Key Concepts and Definitions of Open Source Communities

Ruben van Wendel de Joode
Twynstra Gudde Consulting, The Netherlands
Delft University of Technology, The Netherlands

Sebastian Spaeth
ETH Zurich, Switzerland

INTRODUCTION

Most open source software is developed in online communities. These communities are typically referred to as "open source software communities" or "OSS communities." In OSS communities, the source code, which is the human-readable part of software, is treated as something that is open and that should be downloadable and modifiable to anyone who wishes to do so. The availability of the source code has enabled a practice of decentralized software development in which large numbers of people contribute time and effort. Communities like Linux and Apache, for instance, have been able to connect thousands of individual programmers and professional organizations (although most project communities remain relatively small). These people and organizations are not confined to certain geographical places; on the contrary, they come from literally all continents and they interact and collaborate virtually.

OSS communities resemble other online communities in the sense that they are "collectives of individuals that cohere around a shared spirit" (Hollenbeck, 2006, p. 576) and they make use of the Internet as the dominant medium for communication and interaction (Hinds & Kiesler, 2002). This spirit is frequently associated with the hacker culture in which openness and freedom are essential characteristics (Himanen, 2001). What differentiates OSS communities from most other online communities is that its contributors create a viable and competitive product. For example, the Web server Apache delivers Web pages for roughly two thirds of all Internet domains, the DNS server Bind has a market share of at least 75%, and the operating system GNU/Linux is becoming more and more popular for both servers and desktop computers. Also, private and public organizations are increasingly adopting and moving to OSS.

Like other online communities, open source communities provide a unique possibility to study how social networks are created and how they evolve through time. Since all communication and interaction is publicly available, researchers can –in real time– observe and analyze how such networks are created and sustained (Ellis, 2002; Rheingold, 1994). On top of that, individuals in open source communities have moved beyond the level of mere interaction and have been able to produce a large number of public goods (see also Smith, 1992). Not only have they created competitive software, they have also created a) related artifacts such as sophisticated collaboration tools, and b) a set of legal constructs to protect their way of communication and interaction against outside threats (O'Mahony, 2003; van Wendel de Joode, de Bruijn, & van Eeten, 2003).

To summarize: OSS communities are online communities connecting huge numbers of individual developers and professional organizations. In the communities software is being developed that has achieved worldwide recognition. As such, OSS communities are possibly the largest and therefore arguably the most fascinating examples of online communities.

In this article we will provide an overview of the results from previous research on OSS communities, while simultaneously providing an introduction to the most important elements in the organization of OSS communities. We will focus on a description of what we believe are the most fascinating artifacts that have been created in OSS communities. We believe that these artifacts set them apart from most other online communities. Furthermore, they are crucial to understand how coordination is achieved in OSS communities.

In this article we will especially focus on three topics: a) the social constructs created to protect open source software and the related way of working; b) a selection of the collaboration tools to give an idea of

the innovative infrastructure that is created in OSS communities; and c) the most commonly defined roles of individuals in OSS communities. First, however, we will give some background on the origin of the term "open source."

BACKGROUND: THE ORIGIN OF OPEN SOURCE

Open source software is nothing new, although the term itself did not exist until 1998. In the early days, programmers happily exchanged source code and they were free to run, modify, and improve each others' creations. This practice came under pressure when, in the 1970s, companies started to create business models based on proprietary software and they started licensing binaries without granting users the right to modify the binaries. This caused Richard Stallman to erect the Free Software Foundation (FSF). The goal of the Foundation was and still is to provide free software. One of the most important inventions by Richard Stallman, however, is not software but is the GNU General Public License (GPL). The license makes use of copyright law to circumvent copyright law, which he found too restrictive. This clever tactic is known as "copyleft." The FSF explicitly allows anyone to charge for software. Their mantra is "free as in speech, not free as in beer." Still, the term free has caused confusion. Many confused free with "gratis" (implying not commercially usable) and with "hobbyist" (implying bad quality).

When Netscape in 1998 decided to release the code of their Mozilla Web browser to the public, Eric Raymond and others coined the term "open source." They wanted to emphasize that the source code is open but not necessarily gratis. While the technical definitions of free software and open source are very similar, the difference is that the first group emphasizes the moral and ideological aspects, while the latter focuses on the superior quality and the effective and efficient processes that lead to its creation. Advocates of the free software movement often point out that these advantages are valid and nice to have but not relevant; they would produce and use "free software" even if it were inferior to other software.

In order to avoid the ambiguous term "free," yet to emphasize the aspect of freedom of software, others have started using the term "libre software." Thus, depending on the level of political correctness and intention of an author, the terms free software (FS), open source software (OSS), free & open software (FOSS), and free, libre & open software (FLOSS) are used more or less interchangeably.

MAIN FOCUS OF THE ARTICLE: THE BASICS OF OPEN SOURCE COMMUNITIES

The Software Licenses

Open source licenses are an important aspect of OSS communities, since they determine what users and developers are allowed to do with the source code. As such they largely influence the collaboration and coordination processes that take place between participants in OSS communities. All OSS licenses are listed on the opensource.org Web site. At the time of writing the Web site had a list over 50 different licenses. Yet despite the vast number of licenses, the Freshmeat Web site shows that the General Public License (GPL), the Lesser General Public License (LGPL) and the Berkeley Software Distribution (BSD) license are used in over 75% of the communities. Therefore, we will focus our discussion on these three licenses alone.

The first version of the *General Public License (GPL)* was developed by Richard Stallman and released in 1989. Many projects, however, use the second version, which was released in 1991. One of the main goals of the GPL is to keep the source code in the public domain or the commons (Stallman, 2002). To ensure that the software remains in the commons, the license grants every user a number of rights and restrictions. The GPL gives every user the right to use, copy, modify and redistribute software that is licensed under it. This is because developers can understand and change software only when they have access to its source code. The GPL therefore ensures access to the source code by prohibiting distribution of modified software unless the modified version is accompanied by the complete corresponding source code. In other words, the GPL forbids anyone to modify and distribute GPL-licensed software without providing access to the corresponding source code. Thus, the GPL prohibits people and companies from keeping modifications to distributed source code private.

The GPL is subject to much critique. The main reason is its "viral aspect:" if source code licensed