Project Secretary, the Technical Committee, of Delegates, and of the Project's Developers. By excluding bodies that are of no relevance to the DPL's position, there are only two options:

- The members of Project SCUD (other than the DPL himself) do not actually have any real power, except that the DPL will supports them if any of their decisions are challenged (thus, their power will only exist *de facto*);
- The members of Project SCUD (other than the DPL himself) will be formally appointed as delegates (thus, they will have real power, backed by the constitution).

One main argument against the SCUD team has been that a DPL team should not be a subset of members, but should be open to everyone. Basically there should not be any issues that could not be discussed within everyone. Debian members felt offended by the idea of private meetings between SCUD members. This issue of private meetings came upfront during the Vancouver Meeting discussion 2005, at which a small group of ftp master gathered in a private face-to-face meeting.

Further question marks have placed by Debian community members as to whether or not the creation of a small team increases Debian's transparency or even worse diminishes the openness of the overall Debian project. There have been great concerns from members about attempts to formalise the SCUD team. This formalization can cause new problems such as dispute resolution, unclear areas of responsibility, opaque team member selection processes and further separation between "average" developers and "management". In short: with the SCUD team, there is a threat of cabal. The cabal issue has frequently come to the forefront between Debian community members.

Within the Debian community, there are Debian-private mailinglist that are only accessible for Debian developers and to which almost every developer is subscribed. This is the place were issues are discussed that are unknown for outsiders (e.g. non-developers). However within the Debian-private mailinglist, there has been some rumors that there is also an inner "core-core": the Cabal. A cabal has been defined as consisting of a number of persons united in some close design, usually to promote their private views and interests in church or state by intrigue; a secret association composed of a few designing persons; a junto. However every clue is missing on the cabal within the Debian community and these have been just rumours. As a Debian members explains "The Cabal is like the inner circle. However it is used more like an inside joke. Of course, there must be some truth in it that the 'old-timers' have some more privileges than the newbies in Debian, and of course some of the crucial tasks are done by a small group of people." The SCUD team has brought the cabal discussion again to the fore, in particular with respect to its implications given that it will be a success. Will the team remain the same, or will other people be appointed? If the former is true, what guarantees that accusations of being some sort of 'cabal' will not also become true? If the latter is the case, how will the appointing procedure be implemented? Will the team simply cease to exist, leaving the next DPL to make up his own team, or will the team choose new members among the existing developers? Or will we have a 'DPL Team Election' next year, rather than a 'DPL election'?

Even as the SCUD team has only be in existence for a few months now, it is currently not clear what the prospects for this new team-based approach are. The “Project SCUD” held its first meeting on 24 April 2005. The meeting agenda was mailed to the Debian-project mailing list shortly prior to the meeting. This meeting caused another discussion about the openness of meetings within the Debian community, since the SCUD team declared that is was not an open meeting. However, the minutes of the meeting would be posted to the mailinglist. “We continue to grapple with ways to balance openness and accountability with the discretion that some issues sometimes require…”

While roles and functions are being settled in the SCUD project it has become clear that is characteristics of a team as opposed to a community of practice. While there are a number of differences between the two different organizational forms, the most important difference is the way participation is determined. Within the SCUD team, the participation is 'binary': you are in or you are out. Within the Debian community as a whole the membership is voluntary and participation is determined by social rules and relationships (belong to the core or to the periphery). Debian shows that teams and communities might exist together and members may belong to both simultaneously. However, we believe these are two separate organizational structures with very different behavior patterns, operational rules and roles. So far, not much attention has been paid to these new structures of team-based work within open source communities.

4. Discussion

In exploring the different stages in the development of OSS communities, the paper has traced the notion of team-based work to the growth stage within the Debian community. Team-based work is considered as providing novel solutions to the problem of release management and growth problems within the Debian community. As community of practices has been based on open boundaries, project-based teams seem to signal a return to more closed work practices within OSS communities. The proponents of team-based work consider these work practices as a solution to the dilemma of OSS related to the growing needs of the open software community reduces the time available for the introduction of new releases while requesting an even higher quality of new releases. Opponents consider these forms of work as a deviation from the traditional openness paradigm within OSS.

It will become interesting to see whether or not these new forms of organization will become implemented and even dominant in OSS environments in particular in the Debian project. From the theoretical point of view, the analysis of these organizational forms might provide new insights into the extent to professional expertise gained in software programming has been influential for OSS and the extent to which these new organizational forms might (co-)exist alongside community of practices.

References

Godfrey, M. and Q. Tu (2000). Evolution of Open Source Software: A Case Study. ICSM-00, San Jose, CA.

Hertel, G., S. Niedner, et al. (2003). "Motivation of Software Developers in Open Source Projects: An Internet-Based Survey of Contributors to the Linux Kernel." Research Policy **32**: 1159-1177.

Kogut, B. (2000). "The Network as Knowledge: Generative Rules and the Emergence of Structure." Strategic Management Journal **21**: 405-425.

Lameter, C. (2002). Debian GNU/Linux: The Past, the Present and the Future. Free Software Symposium 2002, Japan Education Centre.

Lee, G. and R. Cole (2003). "From a Firm-Based to a Community-Based Model of Knowledge Creation." Organization Science **14**(6): 633-648.

Markus, M., M. Brook, et al. (2000). "What Makes a Virtual Organization Work: Lessons from the Open Source World." Sloan Management Review **42**(1): 13-26.

Michlmayr, M. (2004). Managing Volunteer Activity in Free Software Projects. 2004 USENIX Annual Technical Conference, Boston MA.

O'Mahoney, S. and F. Ferraro (2003). "Managing the Boundary of an Open Project." Harvard Business School Working Paper.

Rasters, G. (2004). Communication and Collaboration in Virtual Teams. Did We Get the Message? Nijmegen, Ipskamp.

Raymond, E. (1998). "The Cathedral and the Bazaar." First Monday **3**(3).

Schweik, C. and A. Semenov (2003). "The Institutional Design of Open Source Programming: Implications for Addressing Complex Public Policy and Management Problems." First Monday **8**(1).

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

Van Wendel de Joode, R. (2002) Coordination and collaboration in Open Source Communities, in: Ten Heuvelhof, E.F. (2002) Proceedings of the first International Doctoral Consortium on Technology, Policy and Management, ISBN 90-5638-094-x.

Varghese, S. (2003). "Living Up to the Linux Name." The Age.

Wenger, E. and W. Snyder (2000). "Communities of Practice: The Organizational Frontier." Harvard Business Review **January-February**: 139-145.

Table 1: New releases and important events in Debian History

Timeline	Release	Packages	Developers	Events
Fall-Winter 1993	Several Internal Releases			
January 1994	Public Release of Debian 0.91.			A dozen or so people were involved in development, though Ian Murdock was still largely packaging and integrating the releases himself.
Summer 1994				Other people besides Ian Murdock began to join in the packaging and integration process by following guidelines that explained how to construct packages that were modular and integrated into the system without causing problems.
Fall 1994				Ian Murdock, transferred responsibility of the package system to Ian Jackson, who proceeded to make many valuable enhancements, and shaped it into the current system
1995	First distributed release (Debian 0.93 Release 5)	250	60	The Debian Project, as it had come to be called, had grown to include over sixty people.
Summer 1995				Ian Murdock transferred responsibility of the base system, the core set of Debian packages, to Bruce Perens, giving Ian Murdock time to devote to the management of the growing Project.
March 1996				Ian Murdock left the Debian Project in March of 1996; Bruce Perens assumed the leadership role.
June 1996	1.1 (Buzz)	474	90	
End 1996	1.2 (Rex)	848	120	
1997	1.3 (Bo)	974	200	
1998	2.0 (Hamm)	1500	400	
1999	2.1 (Slink)	2250	410	Freeze on accepting new maintainers. Constitution of the New Maintainer process
2000	2.2 (Potato)	3900	450	
2002	3.0 (Woody)	9000	1000	
2005	3.1 (Sarge)			Leadership elections within a new format oriented towards team approach.
	(Etch), no release date set yet.			

Source: (Lameter 2002) and own information

Table 2: Changes in leadership style and organization

Phase in Debian history	Year	Project Leader	Style of leadership	Changes in organization
Inititation and Going open phase	1993 – March 1996	Ian Murdock	Founder, visionary	
Growth phase	April 1996 – December 1997	Bruce Perens	Nominated, "strong leader"	Open community of practice
	January 1998 – December 1998	Ian Jackson	Formal style and strategic vision	First project leader elected, Jackson only candidate
"Growth Crisis"	1999 – 2001	Wichert Akkerman	"Relaxed informal style"	Elected twice, leadership debate and speech of opponents
	April 2001 – April 2002	Ben Collins	More visibility as a leader	
	April 2002 – 2003	Bdale Garbee	Networker and "Facilitator", Spokesman for Debian	
	2003 – 2004	Martin Michlmayr	Motivator and (internal) coordinator	
	2005	B.Robinson	Coordinator	Calls for closed team-based project work

04.15 B. Verspagen
Innovation and Jobs: a Micro-and-Macro Perspective

04.16 V. Gilsing & B. Nooteboom
Density and strength of ties in innovation networks: an analysis of multimedia and biotechnology

04.17 G. Silverberg & B. Verspagen
The size distribution of innovations revisited: an application of extreme value statistics to citation and value measures of patent significance

04.18 J. Jacob
Late industrialisation and structural change: the Indonesian experience

04.19 C. Garavaglia
"History Friendly" Simulations for Modelling Industrial Dynamics

04.20 K.H. Heimeriks, G.M. Duysters, W.P.M. Vanhaverbeke
The evolution of alliance capabilities

04.21 K.H. Heimeriks & G.M. Duysters
A study into the alliance capability development process

05.01 V.A. Gilsing & B. Nooteboom
In search of the origins of novelty: exploring novel combinations in allopatric speciation

05.02 V.A. Gilsing & C.E.A.V. Lemmens
Strategic alliance networks and innovation: a deterministic and voluntaristic view combined

05.03 M.C.J. Caniëls & H.A. Romijn
What Works, and Why, in Business Services Provision for SMEs: Insights from evolutionary theory

05.04 C. Macleod & A. Nuvolari
'The Ingenious Crowd': A Critical Prosopography of British Inventors, 1650-1850

05.05 B. Nooteboom, W.P.M. Vanhaverbeke, G.M. Duysters, V.A. Gilsing, A.J. van den Oord
Optimal cognitive distance and absorptive capacity

05.06 P. Criscuolo & B. Verspagen
Does it matter where patent citations come from?Inventor versus examiner citations in European patents

05.07 G. Silverberg & B. Verspagen
Self-organization of R&D search in complex technology spaces

05.08 H.A. Akkermans & J.E. van Aken
Strategic decision-making in turbulent setting: creating strategic momentum

05.09 B.M. Sadowski & G. Rasters
The end of communities of practice in open source projects? Evidence from the Debian case.