

Open Source Community Organization

Onkgopotse MOLEFE, Thomas FOGWILL
Meraka Institute of the CSIR, PO Box 395, Pretoria, 0001, South Africa
Tel: +27 12 8412076, Fax: +27 12 8414570,
Email: omelefe@csir.co.za, thomas.fogwill@meraka.org.za

Abstract: Open Source communities (OSCs), sometimes referred to as virtual or online communities, play a significant role in terms of the contribution they continue to make in producing user-friendly Open Source Software (OSS) solutions. Many projects have grown rapidly due to the input provided by members of the Open Source community, who, some as volunteers and some professionally, have dedicated their time and effort to developing and improving open source software code. This paper looks at the ways in which open source communities are created, governed and maintained, and grown.

Keywords: Open Source Software, Open Source community, Community governance.

1. Introduction

Open Source communities are the focus of this study. In particular, a framework for discussing open source communities is presented and is then used to do conduct a case study analysis of several existing and successful open source communities. The aim is to provide insight into how successful open source communities develop, are governed, grown and maintained. This study will cover three important questions (1) how to establish an open source community; (2) how to govern such a community; and (3) how to grow and maintain it. All the questions will be dealt with in detail in section 6.

The paper begins with definitions of terms in section 1.1, followed by a description of the purpose of the study in section 2 and the methodology and approaches that were used for data collection in sections 3 and 4, respectively. Some background on the study is provided in section 5 and section 6 presents a framework on how communities are organised, established, governed, maintained and grown. A comparison and analysis of some cases done using this framework are presented in section 7, while the paper is concluded in section 8, which highlights the important findings from this research.

1.1 Definition of Terms

Open Source Software (OSS) is sometimes abbreviated as Free and Open Source Software (FOSS or F/OSS), Free and Open Source Software and Open Content (FOSS/OC), Free/Libre Open Source Software (FLOSS) or Open Source (OS). OSS is software that is licensed in a manner that promotes collaboration and the sharing of ideas and concepts [1].

Open source community is defined by Munro et al as consisting of those who participate in developing and improving Open Source software [2], and by Dibben as a group of people, usually like minded-people, who come together on-line to participate, debate and share information. This community uses a range of on-line tools from discussion forums, chat, discussion email lists, and Information Management Systems (WebCT) [3]. Despite the importance and ad-hoc nature of this on-line community, it is important to

remember that open projects are more than anarchical communities joined only by ideologically-oriented individuals writing code in their spare time on a voluntary basis, as explained by Bonacorsi et al [4].

For the purpose of this study, an open source community is defined as an informal term that describes a number of participants (such as users and developers) with a shared common interest, collaborating on-line using a wide range of communication tools (e.g. mailing lists and IRC) to develop and produce a particular piece of software (voluntarily in their spare time, or in their professional capacity).

These people often do not expect direct monetary compensation, but participate as a hobby or passion, to develop software that will satisfy existing computing needs.

2. Objectives

Little is published on how to go about building a successful Open Source community [5][6]. The purpose of this study is therefore:

1. To provide a clear description of the steps that community team leader or founders should follow when planning to enable them to build an effective and vibrant Open Source community. This simple set of steps will cover how to go about establishing an Open Source community, how to grow it, how it should be governed, how it can be coordinated, and how to maintain it.
2. To analyse and compare some examples of existing and successful Open Source communities, to determine what made them successful.

By addressing these 2 points, this study provides guidelines to open source project leaders to help them develop and grow healthy communities around their projects.

3. Methodology

This paper is based on a comparative case study analysis of a number of successful Open Source communities, including those of Scilab, Debian, Apache and Ubuntu. It tests the manner in which these communities were created, grown, governed and maintained.

It is therefore a qualitative study that involves the collection, study, interpretation and drawing of conclusions from qualitative data. Typical qualitative data “take the form of transcribed interviews, documents, data obtained through observations, field notes and other (typically) non-numerical data” [7]. A qualitative approach was chosen for this study to enable researchers to make sense of a situation and gain a much richer understanding of the process involved when planning and forming an Open Source community, via the analysis of people’s spoken words and writings.

4. Data Gathering Instruments

The specific data gathering methods that was used for this study include: A review of the current literature on the establishment of Open Source communities; Document analysis, including the comparison and analysis of industry documents and organisational guides associated with OSS projects and OS communities; Review, comparison and analysis of several published case studies, to identify possible patterns.

5. Background

There is a lack of published guidelines on how to build new, vibrant and successful Open Source communities [6]. Similarly, little attention has been paid to the governance structures of these communities [8].

Open Source communities are needed to benefit individuals, teams, projects, and companies. This is recognised by Foster, who suggests that Open Source communities are needed for the following reasons [9]:

1. To engage people: communities can engage users/customers and other people who may be interested in the project's (or vendor's) products and services. A community of users provides a means of users to come together virtually in a central place where they can engage on-line and bring up different ideas on how the product can be further improved to meet their needs.
2. For product innovation: the community provides feedback about your product. Having a community allows developers/vendors to ask questions about how people use their software, how satisfied they are, and what the future requirements are for the software.
3. To recognize those that are loyal to your brand: creating a community creates opportunities for recognising and rewarding those users who are fanatically loyal to your project, and who typically act as evangelists.

The next section fully describes the structure or framework that communities can use to organize themselves. It also describes a potential way to govern and maintain an effective Open Source communities

6. Framework for Organizing Open Source Communities (OSCs)

A community framework can be defined as a life cycle consisting of stages or phases that a community has to pass through to test whether it will be successful or not. The type of a framework developed here was inspired by Christoph and Stefan [8] who developed the following life cycle of Open Source communities: (1) introduction stage of a community, (2) growth stage, (3) maturity stage (4) decline stage or revival stage . First, different ways of forming and growing a community will be discussed, followed by a discussion on ways of governing and maintaining a community.

6.1 Establishing an OSC

There are several ways of starting a community, including:

1. Building a completely new one from scratch, which occurs for new projects or when organisations create new projects to attract new members to join them.
2. Creating a new community by splintering or forking an exist project, resulting in the community also splitting/forking.
3. Turning a commercial product into an OSS product, attracting members who were previously using the proprietary license into using your project.

Whitehouse [5] believes open source communities are examples of communities that 'just happen' and operate on intuition and unwritten social rules – they get formed instantly with minimum proper planning. However, developing an effective on-line OS community takes a considerable amount of time, energy and resources [3]. This is supported by Sadowski et al, who state that while communities emerge based on 'informal and self-organizing' mechanisms, they do 'benefit from cultivation' [10].

Open source projects get started for a number of reasons. The most common one, as described by Godfrey and Tu, is when people realize an unfulfilled computing need/requirement (or an unsolved problem) and decide to take solve it [11]. Here the "itching problem", described by Raymond, comes into play: "every good work of software starts by scratching a developer's personal itch" [12].

Most Open source projects are started as a hobby with the aim to satisfy a certain need or personal passion supported by a specific capability to do the job. For instance, in the case of the Mozilla foundation, individual volunteers usually began their participation in the project as a way to "scratch their own itch," for example by reporting or (in some cases) fixing bugs that affect them, or even developing new code or new extensions to perform some function important to them [13].

This same drive could explain how communities are formed and grown. As with software projects, people get involved in open source communities because of a need to scratch an itch or meet specific computing needs. This could be in their capacity as users of the software, who join the community because they need the functionality offered by the software or need support from other community members, or it could be as developers, who join the project to improve the software for their own purposes.

An important factor of any successful community is good communication, which is made possible by a wide range of communication tools including e-mail/ mailing lists, on-line discussion forums, personalised project homepages, blogs, internet relay chat (IRC), etc. This communication makes teamwork and project coordination possible for all members of the community worldwide [8] [3].

Early in the project, the team leader or founder has to take a step back and clearly think of how to make the community work, especially a new community created from scratch.

The next logical step is to ensure that your community fits well within the goals and objectives agreed up front by the team. If it does fit well, then you create the communication and development infrastructure the community will need to collaborate, create informative guidelines new members, and publicise the creation of the project.

It is important to ensure appropriate resources (people, finances, software tools) are in place to contribute towards maintaining the project & community on a long term basis [15].

6.2 *Growing an OSC*

People participate in open source projects and communities for many reasons, including: Personal interest by hobbyists (people who pursue an activity in their spare time for pleasure) and professionals; Non-monetary extrinsic motivations, like reputation, group identification processes, and career plans [8]; Meeting new software, programming and other requirements (e.g. creating “add-on” tools, testing, fixing software problems, posting bug reports, writing documentation, reviewing software, etc); Collaborative alliances (a shared desire to effect change, a shared understanding of a problem, high level of interdependence, or a desire to gain a competitive advantage); Intellectual interest in the software; Association with some other organization or individual that is interested in or actively using the software [2]; Economic and technical benefits, including the ability to experiment with new software development models, to gain market share over the competition, to boost public relations by contributing to a public good, to develop a new profit opportunities (especially relevant for companies getting involved in OS communities) [14]; Desire to learn and share or exchange knowledge; Extend communication or network with like minded groups; Build self-profile or expand self-growth; Desire to be creative [5]; Freedom to showcase skills and creativity; and a Feeling of ownership.

To grow an open source community, it is important to understand what motivates people to join, to make it easier for people to get involved, and to recognise their contribution when they do.

6.3 *Governing an OSC*

Community governance is vitally important, as a strong governance structure and particularly a strong legal framework is required to help participants to feel secure [14]. A community must have its own internal structures to help it function effectively.

Typical components of an Open Source community include:

- A project leader, either elected by the members of the community to lead the project, or the person who started the project and continues to act as a “benevolent dictator”
- Community leader/champion who leads and looks after user & developer communities

- Developer community, who develop and maintain the software
- Users community, who have some problem that is solved by software
- A procedural infrastructure to provide long-term stability
- The visionary who makes the long-term decisions (often the same person as the leader)
- The tester community who find and manage bugs
- The documentors who writes the manuals
- The translators who translate the software into other languages
- The communicator and evangelist who tells other people how good it is
- The distributor who packages it

Proper governance and coordination are required for any OSC. According to Christoph and others, a considerable amount of research has been devoted to the growth and expansion of open source communities, but little attention has been paid to open source community governance structures (e.g. control, monitoring, supervision).

Sadowski et al propose introducing a variety of governance mechanisms once the community starts to flourish and becomes productive and mature. Good governance will enable you to manage member participation and coordinate the launch of new releases [10].

Different governance mechanisms are applied to different types of communities and without proper governing rules in place, any kind of a project or community, whether Open Source or not, can fail. Open source projects are radically different to standard enterprises - in open source projects much of the work is done on a voluntary basis, thus there are little guidelines regarding time and intensity of work. It is therefore likely that the ideal governance mechanisms of an OSC would differ from those of common enterprises in terms of their coordination and organisational structure. [8].

Leadership is important component when it comes to managing a project [14]. In order to govern and build a successful community, a strong and dedicated leader is required with good personality and who shows respect and humility towards members of the community, at the same time earning their trust. This individual must be a senior member of that specific community. Howison et al. (2003) found that the majority of successful open-source projects, large and small, retained a single leader for long periods of time, while Fedorowicz et al. (2006) proposed the importance of having champions from all participating organizations instead of one lead champion. This creates an environment where everyone will have a say and the decisions made will not rely on a leading champion only. Leadership is also important when it comes to creating agendas, developing goals, and ensuring that the project continues to move forward without splintering or forking. However, community leaders must also bear in mind that they are servants of the community and that good leadership is accompanied by appropriate behaviour.

A Constitution or Code of Conduct must be drafted to formalize leadership and member roles, rights and responsibilities [17]. When it comes to the decision making process, projects are normally governed and driven by volunteers. This is sometimes referred to as "do-ocracy", the power of those who do [16].

Apart from leadership, the community consists of other members, including volunteer contributors and users. These members of the Open Source community can choose their own self-assignments or development tasks and they are allowed to take on different roles depending on their interests and skills. Success of the project depends very much on the availability of a competent technical group of developers who guarantee ongoing development, quality of the product and support in a growing community [2].

6.4 Governance and Leadership Styles (Democratic or Participatory Mechanisms vs. Bureaucratic or Autocratic Mechanisms in Community Forms)

Open source communities are governed in a number of different ways. Some are democratic, where leadership is elected by the community, while others (particularly smaller projects) have a “benevolent dictator”. Some projects use merit and a track record of contribution as determinants of seniority in the project, while others have rigid hierarchical structures.

Since the operation of an Open Source community is based on the voluntary contribution of members of the team, it is not surprising that democratic and participatory mechanisms are frequently used to populate a governance structure for the team, especially in larger teams where the size of the community make the “benevolent dictator” model infeasible. As Machado states, “Democratic mechanisms enable the community’s governance system to adapt as members learn how to interpret leadership and authority in a community context. This suggests an evolving and context-dependent notion of meritocracy and that democratic mechanisms serve as an important adaptive function” [17].

A “Meritocracy” is literally a government by merit. Many Open Source communities operate as meritocracies, where the group grants special access, permissions or leadership to those they feel had "earned it", i.e. based on merit [16].

Most OSS communities are not restrictive in their access policy - they often rely on referral and reputation for growth and projects are normally governed and driven by the people who volunteer for the job.

OS projects need to clearly state their copyright policy so that contributors know the implications of their contributing to the project, especially in terms of ownership, copyright, intellectual property rights and liability [3].

6.5 Maintaining a Community

Just as developing a community takes a considerable amount of time, energy and resources, so does maintaining such a community on a long-term basis.

No community is perfect, therefore the leader of the community and all its members need to keep active and positive at all times. Things will go wrong and when they do, it is the leader's responsibility to resolve problems quickly (with the help from the rest of the team). The leader must also maintain open communication channels and deal with imperfections and issues as quickly and openly as possible [9]. Furthermore, it is the leader's responsibility to ensure that the communities constitution is upheld, and to always act in the best interest of the community.

Examples of difficulties experienced by communities include negative comments, decreasing number of developers and users, loss of interest, fewer downloads of products, spam and website defacement, community members who act disrespectfully towards others, or who do not conduct themselves in the spirit of the project and community.

6.6 Rules for a Successful OSC

A number of rules have been developed by community experts to help ensure the success of open source communities [6]. While there is no guarantee that these rules will work in all cases, they will increase the chance of a community succeeding:

1. Set the rules and policies as early as possible – they make it easier for all participants to understand their respective roles and responsibilities, and their place in the community. It also helps to define the ideal members of the community, and to think about how contribution will be rewarded/recognised.

2. Play by the community rules: the community rules should be respected and enforced at all times, else they become meaningless.
3. Don't change the rules unnecessarily: as companies/projects evolve, there is sometimes pressure for them to change how their community operates. For example, companies may decide they want to take more support business from the community. In general, this is not a good idea, unless the community itself benefits from the change of rules.
4. Encourage participation and actively engage with the community: the team should always engage the community by actively participating, by creating new content, responding to feedback, and in general being visible in the community. Experienced community members should be recognised and asked to help by providing support to new participants in any way possible.
5. Encourage new members to join: new members must always be welcomed and recognised by the team, especially as they become active in the community.
6. Make it easy for people to participate, contribute and access the community's knowledge and practices
7. Focus on topics important to the community: by doing this the project will be able to attract key contributors

7. Analysis of Case Studies

The focus in this section is on comparing and analysing case studies against the framework described in Section 6.

The case studies will be used to look at how different communities are organised, governed and maintained in the real world. These communities include those of Scilab, Debian, the Apache Software Foundation and Ubuntu.

7.1 Scilab Community

Scilab is an open source software consisting of numerical scientific software packages intended for engineering and scientific applications [18].

How it began / how was it organised or established? The Scilab community began in the 1990's as a self-organized group, which initially consisted of six researchers actively collaborating with several external developers. It was started to encourage the sharing of open source software developed for engineering and scientific entities.

Scilab was initially supported by INRIA, but with the ever-increasing number of users, INRIA decided to create the Scilab Consortium, with the support of several companies and academic organizations to ensure its sustainable future. The Scilab consortium was officially launched in May 16th 2003.

How is it governed (policies & procedures)? Scilab is governed by the CeCILL license under French law which conforms to the rules of distribution of free software [20]. At the top of the Scilab community's governance structure is a steering committee with an Executive Board and a Scientific Board. A research and development team also exists. All the members engage with everyone to make the community a success.

How is it grown and maintained (memberships)? Growing the community was challenging, but the Scilab community managed to produce and release marketable and competitive products, thus attracting new key contributors and leading to community growth.

For maintenance, every crisis experienced by the team is taken care of without delays. Bugs are fixed quickly enough and documentation with typos is also corrected by management and the rest of the team. There is no fee required to become a member of the Scilab community. Contribution and commitment are the main requirements.

7.2 *Debian Community*

Debian is a completely free and open source Linux distribution that is completely community owned and driven.

How it began / how was it organised or established? The Debian project, founded by Ian Murdock, was created as an effort to give something back to the community. Debian is today a massive open source project, with many contributors and users, and seen by many as the universal Linux distribution [19].

The Debian community offers its members a meeting point for people to further polish their skills and to interact with like-minded individuals.

People choose Debian not only because of the community, but because of its strengths: it is stable, completely free and open source, usable and has a user-friendly technical base, it allows customisations and improvement of products and has long release-cycles.

The Debian community continually aims to become fully localised in future, all the majority of Debian users will be able to use their native language (mother tongue).

How is it governed (policies & procedures)? The Debian community introduced new mechanisms of governance and informal administrative control. These include a constitution and elected leaders, as well as the use of interactive communication channels (like mailing lists or IRC channels) to help provide for community effects and feedback [10]. Since its inception the Debian project and its community have been headed up by a number of leaders with very different leadership styles to help them deal with the increasing number of new members joining the project. The Debian project has developed due to the hard work and participation made by strong leaders and community members.

How is it grown and maintained (memberships)? The Debian community relies on members participation, donations, referrals and reputation for growth. In terms of maintenance, whenever a crisis appear everyone within the team including top management keeps the focus on fixing the problem by dealing with these issues as quickly and openly as possible. The best way to become a Debian project member is to get involved with one of its many communities. Loyalty and commitment at most are of benefit to the community.

7.3 *Apache Software Foundation Community*

How it began / how was it organised or established? The Apache Software Foundation (ASF) was created in 1999 by a group of people calling themselves the "Apache Group", who came together several years earlier to start up a community that would continue to support and maintain the HTTPD web server written by the NCSA [16].

The community started as a diverse group of people that shared common interests and got to know each other by exchanging information, fixes and suggestions. As the group started to develop their own version of the software and moving away from the NCSA version, more people were attracted and started to get involved, first by sending little patches and suggestions, later by more important contributions to improve the software. This was done via the mail list. The number of contributors kept on growing.

How is it governed (policies & procedures)? The Apache Software foundation is governed based on a basic principle called "meritocracy". It also has a code of conduct called Bylaws [21].

The governance structure of the Apache Software foundation has the following entities:

- Board of Directors (board), who govern the foundation, consisting of nine individuals, elected between the members of the foundation. This board is responsible for management and oversight of the business affairs of the corporation.
- Project Management Committees (PMC), which governs the projects and they are composed of committers (every member is, by definition a committer).

How is it grown and maintained (memberships)? For growth the Apache community relies on attracting key contributors, providing support to existing member's contributions, and also on referral and reputation. The process for becoming a new member is to contribute by helping with the developments of the projects and showing commitment.

7.4 Ubuntu Community

Ubuntu is a massively popular Linux distribution that is based on Debian.

How it began / how was it organised or established? The Ubuntu community started in April 2004 when Mark Shuttleworth began to round up a small number of talented and dedicated group of open source developers to create a revolutionary new Linux desktop [22]. To interact, the community then used (and continue to do so today) the sounder mailing list and ubuntu-users mailing list as an open discussion forum for the community. In addition, they use the ubuntuforums for web forums, IRC and the web-based launchpad project collaboration platform.

From the onset the community focus of the project attracted key contributors, and the number of users and contributors increased dramatically. There were nearly 3000 messages on the ubuntu-users mailing list within the first two weeks, which led to the formation of one of the first community driven teams, the Ubuntu Documentation Team, which was founded in late 2004.

How is it governed (policies & procedures)? The Ubuntu community has several structures of community governance to help it function effectively. Those structures include different kinds of teams composed of Ubuntu team, Local Community Team (LoCo Team), Technical Board, Ubuntu Community Council, and President or Founder. Participation in every structure is open to every member of the community.

Same as the Apache Software Foundation, the Ubuntu Community is governed more on a level of meritocracy than that of democracy. They operate more on consensus than on votes to seek agreement from the people who do the work. The Ubuntu project is led by Mark Shuttleworth as president and benevolent dictator. His duty is to ask people to work on specific projects, specific feature goals, and specific bugs.

How is it grown and maintained (memberships)? For growth the Ubuntu community depends on referrals, reputation and attracting immense number of interested new members. The variety of skills provided by the participants helps to grow the community. Maintenance and arising issues are taken care of by the whole team as early as possible.

To contribute to the Ubuntu community, simply join one of the teams, which amongst others include the development team, design team, bug squad, LoCo team, documentation team, support team and testing team. Membership is open to every one at no financial cost.

8. Comparative Analysis

8.1 Community Establishment

Each of the four projects were initiated with the purpose of driving the development of open source software products. Debian and Ubuntu were started by individuals, but for different reasons: Ubuntu as a way of accelerating Linux adoption, Debian as a way of giving back to the OSS community and creating easier ways of installing Linux. Apache was started by a team of developers who merely wanted to ensure the survival of their previous project. Scilab started as a project driven within a research organisation to address a specific gap in the OSS software landscape.

8.2 Community Governance

Scilab is governed by a consortium, with representation from major collaborators and stakeholders. Ubuntu is governed by various councils and boards, whose members are approved (and sometimes voted for) by their peer. However, Mark Shuttleworth, as benevolent dictator, has ultimate deciding power. Debian is a purely democratic system, with annual leadership elections, and with a very strong constitution. Apache is governed by leaders who attain their positions of leadership based on their track record of contribution to the project.

8.3 Community Growth and Maintenance

All the projects studies strive to grow the communities, especially through word-of-mouth referral, by lowering barriers-to-entry, by treating new users and members respectfully, and by recognising contribution. However, Ubuntu stands out as being the most successful at attracting new users and contributors, due largely to the mandatory compliance of existing members to a code of conduct that requires new users to be treated courteously. In addition, Ubuntu provides excellent documentation, both on technical matters, and on community processes, and has an open and transparent governance process.

9. Conclusions

This study covered the topic of Open Source community formation, governance structures, maintenance and growth. Case studies were used as a reference to show that Open Source communities do exist and that with proper governance and coordination structures these type of communities succeed in maintaining themselves. The case studies were evaluated in the context of a proposed framework for open source community models, and compared to one another. The key observations are that the process for establishing a community are mainly similar for most projects. In each case studied, the communities were initiated for the purpose of developing software products to address users' computing needs that were not satisfied by existing software products. In terms of growth, all the communities use similar processes to attract new members or key contributors with good reputation.

For open source projects to successfully establish, grow and maintain their communities, they need to address a few key points. The community must be accessible for new contributors and users, must provide a stimulating, stable and safe environment for them to collaborate in, and must very clearly stipulate what its constitution encompasses. A strong, respectful and dedicated leader is also key to the success of the community.

Some topics were not included in this study, because they were already covered by other academic disciplines or organisational studies, however further research could include the development of a repeatable model that can be used by new Open Source communities to identify appropriate resources needed when starting up or growing. Future work could also include an investigation into how the need for transparency and openness can conflict with the need for strong governance structures, and how this impacts the size and composition of the community.

References

- [1] go-opensource.org (2005), Open Source Software, Last accessed 17/04/2007:<http://www.go-opensource.org/software/>
- [2] R. Munro, J. Negrette and A. Aitken, Engaging with the Open Source Community (Part One), (Accessed: November 18, 2008):<http://www.theinquirer.net/en/inquirer/news/2003/06/22/engaging-with-the-open-source-community-part-one>, 2003
- [3] K. Dibben, Online communities educause, (Accessed: November 21, 2008):<http://209.85.173.104/search?cache:oyvhvzof6mkJ:www.educationau.edu.au/jahia/webdav/site/myjah>

- iasite/shared/papers/online_communities_educause.ppt+making+online+communitis+work&hl=en&ct=clnk&cd=1&gl=au), 2004
- [4] A. Bonaccorsi, D. Lorenzi, M. Merito, C. Rossi, FIRMS' INVOLVEMENT IN THE PROJECTS OF THE OS COMMUNITY, PRIME 3rd Annual General Conference, 2007, pp. 1-2.
- [5] P. Whitehouse, Building an Open Source Community, (Accessed: November 12, 2008):<https://fossbazaar.org/?q=content/building-open-source-community>, August 12, 2008
- [6] C. Keene and C. Jackson, The Silverado Rules for Open Source Success, (Accessed: November 12, 2008): <http://www.keeneview.com/2008/02/silverado-rules-for-open-source-success.html>,
- [7] M. D. Myers, Qualitative Research in Information Systems, (Accessed: November 19, 2008):<http://www.qual.auckland.ac.nz/>, MIS Quarterly Discovery, June, 1997
- [8] C. Lattemann and S. Stieglitz, Framework for Governance in Open Source Communities, Proceedings of the 38th Hawaii International Conference on System Sciences, 2005
- [9] D. Foster, Maintaining-a-successful-corporate-community, Accessed: October 23, 2008):<http://fastwonderblog.com/2008/09/17/maintaining-a-successful-corporate-community/>, 16 April 2008
- [10] B. M. Sadowski, G. Sadowski-Rasters and G. Duysters, Transition of Governance in a Mature Open Software Source Community: Evidence from the Case, Working Paper Series, 2007.
- [11] Godfrey, M. and Q. Tu, Evolution of Open Source Software: A Case Study. Paper presented at ICSM-00, San Jose, CA, 2000.
- [12] Raymond, E, The Cathedral and the Bazaar. First Monday, pp. 3(3), 1998.
- [13] Mozilla Foundation, "Mozilla Foundation Reorganization: FAQ", (Accessed: May 28, 2008):<http://www.mozilla.org/reorganization/#q24>, 2005
- [14] M. P. Hamel (2007), Open-Source Collaboration in the Public Sector: The Need for Leadership and Value , NCDG Working Paper No. 07-004, Submitted June 22, 2007
- [15] D. Foster, Custom Corporate Communities: Planning and Getting Started, (Accessed: October 23, 2008): <http://fastwonderblog.com/2008/09/15/custom-corporate-communities-planning-and-getting-started/>, 15 September, 2008,
- [16] The Apache Software Foundation, How the ASF works, (Accessed: November 12, 2008): <http://www.apache.org/foundation/how-it-works.html>, 2008
- [17] André Felipe Machado, (Accessed: November 12, 2008): http://www.techforce.com.br/index.php/news/linux_blog/scientific_study_about_governance_and_organization, 18 April, 2008
- [18] Scilab, (Accessed: November 12, 2008): <http://www.scilab.org/>, 1989-2008, INRIA
- [19] Debian, (Accessed: November 13, 2008): <http://debian-community.org/>
- [20] CeCILL FREE SOFTWARE LICENSE AGREEMENT, (Accessed: November 12, 2008)(http://www.cecill.info/licences/Licence_CeCILL_V2-en.html), version 2.0, 2006
- [21] The Apache Software Foundation, Bylaws of the Apache Software Foundation, (Accessed: November 12, 2008): <http://www.apache.org/foundation/bylaws.html>
- [22] Ubuntu, The Ubuntu Community, (Accessed: November 12, 2008): <http://www.ubuntu.com/>