Managing Firm-Sponsored Open Source communities
- A case study of Novell and the openSUSE project

Master Thesis (60 credits / 1 year research)
Author: Jan Fredrik Stoveland janfst@ifi.uio.no
University of Oslo
Date: May 2008

Executive summary

Background
Ever since Netscape Communications announced their plans to make their Web browser an open source product by establishing the Mozilla project in January 1998, an increasing amount of software companies have been pledging to open source software development. We are now witnessing a range of different shapes and forms of open source initiatives blending in various ways with proprietary software and for-profit companies. Some established software firms have recently taken a step further than simply participating in open source communities and using open source software, as they have chosen to open up their own software projects in an attempt to create surrounding open source communities to support their own development. This is what O’Mahony and West describe as creating firm-sponsored open source communities (O’Mahony, 2007a; J. West & O’Mahony, 2005), which is the main theme for this thesis. In this research, we find that a major difference between firm-sponsored and autonomous communities is that the former need to handle a tension between openness and control in their product development. Given that developers earn the trust and prove their competence, the autonomous communities can basically offer the same opportunities for everyone to gain rights to commit code and part-take in strategic decisions. For firms, it is more problematic to offer this kind of openness to external community members, due to their need for control of the product upon which their business model relies. This relationship is equally present in the case of this study.

1 This thesis is submitted under the masters program of “Technology, Organization and Learning” at the University of Oslo, which is under the administration of the Department of Informatics. The interdisciplinary program is a collaboration of the departments of informatics, sociology and pedagogy, and emphasizes the role of information technology in organizational change and learning. Although this thesis draws upon knowledge from all domains, the theories I use are weighted towards a sociological perspective.
Methodology

I have studied the American software company Novell and the openSUSE project, largely guided by an inductive, qualitative approach supplemented by some quantitative methods. The majority of data were collected during my 1-month stay at Novell’s Linux R&D team in Nuremberg, Germany. The data include:

- 26 in-depth interviews with employees, managers and external community members
- 1 month observation at the company and participation at open source conferences
- Survey among active external contributors in the openSUSE community (280 respondents)
- Statistics on mailing list activity

The majority of the data was acquired in October 2007. The descriptions in the thesis present a snapshot of Novell’s evolution. A few months after the data were gathered the situation had already changed. This development is taken into account in the discussion of the future perspectives of the project.

The discussion of the case of Novell is as much an attempt to explore and test theories as it is an effort to describe and explain this empirical reality. This summary will however emphasize the more practical aspects of the findings in the study.

Research questions

By establishing the openSUSE community, Novell have on the one-hand extended their own organization. At the same time they are keeping the external developers at arms-length by not granting them any direct authority to commit code to the system. The first research question in the thesis has pursued the nature of this relationship: 1. What is the (theoretical) distinction between the sponsor firm and the sponsored community? I have found that the openSUSE community is distinguished as an interactive system from Novell’s decision-network. The openSUSE community is not an organizational system in its own right, but has features that nevertheless make it a fairly stable system that has a boundary towards Novell in the development of the openSUSE distribution. With the help of Niklas Luhmann’s theory on autopoietic organizational systems, we are able to see how Novell and the openSUSE community are separated apart from each other.

The relationship between these two systems are affected by the tension between openness and control, described above. The next step has been to investigate the exact nature of this

\footnote{Thanks to Justin Steinman, Meike Chabowski, Michael Löffler and Andreas Jaeger for accepting and facilitating my visit.}
relationship, and in particular how they are held together in collaboration: 2. What is the nature of the relationship between the organizational and the external interactive system, and how are the two systems bound together? Given the differences in the commercial tradition of software development and the open source software model, multiple inherent conflicts could be present in Novell’s open source adventure. If we also take into account the mentioned tension between openness and control in Novell’s development model, many obstacles and difficulties complicate this collaborative attempt. I have therefore been interested in seeking out the mechanisms, forces and arrangements that hold things together. In the thesis, I argue there are three elements that ensure a tight coupling between the two systems. By drawing upon the work of Susan Leigh Star (1989), I first discuss how several development tools and models serve as important boundary objects between the two systems, enabling joint development on a common product. The characteristics of boundary objects are that they are shared and have a mutual identity between all collaborators, and are able to support diverse interests at the same time. Examples of such objects are the Bugzilla repository, the Factory code base and the openSUSE Build Service. I have therefore put emphasis on the social aspects of these powerful software tools, which I find to be very important. For example, the solution for Novell has not been to battle disagreements on opinions and values, but provide room for all of them to co-exist. If a developer does not “win” in a development-dispute, the openSUSE Build Service will allow him to take things in his own direction, rally support and prove the others wrong.

Secondly, the shared communication channels (mailing lists, IRC, Wiki) are vital in creating transparency and providing access through the boundaries of the systems. It has been imperative that employed engineers have moved their development discussions to these channels. Thirdly, the marginal people whom have roles in both systems are crucial for balancing the needs of both of them. This group of individuals (approx. 80 people) are the people that in fact enact the hybrid software model in practice, and use both reactive and proactive strategies that counterbalance the internal and external pressure to separate the two systems, as illustrated below.
Managing openness

In summary, what can the analysis and discussion in this thesis tell us about the problem of managing the tension between openness and control between the sponsor firm and the sponsored community? The discussion shows that it is possible to maintain and develop both control and openness, as these properties are organized through two different kinds of autopoietic systems (the organizational and interactive system). Furthermore, these systems are linked through a boundary, or to be more precise, a set of boundary objects. The objects allow for control to be maintained while they also provide a great deal of freedom and opportunities for external development contributions. The general lesson we may learn from these insights is that the chances for success of the firm-sponsored community model are strengthened when:

1. activity and focus shifts from internal, naturalized object to boundary objects; (i.e. moving development from Autobuild to the Build Service)
2. transparency from the sponsor firm is followed by encouraged participation from the sponsored community on shared channels;
3. the group of marginal people is continuously expanded from both sides – by encouraging employees to communicate externally and by increasingly including external developers in the organizational decision-network; and
4. every possible strategy that may make the common target-object an individual property of everyone is pursued, through mechanisms of contribution that supports shared ownership. The latter may unleash the target-object’s power to motivate and draw collaborators tighter together.

Novell’s transparent form of development – through using the open communication channels and encouraging input and discussion – has been very important for relieving some of the existing tension in the collaboration surrounding the openSUSE distribution. As I have shown in my discussion of Novell’s future perspectives, the pressure may nevertheless build up over time if the situation does not continue to evolve in a direction towards more openness. In the thesis, several various future scenarios are discussed, including:
• Status quo - The relationship between Novell and openSUSE remains as it is.
• Integration - Novell and openSUSE engulf each other and become one and the same system.
• Extinction - The interactive openSUSE-system dies, and the development goes back into Novell.
• Divergence - Novell and openSUSE move in opposite directions, and openSUSE evolves into an autonomous organizational system.
• Failure - Novell’s business model fails and the company pulls out of the openSUSE project. openSUSE’s fate is uncertain and open for anyone to grab.
• Bridge - Novell and the openSUSE system continue to exist as today, but a clear bridge is built between the two systems, enabling mutual influence.

Novell are aware of the fact that more openness is needed to vent this pressure, and are already taking measures to ensure that the arm (that is holding the community at arm’s length) is “shrinking”. This is in accordance with the last strategy “Bridge” that is detailed in the thesis. Does this mean, then, that the firm-sponsored open source community-model must evolve towards more openness (and thus abandon control) to survive over time? My data seem to indicate so, in this particular case. However, Novell’s clever maneuvering and their remarkable technical engineering of the Build Service technology shows that there are ways of devolving control in a “controlled” manner, meaning that the openness can be managed and that the risk is therefore minimized. Further research on the topic of firm-sponsored communities could therefore aim at investigating if it is possible to control openness, and whether this is the solution for managing communities such as these over time. If successful, this model may prove that facilitating open innovation – by drawing upon the best of both the commercial and open source software models – is possible.