UNITED NATIONS
UNIVERSITY

UNU-MERIT

Working Paper Series

#2007-039

Transition of Governance in a Mature Open Software Source Community: Evidence from the Debian Case

Bert M. Sadowski*, Gaby Sadowski-Rasters and Geert Duysters*

* Corresponding author

Transition of Governance in a Mature Open Software Source Community: Evidence from the Debian Case[1]

Bert M. Sadowski*[2], Gaby Sadowski-Rasters[3] and Geert Duysters*[4]

Abstract

As flourishing, productive open source software (OSS) communities mature, they have to introduce a variety of governance mechanisms to manage the participation of their members and to coordinate the launch of new releases. In contrast to other modes of governance of OSS communities, the Debian community introduced new mechanisms of informal administrative control based on a constitution, elected leaders and new functions attributed to interactive communication channels (like mailing lists or IRC channels) that can provide for community effects (and feedback). We show that these control mechanisms were introduced as a response to emerging innovative opportunities due the usage of source packages and heterogeneous learning processes by different groups within the Debian community,

UNU-MERIT Working Papers
ISSN 1871-9872

Maastricht Economic and social Research and training centre on Innovation and Technology, UNU-MERIT

UNU-MERIT Working Papers intend to disseminate preliminary results of research carried out at the Centre to stimulate discussion on the issues raised.

1   Introduction

The continuing fascination with open source software (OSS) communities has led to an explosion in the number of volunteers working in Open Source Software (OSS) communities. The continuous growth of these communities in combination with the increased demands on the open software community has, however, created mounting problems for these same communities in terms of organization and governance. The traditional ways of organizing these communities have proved to be unable to cope effectively with these conditions of exponential growth.

In OSS communities, the creation of new knowledge 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 need 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 this dilemma, the organizational forms to coordinate and govern collaborative work have to be flexible and should be able to adapt easily to heterogeneous learning conditions within different groups in OSS communities. The Debian OSS community fits this general picture with the number of developers increasing from a sheer total of 60 in 1996 to over 9000 in 2005 and with the amount of source packages

rising from 250 in 1995 to 10.869 in 2006.[5] During this roughly ten years period, the growth of the Debian OSS community was accompanied by experimentation with different governance forms based on informal hierarchy after the original founder, Ian Murdock, left Debian in March 1996. Debian is a free Operating System (OS). It 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 very similar to OSS projects like RedHat and SuSE whose Linux strategies focus primarily on the application of Linux for enterprises (e.g. Red Hat Enterprise Server, SUSE Linux Enterprise Server, Novell Open Enterprise Server/Linux, Novell Linux Desktop)

As a respond to mounting organizational challenges Debian as well as other OSS communities like Apache or Linux, came up with new ways of organizing distributed work that differed from traditional work practices as experienced in professional organizations (Franke and von Hippel, 2003, Lee and Cole, 2003, Moon and Sproull, 2000). In contrast to other OSS communities, the Debian case shows that an OSS community can develop new governance mechanisms in the face of increasing technical and structural complexity from a "great person" in charge (Moon and Sproull, 2000) to informal administrative control mechanisms based

---

[5] End February 2006

on a constitution, elected leaders and new functions attributed to interactive communication channels (like mailing lists or IRC channels) that can provide for community effects (and feedback) (Sadowski-Rasters, Duysters, & Sadowski, 2006). The Debian case shows furthermore that informal administrative control mechanisms are a way to foster heterogeneous learning processes within OSS communities.

In the following we briefly characterize the theoretical discussion on changes in organizational structure and governance of OSS communities during their transition from the "going open" to the "growth" stage (Lameter, 2002). Afterwards, we focus on describing different governance mechanisms in the Debian OSS community after the initial founder, Ian Murdock, left the project in March 1996. In using data triangulation, the analysis utilizes a variety of data sources to characterize perspectives of different stakeholders on the governance forms within the Debian OSS community. In this piece we try to answer our main research question of how alternative governance mechanisms have revolutionized an OSS community such as Debian. Finally, we conclude with a brief discussion of our findings.

2 Mature OSS Communities and their Governance Forms

Open source software (OSS) communities are characterized by distinctive features such as a) a shared common interest of members communicating through the Internet without face-to-face contact (Hertel et al., 2002, Rheingold, 2002); b) active pursuing of collective innovation and production processes (Hemetsberger, 2002); c) members bound together by shared as well as complementary expertise, which makes it possible to manage complex projects (Hertel et. al, 2002); and d) are based on reciprocity on the group level as individuals adding code (or providing for other activities) to the group project, receive something from the group in return (for instance other code or bug reports). In contrast to collaborations, OSS communities are less restrictive in their access policy, relying on referral or reputation and develop a more specific community code including sanctions for violating this code. Furthermore, they are less flexible compared to collaborations with respect to change of members in the community. Compared to project based teams OSS communities are less clearly defined and less stable with respect to boundaries, functions, roles, and norms. They are more similar to "communities of practice" (Wenger & Snyder, 2000) which emerge based on "informal and self-organizing" mechanisms and "benefit from cultivation". However, to sustain these "communities of practice", they have to managed (Wenger & Snyder, 2000).

For OSS communities, a critical growth stage is reached at the moment they are moving from the project initiation stage to the stage of "going open" (Rasters, 2004, Schweik and Semenov, 2003). This stage of "going open" can be seen as being critical in

determining whether the OSS project will grow further, reach stability or decline. The main challenge for OSS communities has always been to find an appropriate governance form for this new stage of the OSS community. In this paper we aim to shed more light on this transformation process.

Within organization theory, governance has been characterized as a toolbox for control, supervision and monitoring economic activity. It is aimed at achieving motivation and convergence of different objectives between all members of a group (Ouchi, 1979). Organizational life cycle theorists have shown that the internal structure of organization is changing by going through different growth stages (introduction, growth, maturity or decline). At these stages, appropriate governance mechanisms have to be found that can deal with increasing technical and structural complexity otherwise organizations decline. This discussion, rooted in original contributions by Blau (1970 and Woodward (1965), has shown that organizations cope with increasing technical and structural complexity by increasing differentiation and formalization as well as by employing a larger administrative component.

In coping with growth, OSS communities have deployed a wide variety of differentiated task structures with different degrees of formalized technical as well as administrative structures. The formalization of the technical and administrative structures has been driven by the needs within the OSS community to explore and

exploit knowledge leading to a parallel code structure of the open source software project (Lee and Cole, 2003). The evolution of these tasks structures and formalized structures also required different forms of governance within OSS communities. OSS communities have struggled most with the increasing complexity of the software and the explosion in the number of contributors to the community. This makes coordination in OSS communities a critical issue that separates successful from un-successful communities.

To deall with the increased need for coordination within OSS communities, Demil and Lecocq (1999) have shown that the bazaar structure, i.e. a "great babbling bazaar of different agendas and approaches" (Raymond, 2001), can serve as a new emerging mode of governance within the OSS community (Demil and Lecocq, 1999, Raymond, 2001). Even under conditions of very high uncertainty, the bazaar mode of governance assures coordination based on reputation effects that are induced by the community phenomenon. However, in the face of increasing technical and structural complexity of OSS communities, the bazaar mode of governance does not prove to be efficient enough to account for the increased need for administrative (informal as well as formal) control mechanisms and provides less incentives for effective production compared to other modes of governance. As a result, a number of mixed forms of bazaar governance have emerged ranging from quasi-hierarchical (Linux) to (kind of) centralized (Apache)

approaches (Demil and Lecocq, 1999). For an overview about these different modes of governance of OSS communities see Lynne Markus et al (2000). As we will show below, a unique mixed approach of bazaar governance has been developed within the Debian OSS community.

3 Governance mechanisms in transition: The Debian OSS community

3.1 Characterizing the Debian OSS community

The Debian OSS community has experienced a rapid growth since its establishment in 1993 by Ian Murdock involving currently more than 900 volunteer package maintainers.

However, the Debian OSS community differs from others because the programming work within the community is not concentrated on producing code, but on integrating code into a coherent system. In this respect, Debian is more in line with Red Hat, SUSE and Mandriva than with the Linux kernel, Apache and Mozilla (Bauer and Pizka, 2003, Gonzáles-Barahona et al, 2004, Narduzzo and Rossi, 2003). Therefore, two separate code structures (trees) that are running in parallel can be identified (a stable and a more experimental version of Debian software) but vital has been the integration of both trees. The stable version of Debian has been focused on the package system (dpkg). The experimental version served as a test bed for new features of (public) releases of Debian. This focus on integration of code has also been important to understand the emerging different task structures within Debian compared to other OSS communities.

The task structure of the Debian community has been focused around a "core" which consists of the Debian project leader (DBL),

developers as well as a "periphery" of maintainers. As the "core" is responsible for the production of new code, the periphery deals with the integration of these codes for particular applications. This structure differs from other OSS communities like Linux (Lee and Cole, 2003) as it is sometimes difficult to draw a line between them.[6] Examining the specifics of the code structure used by the Debian OSS and the evolving task structure is essential to the understanding of the development of the different informal governance forms within the Debian OSS community.[7] (For information on the methodology used see Appendix 1)

3.2 Organizational Growth and the Emergence of Informal Forms of Governance

The Project Initiation Stage

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

---

[6] However, there is a spectrum between integrators and code producers rather than a clean line of separation. For instance, many Debian developers are involved in troubleshooting other projects' code, writing patches and "upstream" work. Similarly, Red Hat employs the key developer of the GNU project's C library and Novell employs key GNOME, mono developers and kernel developers specializing in particular hardware platforms.

[7] It furthermore is important to know that the Debian OSS community has not been influenced by strategies of sponsoring companies. Other OSS communities are (still) operating in other market segments (like Ubuntu in the desktop market and in the individual user segment) or specific markets (like Mandriva) and do not

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 OSS 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 et al, 2003), the motivations of the initiators to start up a new project have just recently received some attention in the literature. The second component in the initial stage is related to the 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). These three components can also be found in the initial phase of the Debian OSS community.

The Debian project was started by Ian Murdock from scratch after being dissatisfied with the SLS (Softlanding Linux System) release. Ian Murdock wanted to "draw a few people out the

---

have (yet) an extensive support organization as Debian (or Red Hat or Novell) already provide.

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-organized community of developers and users (Varghese, 2003). The foundation for the parallel code structure were already laid down during this period leading to the (public) releases of Debian and the (rudimentary) package system called dpkg.

The "Going Open" Stage

In order to enter the going open stage, OSS communities face certain challenges such as achieving project and product credibility, developing adequate communication mechanisms, creating effective recruitment strategies as well as the development of appropriate forms of governance. 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 from "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.

When "going open" similar challenges were facing the Debian OSS community when Ian Murdock felt that Debian software was ready to be shared. He made the official announcement on the Internet, and encouraged others to help him to improve it. On September 2nd 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 Summer 1994, early versions of dpkg were becoming usable, and other people besides Ian Murdock 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 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 Fall and Winter 1994. These development releases

had the function to experiment and to further improve on public releases as they were used as a learning device. To account for this experimental tree of development and to include new innovative opportunities, the Debian OSS community has developed later a whole cycle of releases ranging from an 'unstable' over a 'testing' to a 'stable' package. Table 1 provides an overview of Debian releases and major events during this second phase.

Insert Table 1 about here

As can be seen in Table 1, since 1995 the steady growth in the number of packages has been accompanied by an increase in the number of developers in the Debian community. By this time, the Debian Project, as it became known, had grown to include over sixty people. In the summer of 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 1995 to come up with 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 release (called "Buzz" or Debian 1.1) in June 1996. During his leadership period, the Debian Social Contract was ratified by the Debian developers in 1997 which included the Debian Free Software Guidelines (DFSG) and provided the Open Source Definition for the Debian community. As the DFSG provided guidelines on what constitutes free software in the Debian context, new members had to agree with the Debian Social Contract and the DSFG in order to join the Debian OSS community.

The successor of Bruce Perens, Ian Jackson, the first elected Debian project leader (DPL), had major influence on formalizing activities within the growing Debian community that lead to the Debian Constitution which was in 1998 approved by a voting procedure. As shown in Figure 1, the Debian Constitution was a first attempt to define different roles (e.g. the DPL, the Technical Committee, and Developers) in a form of hierarchy within the Debian community (Garzarelli and Galoppini, 2003). The role of the coordinator was assumed by the DPL. He helped to define the project's vision, lent authority to Developers and made any decision that requires urgent action. The Leader also represented Debian the Project to the outside world (e.g., by attending conferences and gives talks). All Debian Developers could vote to elect the Project Leader. Still, the developers, which are at the bottom of this

hierarchy, could override any decision taken by the Project Leader or the Technical Committee. Furthermore, the Constitution did not impose any obligation on anyone to work continuously on the Debian project; in fact, a contributor could leave the project at any time or resign from his or her position or duty by a simple announcement.

Insert Figure 1 about here

During the period 1996 and 1999 there were three more stable releases, which were provided by Debian developers and maintainers. Within the Debian community, a task structure had developed in which certain developers (including the DPL) contributed to new releases even if they were sometimes not directly linked to a particular package and maintainers that were taking an existing open software packages and create a ready-to-install Debian package (Robles et al, 2005). 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 hold on accepting new maintainers. A crisis occurred when the Debian community no longer felt that it could adequately protect its boundaries and closed its doors to new potential members. As the

acting DPL Wichert 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 more than 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 or maintainer. It comprises a registration process of New Maintainers (NM) that is handled by the NM-Committee, which is a body of people who control the New Maintainer process. It is composed of the Front Desk, the Application Managers, and the Developer Accounts Managers. The Front Desk officers receive new application requests and pass them to appropriate Application Managers. The Application Manager is a Debian developer who is assigned to an Applicant in order to monitor their progress through the application process. Developer Accounts Managers (DAMs) manage user accounts on Debian machines, and finalize the details of membership by assigning

accounts to new developers. The DAMs are delegates appointed by the DPL (see Figure 1).

The new maintainer approach has been a way of keeping Debian open, but at the same time, a way to manage its boundaries. It defined a new governance structure by providing a mechanism for managing membership that allowed to evaluate whether (or not) new member's skills, goals, and ideology were in line with that of the community (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.

The Growth Stage

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 communication. By providing for reciprocity and reputation, communication processes were streamlined and coordinated by using, in particular, the various Debian mailing system. The Debian mailing system evolved over the years by continuously including new specific topics lists such as Users, Developers, Internationalization and Translations, Ports, Miscellaneous Debian, Linux Standard Base and Software in the Public Interest. These lists were coordinated by the mailing list maintainer. As one participant described it: "The language on the list is very high tech programming language, a work-do-not–chat-mentality. Many people work behind the scenes and you do not often see them at the mailinglists. However, when they are there, they speak with great authority." Within the Debian project mailing lists fulfilled three different functions (Lanzara and Morner, 2003): First, as virtual construction sites they were used to continuously create, update, modify and repair software constructs; second, as some sort of electronic crossroad they were used to exchange information and problems as well as discuss solutions, and third, as a form of weblog they recorded the history of the Debian OSS community. The mailinglists allowed unrestricted access to

discussions, allowed knowledge circulation and have been a means to structure the communication within the Debian community. At the same time they allowed dissemination activities of the Debian project to take place quasi-automatically, because documentation of built software products or solutions can circulate throughout the web almost instantaneously. The dissemination process has been linked to the development activity, and has been embedded in the Internet-based information and communication structure. As a result of these new functions, mailinglists were considered as a new mechanism of governance within the Debian OSS community (Lanzara and Morner 2003: 37).

A continuous problem of management of the Debian OSS community has been 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 has been stable. Within the Debian OSS community, the Debian project leaders developed their own leadership style to deal to problems of slow release management and for the growth of the community as a whole. 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. At the beginning when there were only a few people involved in the Debian project, strong leadership was accepted. However, other styles of leadership

were used by new Debian project leaders to deal with increasing structural complexity of the Debian community.[8] This was the point in 1996 when leadership elections were arranged by the project leader secretary. 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. Ian Jackson tried, together with the community to "fit the governance structure" to the size of the community and to the feelings of freedom that

---

[8] In discussing the leaderships qualities of former project leaders (Ian Murdock, Ian Jackson), Wichert Akkerman characterized new challenges emerging from the differentiated task structure in his leadership speech as follows "I do not intend to be as dictating and vocal as Bruce was, but neither as silent as Ian was the last year. Both have done a good job, but things are not what they were. Debian has grown to be too big for Bruce's style of leadership, and Ian has laid a great foundation for a new period by giving us the constitution. This also means the role of project leader is now very different: most functions have been delegated, leaving the leader to act as a kind of benevolent overseeing person who nudges the project in a good direction."

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 was 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.

The call for more team-based leadership 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 continuous development. Within the Debian community, criticism increasingly mounted about the slow release management cycle of the project. Within the leadership elections,[9] 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, the 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 Brandon Robinson and Andreas Schuldei suggested forming a small formal team of Debian developers aimed at supporting the project leader. This team, nicknamed "Project Scud",[10] was organized in the last few weeks of 2004. Brandon Robinson, who became in 2005 the new DPL proposed "a new approach to Debian Project leadership" in which he, Jeroen van Wolffelaar, Andreas Schuldei, Enrico Zini,

---

[9] During the 2005 elections, candidates with own platform were M. Garrett, A. Schuldei, A. Lees, A. Towns, J. Walther and B. Robinson.

Steve Langasek, and Ben Garbee, formed the 'Project Scud', i.e. "a team of concerned Debian Developers who have resolved to take some new approaches to resolve long-standing problems within the project". According to Scud members having a 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 DPL. While being part of the DPL team it was possible to micro-delegate tasks to the most appropriate person.

The Scud team identified small teams (up to seven people) as probably the single most important unit for the Debian project to grow in a healthy way. If the team would function well it could solve more problems than individual developers. 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 proposed by the Scud team was that people could form a knowledge pool when cooperating on package maintenance, infrastructural or organizational tasks, and it was less likely that

---

[10] The name Scud was meant to be an internal code-name inspired by the dog named Scud in "Toystory". After the elections the team was operating under the name "DPL team", however Debian members referred to it as "Scud".

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. As these teams could grow and divide, they were considered as self-organizing and would provide for very good scalability in numerical growth.[11]

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 DPL. The Scud proposal has been a source of some concern, especially how it would integrate within the Debian constitution and the existing organizational structure.[12] The discussion on the mailinglists shows that members of the Debian community got confused by the DPL team idea. They argued that the DPL can always delegate tasks to other members of the project and therefore the argument of Scud

---

[11] An example of team-based work being organized in the Debian project was provided by Andreas Schuldei who argued 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.

[12] Some 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 defined 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: First, 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*); second, 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).

members that it is impossible for a single DPL to have time to do everything is not valid.[13] 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 issue that could not be discussed within everyone. Debian members felt offended by the idea of private meetings between Scud members.[14] Further question marks have been 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 formalize the Scud team.

With the upcoming leadership elections it has been time for reflections about the working of the DPL team. Jeroen van Wollfelaar, now one of the running candidates, explains that during this whole year the Debian community was divided on the issue of the Scud team. In general, the community kept a somehow wait-and-see attitude. To his disappointment the DPL team did not work as expected.[15] However, currently it has not been clear whether or

---

[13] "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?"

[14] 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.

[15] "First, because the team had no official status and the chosen DPL did not give the team the priority it deserved. Robinson liked the idea, but was not an

not the Scud team will be established as something permanent within Debian's governance structure.

---

enthusiastic proponent of the team approach. He lacked the leadership skills to lead the team in an effective manner. There have been Scud meetings, and to a certain extent they were useful, but it was not so that the Scud fulfilled DPL functions. These functions still were carried out by the project leader himself."

Summary and Discussion

OSS communities evolve through several different phases; i.e. introduction, growth, maturity or decline. The "going open" stage has generally been considered as critical to OSS communities in deciding whether or not these communities will face further growth, maturity or decline. To facilitate the adaptation of OSS community during these different growth stages, a wide variety of differentiated task structures with different degrees of formalized technical as well as administrative structures have emerged. As the evolution of different task structures has been rooted in heterogeneous processes of learning, the formalization of the technical and administrative structures has been driven by the needs within the OSS community to explore and exploit knowledge. The evolution of different governance forms has therefore to be considered in the context of these task structures as well as technical and administrative structures.

In exploring the different stages in the development of OSS communities, the paper has linked the evolution of different informal governance forms within the Debian OSS community to the particular parallel code structure utilized and the task structure within this community. Even if separate code structures running in parallel can be identified within the Debian OSS community (i.e. a

33

stable and a more experimental version of Debian software), the integration of both structures has proved to be vital.

The task structure of the Debian community differs from other OSS communities like Linux (Lee and Cole, 2003) as the boundaries between core and periphery have been more difficult to trace. Even if the distinction between "core" around the Debian project leader and developers as well as a "periphery" of maintainers can be made. The specifics of the code structure used by the Debian OSS and the evolving task structure has provided an understanding of the development of the different informal governance forms within the Debian OSS community.

The emergence of an elected leader in conjunction with a project leadership team provides new evidence for the need to search for novel and alternative forms of governance of OSS communities. In the face of growing structural and technical complexity, they provide a solution to the dilemma of OSS communities during the "going open" stage of their development.

References

Bonaccorsi, A. and C. Rossi. 2003 Why Open Software Can Succeed. *Research Policy*, 32: 1243-58.

Bauer, A. and M. Pizka. 2003 The Contribution of Free Software to Software Evolution. *Free Open Source Research Community 2003-09*.

Dafermos, G. 2001 Management and Virtual Decentralized Networks: The Linux Project. *First Monday*, 6(11).

Dalle, J. & N. Jullien. 2003. 'Libre' Software: Turning Fads into Institutions? *Research Policy*, 32(1): 1-11.

Demil, B. and X. Lecocq. 1999 Neither Market Nor Hierarchy or Network: The Emerging Bazaar Governance *Research in Organizational Behavior*, 12: 295-336.

Denzin, N. and Y. Lincoln. 1998 *The Landscape of Qualitative Research*. Thousands Oaks, CA: Sage Publishing.

DiBona, C., S. Ockman, and M. Stone. 1999 Introduction. In Dibona, C., S. Ockman, and M. Stone, editors, *Open Sources. Voices from the Open Source Revolution*: O'Reilly Media Inc.

Dibona, C., S. Ockman, and M. Stone. 1999 *Open Sources: Voices from the Open Source Revolution*. Sebastopol, CA: O'Reilly and Associates.

Fielding, R. 1999 Shared Leadership in the Apache Project. *Communications of the ACM*, 42(4): 42-43.

Franke, N. and E. von Hippel. 2003 Satisfying Heterogeneous User Needs Via Innovation Toolkits: The Case of Apache Security Software. *Research Policy*, 32(7): 1199-215.

Garzarelli, G. and R. Galoppini. 2003 Capability Coordination in Modular Organization: Voluntary FSS/OSS Production and the Case of Debian GNU/Linux. *Free Open Source Research Community*, 2003-11.

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

Gonzáles-Barahona, J., L. López, and G. Robles. 2004 Community Structure of Modules in the Apache Project: Open Source Research Community.

González-Barahona, J. and M. Ortuño Perez. 2002 Counting Potatoes: The Size of Debian 2.2. *Free Open Source Research Community 2002-03*.

Johnson, J. 2006. Collaboration, Peer Review and Open Source Software. *Information Economics and Policy*, 18: 477-97.

Hertel, G., S. Niedner, and S. Herrmann. 2003 Motivation of Software Developers in Open Source Projects: An Internet-Based Survey of Contributors to the Linux Kernel. *Research Policy*, 32: 1159-77.

Kimberly, J. and R. Miles.1980. *The Organizational Life Cycle*. San Francisco: Jossey Bass.

Kogut, B. 2000 The Network as Knowledge: Generative Rules and the Emergence of Structure. *Strategic Management Journal*, 21: 405-25.

Kraut, R. and L. Streeter. 1995 Coordination in Software Development. *Communications of the ACM*, 38(3): 69-81.

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

Lanzara, G. and M. Morner. 2003 The Knowledge Ecology of Open-Source Software Projects. *Proceedings of the 19the EGOS Colloquium, Copenhagen, July 3-5*.

Lee, G. and R. Cole. 2003 From a Firm-Based to a Community-Based Model of Knowledge Creation. *Organization Science*, 14(6): 633-48.

Lynne Markus, M., B. Manville, and C. Agres. 2000 What Makes a Virtual Organization Work? *Sloan Management Review*, 42(1): 13-26.

March, J. 1991. Exploration and Exploitation in Organizational Learning. *Organization Science*, 2(1): 71-87.

Marshall, C. and G. Rossman. 1999 *Designing Qualitative Research*. Thousand Oaks, CA: Sage Publishing.

Michlmayr, M. 2004 Managing Volunteer Activity in Free Software Projects. Paper presented at 2004 USENIX Annual Technical Conference, Boston MA.

Moon, J. and L. Sproull. 2000 Essence of Distributed Work: The case of the Linux Kernel. *First Monday*, 5(11).

Narduzzo, A. and A. Rossi. 2003 Modularity in Action: GNU/Linux and Free/Open Source Software Development Model Unleashed. *Free Open Source Research Community 2003-5*.

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

Ouchi, W. 1979 A Conceptual Framework for the Design of Organizational Control Mechanisms. *Management Science*, 25(9): 833-48.

Powell, W. 1990 Neither Market Nor Hierarchy: Network Forms of Organization. *Research in Organizational Behavior*, 12: 295-336.

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

Raymond, E. 2001 *The Cathedral and the Bazaar* Sebastopol, CA: O'Reilly and Associates.

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

Robles, G., J. Gonzáles-Barahona, and M. Michlmayr. 2005 Evolution of Volunteer Participation in Libre Software Projects: Evidence from Debian. *Proceedings of the 1st International Conference on Open Software Systems, 2005 Graz, Austria*.

Ronneburg, F. 2006 *Debian GNU/Linux Anwenderhandbuch*: Creative Commons Namensnennung.

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).

Varghese, S. 2003 Living Up to the Linux Name. *The Age*.

von Hippel, E. and G. von Krogh. 2003 Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science. *Organization Science*, 14(2): 209-23.

Wall, L. 1999 The Origin of the Camel Lot in the Breakdown of the Bilingual Unix. *Communications of the ACM*, 42(4): 40-41.

Williamson, O. 1985 *The Economic Institutions of Capitalism*. New York: The Free Press.

Williamson, O. 1975 *Markets and Hierarchies: Analysis and Antitrust Implications*. New York: The Free Press.

Table 1:   New releases and important events in the Debian History (1993 – March 2006)

Timeline	Release	Package System dpkg	Packages	Developers	Events
Fall-Winter 1993	Several Internal Releases				Founder Ian Murdock
January 1994	Public Release of Debian 0.91.	Rudimentary dpkg		Small	Ian Murdock still largely packaging and integrating the releases himself Rudimentary packing system used for manipulating packages
Summer 1994		Usable early versions of dpkg			With early versions of dpkg and guidelines explaining how to construct packages other people besides Ian Murdock join packaging and integration.
Fall 1994		Responsibility over dpkg (I. Jackson)			Responsibility of the package system is transferred to Ian Jackson
1995	First distributed release (Debian 0.93 Release 5)		250	60	It now is called The Debian Project.
Summer 1995	Responsibility over base system (Perens				Ian Murdock transfers responsibility of base system (core set of Debian packages) to Bruce Perens, he still is responsible for Debian management.
March 1996					Ian Murdock leaves the Debian Project in March 1996; Bruce Perens assumes leadership role.
June 1996	1.1 (Buzz)		474	90	
End 1996	1.2 (Rex)		848	120	
1997	1.3 (Bo)		974	200	Debian Social Contract including Debian Free Software Guidelines (DFSG) and Open Source Definition
1998	2.0 (Hamm)		1500	400	Debian Constitution ratified by vote (constitution includes election methods, leadership debate), first elected leader Ian Jackson
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)		10869	> 9000	Leadership elections within a new format, Discussion about a Debian Project Leader (DPL) team
no release date yet	(Etch)				

Source: (Lameter, 2002) and own information

Table 2: Informal hierarchical forms of governance within the Debian community (1993 – 2006)

Phase in Debian history	Year	Project Leader	Leadership Characteristics*	Informal hierarchical forms of governance		
				Authority	Principles	Control
Initiation and Going open stage	1993 – March 1996	Ian Murdock	"Visionary"	Founder	Open community	
Growth stage	April 1996 – December 1997	Bruce Perens	"Strong leader"	Nominated Leader	"Debian Social Contract"	
	January 1998 – December 1998	Ian Jackson	"Formal style and strategic vision"	First project leader elected, Jackson only candidate	"Debian Constitution"	
"Growth Crisis"	1999 – 2001	Wichert Akkerman	"Relaxed informal style"	Elected twice	Leadership debate and speech of opponents	"New Maintainer Process"
	April 2001 – April 2002	Ben Collins	"More visibility" as a leader	Elected		
	April 2002 – 2003	Bdale Garbee	"Networker and Facilitator", Spokesman for Debian	Elected		
	2003 – 2004	Martin Michlmayr	"Motivator and internal coordinator"	Elected		
	2005	Brandon Robinson	"Coordinator"	Elected, Discussion about a Debian Project Leader (DPL) team	Leadership elections in new format	

(* quotes refer to leadership characteristics used to describe these leaders in leadership speeches or interviews with participants)
Source: Based on own information.

Figure 1: The Debian Constitution
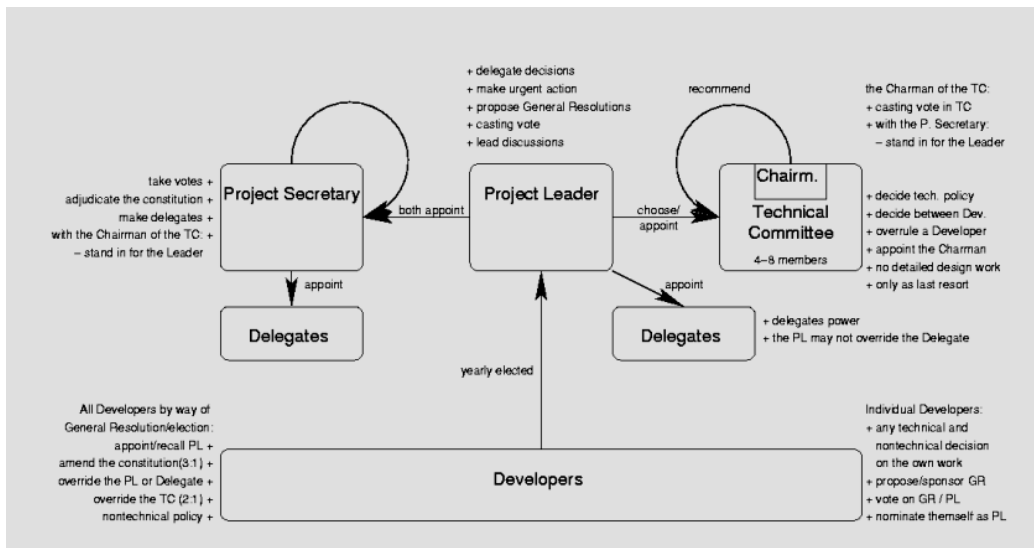
Source: (Ronneburg, 2006)

Appendix 1: Methodology

For our analysis, the 'community' phenomenon was central to the analysis of the history of the Debian OSS community. We concluded that this perspective better explains the organizational changes during the growth of the Debian community compared to other explanations found in the literature on OSS communities.[3] We closely followed the development of other OSS communities (such as Apache, Linux or Pearl) and other OSS communities developing packaged software distributions (Red Hat, SuSe). Our aim was not only to better understand the specifics of open software programming and distribution (e.g. Kraut and Streeter (1995)) but also to characterize general (as well as specific) factors driving the growth of OSS communities. For this purpose, we extensively examined websites of these OSS communities and subscribed to different mailinglists such as floss or the linux kernel.

In order to characterize governance mechanisms during transition of OSS communities, we examined the history of the Debian OSS community based on data triangulation. As this method involves the use of different sources of data/information (Denzin and Lincoln, 1998, Marshall and Rossman, 1999), it allows to characterize the different perspectives of stakeholders within the Debian community like Debian project leaders, maintainers or

developers. It also enabled us to get an understanding of specifics of the Debian community compared to other OSS communities.

To examine, in more detail, the development of the Debian OSS community, a wide variety of data sources were consulted: Primarily we used internal documents related to the content and context of different Debian projects. We complemented the analysis with semi-structured interviews (both face-to-face and by telephone) with key individuals (DPL leaders, maintainer, developers) during the period 2002 - 2005.

Similar to Dafermos (2001), we used semi-structured interviews as they provide more detailed information of greater value than straightforward question and answer sessions, especially when the research is explorative (Dafermos, 2001). These semi-structured interviews were also useful in engaging in a continuous conversation with the interviewees. The face-to-face interviews were taped and transcribed verbatim. As a check, the interviews were sent to the interviewees for comments. The interviews that were undertaken by telephone were written down as accurately as possible. Again, the transcripts were sent to interviewees in order to check their accurateness. The enormous willingness of participants to contribute to this research, e.g. by interviews and e-mail interaction has been remarkable in particular in the Netherlands. Debian developers were very supportive and helpful and always willing to travel to participate in interviews. Even developers from

other places in the world said that they would help, however as one of them remarked: "Of course, I'm willing to contribute, but when I detect 'cluelessness' from the side of the researcher, I will invest my time in something else."

Furthermore, we attended several Debian conferences and were "lurking around" on the Debian mailinglists, websites, IRC channels, etc. We identified the Debian-devel(opment) mailinglist, as it is the most important (the "head" mailinglist) of the project, and we analyzed a few threads of messages on the Debian-devel mailinglist. Interviews were used to gain further insights into the Debian community. In addition, articles on Slashdot.org, members' biographical writings and diaries, previous interviews with key members and descriptions of the community written by other researchers and key people were extensively utilized.   After having established initial contacts, a kind of network of participants developed. Members of the community pointed out: "You could ask this member about that," or, "I know someone who can help you with that." In that way we were introduced to most interviewees and important contributors to the Debian project. Several pages on the Debian homepage also pointed out key people in the Debian project. Based on this approach, we met diverse programmers, from the inner circle to newcomers on the project, which made the range of responses quite broad. In addition, we posted an overview of this case study on one of the Debian mailinglists and asked

people for comments; this also brought us in touch with members of the community. A draft of the case was send to Debian members, who provided additional (and valuable) comments. As a result we were able to follow the Debian project in great detail with respect to its history as well as its ongoing development and activities.

This methodology enabled us to characterize the growth of the Debian OSS community as a process in which not only a differentiated role structures emerged that both reflected and supported its activities but different forms of governance were implemented.

The UNU-MERIT WORKING Paper Series

2007-01 *Developing science, technology and innovation indicators: what we can learn from the past* by Christopher Freeman & Luc Soete

2007-02 *The impact of innovation activities on productivity and firm growth: evidence from Brazil* by Micheline Goedhuys

2007-03 *Estimations of US debt dynamics: Growth cum debt and the savings glut in Kouri's model* by Thomas Ziesemer

2007-04 *States and Firms on the Periphery: The Challenges of a Globalising World* by Gabriel R.G. Benito & Rajneesh Narula

2007-05 *How Do Consumers Make Choices? A Summary of Evidence from Marketing and Psychology* by Zakaria Babutsidze

2007-06 *Inter-firm Technology Transfer: Partnership-embedded Licensing or Standard Licensing Agreements?* by John Hagedoorn, Stefanie Lorenz-Orlean & Hans

Kranenburg

2007-07 *The Behavior of the Maximum Likelihood Estimator of Dynamic Panel Data Sample Selection Models* by Wladimir Raymond, Pierre Mohnen, Franz Palm

& Sybrand Schim van der Loeff

2007-08 *Location and R&D alliances in the European ICT industry* by Rajneesh Narula & Grazia D. Santangelo

2007-09 *How do social capital and government support affect innovation and growth? Evidence from the EU regional support programmes* by Semih Akcomak & Bas ter Weel

2007-10 *The Micro-Dynamics of Catch Up in Indonesian Paper Manufacturing: An International Comparison of Plant-Level Performance* by Michiel van Dijk & Adam Szirmai

2007-11 *Financial Constraint and R&D Investment: Evidence from CIS* by Amaresh K Tiwari, Pierre Mohnen, Franz C. Palm & Sybrand Schim van der Loeff

2007-12 *The Spatial Hierarchy of Technological Change and Economic Development in Europe* by Bart Verspagen

2007-13 *The origins and implications of using innovation systems perspectives in the design and implementation of agricultural research projects: Some personal observations* by Andy Hall

2007-14 *Technology supply chain or innovation capacity?: Contrasting experiences of promoting mall scale irrigation technology in South Asia* by Andy Hall, Norman Clark and Guru Naik

2007-15 *Are firms that received R&D subsidies more innovative?* by Charles Bérubé & Pierre Mohnen

2007-16 *Foreign direct investment and firm level productivity. A panel data analysis* by Geoffrey Gachino

2007-17 *Technological spillovers from multinational presence towards a conceptual framework* by Geoffrey Gachino

2007-18   *Technological capability building through networking strategies within high tech industries* by Wim Vanhaverbeke, Bonnie Beerkens and Geert Duysters

2007-19   *External technology sourcing: the effect of uncertainty on governance mode choice* by Vareska van de Vrande, Wim Vanhaverbeke & Geert Duysters

2007-20   *Exploration and Exploitation in Technology-based Alliance Networks* by Wim Vanhaverbeke, Victor Gilsing, Bonnie Beerkens, Geert Duysters

2007-21   *ICT Externalities: Evidence from cross country data* by Huub Meijers

2007-22   *Knowledge Flows, Patent Citations and the Impact of Science on Technology* by Önder Nomaler & Bart Verspagen

2007-23   *R&D offshoring and technology learning in emerging economies: Firm-level evidence from the ICT industry* by Zhe Qu, Can Huang, Mingqian Zhang & Yanyun Zhao

2007-24   *The Environmental Porter Hypothesis: Theory, Evidence and a Model of Timing of Adoption* by Ben Kriechel & Thomas Ziesemer

2007-25   *Measuring the Effectiveness of R&D tax credits in the Netherlands* by Boris Lokshin & Pierre Mohnen

2007-26   *The productivity effects of internal and external R&D: Evidence from a dynamic panel data model* by Boris Lokshin, René Belderbos & Martin Carree

2007-27   *National System of Innovations and the role of demand. A cross country comparison* by M. Abraham Garcia-Torre

2007-28   *The Global Challenges of the Knowledge Economy: China and the EU by* Can Huang and Luc Soete

2007-29   *Redefining Foreign Direct Investment Policy: A Two Dimensional Framework* by Sergey Filippov & Ionara Costa

2007-30   *Redefining the Nexus between Foreign Direct Investment, Industrial and Innovation Policies* by Ionara Costa & Sergey Filippov

2007-31   *Innovation and Competitive Capacity in Bangladesh's Pharmaceutical Sector* by Padmashree Gehl Sampath

2007-32   *R&D collaboration networks in the European Framework Programmes: Data processing, network construction and selected results* by Thomas Roediger-Schluga & Michael J. Barber

2007-33   *Determinants of alliance portfolio complexity and its effect on innovative performance of companies* by Geert Duysters and Boris Lokshin

2007-34   *Strategic Partnering with Chinese companies: Hidden motives and treasures* by Geert Duysters, Tina Saebi & Dong Qinqin

2007-35   *Approach for analysing capabilities in latecomer software companies* by Rossitza Rousseva

2007-36   *Foreign-owned firms and technological capabilities in the Argentinean manufacturing industry* by Ionara Costa & Anabel Marin

2007-37   *Short-term effects of new universities on regional innovation* by Robin Cowan and Natalia Zinovyeva

2007-38   *Challenges to Strengthening Agricultural Innovation Systems: Where Do We Go From Here?* By Andy Hall

2007-39   *Transition of Governance in a Mature Open Software Source Community:*
          Evidence from the Debian Case by Bert M. Sadowski, Gaby Sadowski-Rasters
          and Geert Duysters