Encyclopedia of Virtual Communities and Technologies

Subhasish Dasgupta
George Washington University, USA
Open Source Software Communities

Kevin Carillo
Concordia University, Canada

Chitu Okoli
Concordia University, Canada

INTRODUCTION

Open source software (OSS) development has continued to appear as a puzzling and enigmatic phenomenon and has drawn increasing attention as its importance has grown. Relying upon an alternative way to develop and to distribute software, open source communities have been able to challenge and often outperform proprietary software by enabling better reliability, lower costs, shorter development times, and a higher quality of code (Raymond, 2004). Behind the software is a mass of people working together in loose coordination, even portrayed as a rowdy marketplace (Raymond, 2001, p. 1):

No quiet, reverent cathedral-building here—rather, the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches ... out of which a coherent and stable system seemingly emerges only by a succession of miracles.

More precisely, the people behind open source projects have been defined as: “Internet-based communities of software developers who voluntarily collaborate in order to develop software that they or their organizations need” (von Krogh, 2003, p. 14). In contrast to the sacred cathedral-like software development model that gave birth to most commercial and proprietary systems, such bazaar-like communities seem to have basing their success on a pseudo-anarchic type of collaboration and developers’ interaction (Raymond, 2001). However, in spite of the apparent disorganization of these bazaars, a closer look distinguishes common values and norms that rule them, specific roles that can be identified, similar motives shared by people, and practices that follow patterns. This article highlights key aspects of what forms the communities that support these projects.

Definition of Open Source Software

The basic definition of OSS as expressed by the Open Source Initiative (www.opensource.org) goes beyond the notion of free code. It encompasses broader issues such as distribution and licensing that stipulate free exchange and modification rights of source code (OSI, 1997):

- Free redistribution of source code
- Free redistribution of compiled (binary) programs
- Derived works must be permitted
- Integrity of the author’s source code
- No discrimination against persons or groups
- No discrimination against fields of endeavor (e.g., commercial and military uses must be permissible)
- Mandatory distribution of open source license
- License must not be specific to a product
- License must not restrict other software’s licenses
- License must not restrict redistribution to a particular delivery technology

A BRIEF HISTORY OF THE OPEN SOURCE PHENOMENON

During the 1960s and 1970s, scientists and engineers in academic and corporate laboratories freely shared, exchanged, and modified the software they produced. However, by the early 1980s, software was increasingly shifting from its original shared nature to becoming increasingly commercialized, with licenses that forbade the free sharing of source code. In 1983, Richard Stallman left MIT to found the Free Software Foundation (FSF) with the principle aim of defining and diffusing legal mechanisms and conceptual principles of “free software” (Hars & Ou, 2001; West & Dedrick, 2001). “Free” here refers to freedom, the liberty to do whatever desired with the software. Hence, free software in the open source sense is distinct from “freeware”, which is software sold at no price. In fact, one of the explicit rights given to users of “free software” is the right to sell it commercially. It is noteworthy that most OSS is freeware (provided at no charge), but most freeware is not open source (the source code is not provided, and users are forbidden from modifying the program code even if they could).

Stallman’s publication of the GNU Manifesto (1985) allowed him to communicate his ideological insights about
the nature of software (von Krogh, 2003), and he convinced developers to join him in the GNU Project, whose primary goal was—and still is—the creation of a Unix-like free operating system. ("GNU" is a recursive acronym meaning, "Gnu’s Not Unix"). Accompanied by the continuous improvements of networking capabilities and of the Internet, this major step signaled the beginnings of open source practices organized through the formation of virtual communities. In 1989, the Free Software Foundation released the GNU General Public License (GPL) in order to ensure the preservation of certain freedoms in the copies and derivative works of a piece of software. The GPL assures these freedoms via the copyright mechanism, which permits free copying, modification, and distribution of software, with the condition that any distributed derivative works explicitly accord others the same rights.

In 1991, Linus Torvalds, a 21-year-old Finnish programmer, created Linux, a kernel for a Unix-based operating system that uses the operating system tools created by the GNU Project. Since then, this multi-user/multitasking platform has met tremendous success and is known for being powerful, fast, efficient, stable, reliable, and scalable (Edwards, 1998). In 1999, a survey estimated that the GNU/Linux operating system (popular known simply as “Linux”) was the operating system of more than 30% of Internet server sites. A recent release of the kernel (Linux 2.2.10) credits 190 key developers, though the total number of contributors was estimated to be around 1,200 (Dempsey, Weiss, Jones, & Greenberg, 2002; Stewart & Ammeter, 2002).

In 1998, Bruce Perens and Eric Raymond of the Open Source Initiative pointed out that the mistrust of many traditional firms towards the GPL was in Stallman’s term “free”, which is not very an attractive idea to the business world (von Krogh, 2003)—conventional wisdom says that you get what you pay for. Thus, they tried to refocus what they rebranded as the “open source” software movement by primarily focusing on the economic and engineering superiority of the “open source” approach to software development, in contrast to FSF’s more philosophically antagonistic approach towards “source-hoarding”. In response to their positive marketing, the term “open source” has become the preferred terminology for this approach. Other major open source projects have followed Linux’s success, including the Apache Web server (started in 1995) and the Mozilla Internet suite project (started in 1999 when Netscape released its Communicator suite as open source; Mozilla released the Firefox Web browser and Thunderbird e-mail client at the end of 2004).

PROFILE OF OPEN SOURCE COMMUNITIES

Lee and Cole (2003, p. 51) defined a virtual community as “a cyberspace supported by computer-based information technology, centered upon communication and interaction of participants to generate member-driven contents, resulting in a relationship being built up.” OSS development communities have been defined as groups of loosely connected programmers, who use the Internet as a medium for collaboratively developing, improving, and disseminating software (O’Reilly, 1999), and are recognized as a particular genre of virtual community (Ljungberg, 2000) and as virtual organizations (Crowston & Scozzi, 2002), which are characterized by: group members with a shared interest or goal; geographical distribution; and use of information and communication technology to communicate and manage interdependencies (Ahuja & Carley, 1998). They have been classified as communities of transaction (Hagel & Armstrong, 1997) and as “task- and goal-oriented” communities; that is, “communities striving to achieve a common goal by way of cooperation” (Stanojevska-Slabeva & Schmid, 2001).

Ideology and Values

One of the main emphases in OSS communities is the social interaction among participants through electronic communication. Rheingold (1993, p. 5) defines virtual communities as “social aggregations that emerge from the Net when enough people carry on public discussions long enough, with sufficient human feeling, to form webs of personal relationships in cyberspace”. This definition highlights the creation of relationships among people. As a consequence, the participants of an OSS community share a set of common values, and they adhere to the same ideology: the OSS culture.

The most commonly accepted view of OSS communities’ culture has been proposed by Raymond (2001) in his landmark paper, “The Cathedral and the Bazaar”, where he characterizes it as a “gift culture”, as opposed to an “exchange culture”. Exchange cultures are based on scarcity whereas gift cultures rely on abundance. Raymond argues that in gift cultures, social status is determined “not by what you control, but by what you give away”. Bergquist and Ljungberg (2001) have empirically demonstrated that such giving values and behaviors enable social relationships to be created and maintained in virtual environments. Assuming any ideology is a common set of shared norms, values and beliefs among the members of
Open Source Software Communities

Table 1. Ideologies and values of open source software communities (Stewart & Gosain, 2001)

<table>
<thead>
<tr>
<th>Norms</th>
<th>Values</th>
<th>Beliefs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taboo against forking projects</td>
<td>The best craftsmanship wins</td>
<td>With enough eyeballs all bugs are shallow</td>
</tr>
<tr>
<td>Distributing changes without cooperation of moderators frowned upon</td>
<td>All information should be free</td>
<td>Practice is better than theory</td>
</tr>
<tr>
<td>Removing a person’s name from project history, credits or maintainers</td>
<td>You don’t become a hacker by calling yourself a hacker—you become a</td>
<td></td>
</tr>
<tr>
<td>is not done without explicit consent</td>
<td>hacker when other hackers call you a hacker</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Non-trivial extensions of function are better than low-level patches</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and debugging</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Work that makes it into a big distribution is better than work that</td>
<td></td>
</tr>
<tr>
<td></td>
<td>does not</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ideologies</th>
<th>Language, Symbols</th>
<th>Narratives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stallman</td>
<td>&quot;Disre&quot;</td>
<td>The Halloween Papers</td>
</tr>
<tr>
<td>Raymond</td>
<td>&quot;SuRe&quot;</td>
<td>The Cathedral and the Bazaar</td>
</tr>
<tr>
<td></td>
<td>Free Software Foundation</td>
<td>BashoNet, Freshmeat, Sourceforge</td>
</tr>
<tr>
<td></td>
<td>CopyLeft</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Open source licenses</td>
<td></td>
</tr>
</tbody>
</table>

A community, Raymond’s view of OS communities can be characterized by the features listed in Table 1 (Stewart & Gosain, 2001).

Furthermore, another dimension that may be added to the OSS ideology is the hostility towards commercial software. The intensity varies from one community to another one, from a fundamentally hostile view of commercialized software (almost universally expressed towards Microsoft Corporation, currently the largest commercial software enterprise) to a moderate view that accepts the symbiotic nature and coexistence of open source and proprietary software (Ousterhout, 1999).

An organizational culture approach elicits further insights about the rules governing OS communities. Schein’s (1984) framework posits that the culture of organizations can be understood by examining their artifacts, values, and core assumptions. Sharma et al. (2002) have applied this framework to OSS, with the results summarized in Table 2.

Researchers in the area of virtual communities have pointed out the overall importance of trust. For instance, Carver (1999, 114) affirms, “People are drawn to virtual communities because they provide an engaging environment in which other people ... create an atmosphere of trust and real insight”. Trust is considered as a core assumption of the OSS ideology, considering that core developers need to work closely with one another as they implement different but interrelated code segments (Sharma et al., 2002). As a result, it was found that trust among group members in OSS project groups plays an essential role in facilitating development success (Stewart & Gosain, 2001). Thus, OS development processes have to be managed through trust-based relationships in order to ensure OSS project success and quality.

Participants’ Profiles and Roles

It has been common to characterize OSS community members as anarchistic crackers operating on the fringes of society, a stereotype that has been soundly dispelled by recent surveys (Fitzgerald, 2004). Several surveys have identified consistent traits among OSS community members. Over 95% of members are male. The average age is around 30, with a majority of people between 20 and 30. 20% of members are students and over 50% of people have IT jobs. Most developers have a bachelor’s (or equivalent) degree, many have a master degree, and under 10% have a PhD degree (Ghosh, Glott, Krieger, & Robles, 2002; Hars & Ou, 2001; Lakhani, Wolf, Bates, & DiBona, 2002).

OS developers tend to participate in a limited number of projects—an average of just two or three each (Dempsey et al., 2002). Developers modify the program source code and make important decisions concerning any future development. Non-developers are valuable in reporting bugs, and in suggesting feature enhancements. Although
Table 2. OSS organizational culture (Sharma et al., 2002)

<table>
<thead>
<tr>
<th>Artifacts</th>
<th>0 Electronic communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>0 Altruism</td>
</tr>
<tr>
<td></td>
<td>0 Reciprocity</td>
</tr>
<tr>
<td></td>
<td>0 Gift Giving</td>
</tr>
<tr>
<td></td>
<td>0 Reputation</td>
</tr>
<tr>
<td></td>
<td>0 Fairness</td>
</tr>
<tr>
<td></td>
<td>0 Shared risks and ownership</td>
</tr>
<tr>
<td>Core Assumptions</td>
<td>0 Trust</td>
</tr>
<tr>
<td></td>
<td>0 Loyalty</td>
</tr>
</tbody>
</table>

it is common to think of OSS participants as willing, independent enthusiasts, on deeper analysis it is evident there is a certain kind of hierarchy inherent in any project (Glass, 2003).

CONCLUSION

OSS communities are an important type of virtual community today, where members convene online with the common goal of producing software that is valuable both to developers and for the general public. While most OSS communities do not explicitly restrict membership (except perhaps to the inner core of developers), they are particular in their membership criteria, which by its nature requires considerable skills in computer programming and software development. However, as we have discussed, there is also plenty of room for non-developer user members who contribute ideas and directions valuable to projects, such as identifying bugs and suggesting new features. As the open source movement continues to grow both as a software development methodology and as a philosophical/social/political approach to intellectual property, OSS communities will have an increasingly important role in the software industry.

REFERENCES


**KEY TERMS**

**Free Software:** An earlier name for open source software, emphasizing the liberties given to end users and developers of derivative works. There is no requirement that the software be distributed at no charge; thus, distinct from freeware.

**Freeware:** Software provided at no charge to the user. Might be open source or proprietary; that is, the developer only permits redistribution and use, with no modifications permitted. In fact, most open source software is freeware, but most freeware is not open source.

**GNU General Public License:** The first and still the most radical open source software license, created for the GNU Project. Requires that all derivative works be equally free (in the open source sense); that is, all derivative works must provide the full source code and must permit free use, modification, and redistribution.

**GNU Project:** (Stands for, “Gnu’s Not Unix”) Established by Richard Stallman in 1983 under the auspices of the Free Software Foundation. Its goal was, and still is, to create an open source Unix-based operating system. This goal was realized in 1991 by Linus Torvald’s creation of Linux.

**Linux:** A Unix-based open source operating system designed for Intel-based microcomputers. The kernel was created in 1991 by Linus Torvalds, and it was added on to the GNU Project to form what is properly called the GNU/Linux operating system.

**Mozilla Project:** A project formed in 1998 when Netscape released its Internet tools suite for open source development. Released its flagship Firefox Web browser and Thunderbird e-mail client at the end of 2004. The Sunbird calendar component is currently under development.

**Open Source Software:** Software whose source code is liberally made available for use, modification, creation of derivative works, and redistribution.

**The Cathedral and the Bazaar:** Paper by Eric Raymond (most recent version in 2001) that contrasts the “Cathedral” software development approach of a closed hierarchy (e.g., for proprietary software and most open source software such as the earlier GNU Project) with the “Bazaar” approach of loose collaboration with light centralized moderation (as was used for the Linux and Fetchmail open source projects).