IJCSC 0973-7391 Volume 7 • Number 2 March 2016 - Sept 2016 pp. 258-261 available online at www.csjournals.com

Open Sources Technology: A Perspective Approach

Sharda Rani
Assistant Professor (Computer Science), RKSD College, Kaithal (Haryana)
duhan.sharda@gmail.com

Abstract: The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. For twenty years it has been building momentum in the technical cultures that built the Internet and the World Wide Web. Now it's breaking out into the commercial world, and that's changing all the rules. Are you ready**?** Over the past two years, there has been a surge of interest in open source software development. Interest in this process, which involves software developers at many different locations and organizations sharing code to develop and refine software programs, has been spurred by three factors.

I. Introduction

Software is just the beginning … open source is doing for mass innovation what the assembly line did for mass production. Get ready for the era when collaboration replaces the corporation

1. The rapid diffusion of open source software

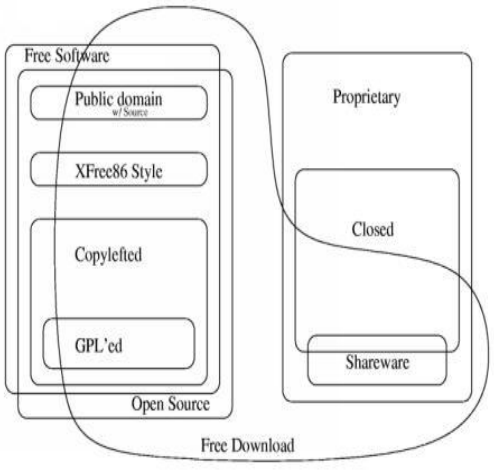
A number of open source products, such as the Apache web server, dominate their product categories. In the personal computer operating system market, International Data Corporation estimates that the open source program Linux has between seven to twenty-one million users worldwide, with a 200% annual growth rate. Many observers believe it represents a leading potential challenger to Microsoft Windows in this important market segment.

2 The significant capital investments in open source projects

Over the past two years, numerous major corporations, including Hewlett Packard, IBM, and Sun, have launched projects to develop and use open source software. Meanwhile, a number of companies specializing in commercializing Linux, such as Red Hat and VA Linux, have completed initial public offerings, and other open source companies such as Cobalt Networks, Collab.Net, Scriptics, and Sendmail have received venture capital financing.

3 The new organization structure

The collaborative nature of open source software development has been hailed in the business and technical press as an important organizational innovation.



II. Open source vs. free software

Open source software and free software are different terms for software which comes with certain rights, or freedoms, for the user. They describe two approaches and philosophies towards free software. Open source and Free software both describe software which is free from onerous licensing restrictions. It may be used,

copied, studied, modified and redistributed without restriction. Free software is not the same as freeware, software available at zero price. The Open Source Initiative believes that more people will be convinced by the experience of freedom. The Free Software Foundation believes that more people will be convinced by the concept of freedom. The Free Software Foundation believes that knowledge of the concept is an essential requirement, insists on the use of the term *free* and separates itself from the Open Source movement. The Open Source Initiative believes that *free* has three meanings: free as in beer, free as in freedom. *Open Source* says nothing about the freedom to modify and redistribute, so people who think that source access without freedom is sufficient are misusing it.

Open source is becoming more and more crucial to the information technology and consumer electronics industries every day.

III. What motivates programmers?

A programmer participates in a project, whether commercial or open source, only if person derives a net benefit (broadly defined) from engaging in the activity. The net benefit is equal to the *immediate* payoff (current benefit minus current cost) plus the *delayed* payoff. A programmer working on a software development project incurs a variety of immediate benefits and costs. First, the programmer receives monetary compensation if person is working for a commercial firm. Second, the programmer may be fixing a bug or customizing a program for his own benefit. Third, the programmer incurs an opportunity cost of person's time. While person is working on this project, he is unable to engage in another programming activity. The actual cost of this time depends on how enjoyable the work is.

The past few years have seen unprecedented growth of open source software. At the same time, the movement has faced a number of challenges. We will highlight two of these here: the "forking" of projects (the development of competing variations) and the development of products for high-end users. One issue that has emerged in a number of open source projects is the potential for programs splintering into various variants. In some cases, passionate disputes over product design have led to the splintering of open source projects into different variants. Examples of such splintering are the Berkeley Unix program and Send mail during the late 1980s.

Open Source process has some benefits over the closed source approach. As we noted, signaling incentives are stronger, the more visible the performance and the more attributable the performance to a given individual. Signaling incentives therefore may be stronger in the open source mode for three reasons:

1) Better performance measurement: Outsiders can only observe inexactly the functionality and/or quality of individual elements of a typical commercially developed program, as they are unable to observe the proprietary source code. By way of contrast, in an open source project, the outsiders are able to see not only what the contribution of each individual was and whether that component "worked," but also whether the task was hard, if the problem was addressed in a clever way, whether the code can be useful for other programming tasks in the future, and so forth.

2) Full initiative: The open source programmer is her own boss and takes full responsibility for the success of a subproject. In a hierarchical commercial firm, however, the programmer's performance depends on her supervisor's interference, advice, *etc*. Economic theory would predict that the programmer's performance is more precisely measured in the former case.

3) Greater fluidity: It may be argued that the labor market is more fluid in an open source environment. Programmers are likely to have less idiosyncratic, or firm specific, human capital that limits shifting one's efforts to a new program or work environment. (Since many elements of the source code are shared across open source projects, more of the knowledge they have accumulated can be transferred to the new environment).

IV. The "Gift Model"

Some describe the reasons they participate in open source projects is because they want to "scratch their own itch" or meet a need they have for the software personally or professionally. Another reason often given is to establish a "reputation" and get real world work experience working on a team. These motivations are confirmed in the following survey.

Top Motivating Factors

1. 44.9% say it is intellectually stimulating
2. 41.3% want to improve their skills
3. 3.8% say it is a function of their work
4. 33.1% believe the code should be open and available to everyone
5. 28.5% want to give back to the community and feel obligated because they use the software

V. Linux v/s windows

The main difference is that with Linux, you also have access to the source code, which allows you to build your own version optimized for your hardware (e.g. building for your old 486 pc will run faster than a generic build targeted at the most recent hardware), allows you to adapt it for your own specific non-standard purposes, allows you to fix bugs and security problems, and crucially allows you to "peer-review" the code, to look for loopholes and potential security breaches, and to verify that "it does what it says on the tin." Windows is written mostly by Microsoft employees, whereas Linux is written by contributing individuals and groups

Examples:

The basic philosophy behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, and people fix the bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing. Here are a few interesting facts/example Open Source projects:

Apache is the number one web server at nearly 62% of all installations. Second place was help by Microsoft at just under 27%. (Source: Netcraft)

GNU/Linux is the number two operating system at nearly 30% of all servers behind Microsoft operating system at nearly 50%. (Source: Netcraft)

Sendmail (a mail transport agent) sendmail has become one of the standards of the Internet's infrastructure (TCP/IP, Apache, sendmail).

MySQL founded in 1995 by two open source veterans, Michael "Monty" Widenius and David Axmark, with the help of Allan Larsson.and claims 4 million installations worldwide and 30,000 downloads of the software per day making MySQL by far the planet's most widely distributed open-source database. (Source: mySQL)

PHP is a widely used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML and as of May 2003 it was being used on 12,487,030 domains and 1,321,203 IP addresses. (Source: PHP)

Linux Operating System is over three years old and has grabbed 13.7 percent of the $50.9 billion market for server computers, and that figure is expected to jump to 25.2 percent in 2006, putting Linux in the No. 2 position. (Source: IDC)

SourceForge.net a site providing support tools and resources for the OSS/Free Software movement recently announced a major two year milestone of having more than a half million registered users, as well as a site record of nearly 63,000 registered projects. This is amazing growth of 25,000 new users per month. (Source: SourceForge.net)

VI. Types of Open Source Licenses

Open source licenses may be broadly categorized into the following types:

(1) Those that apply no restrictions on the distribution of derivative works (we will call these Non-Protective Licenses because they do not protect the code from being used in non-Open Source applications); and

(2) Those that do apply such restrictions (Protective Licenses because they ensure that the code will always remain open/free).

Software that has been placed in the public domain is free of all restrictions, all rights under copyright having been granted to the public at large. Licensors of Non-Protective Open Source licenses retain their copyright, but they grant all rights under copyright to the licensee. Licensors of Protective Open Source licenses retain their copyright, grant all rights under copyright to the licensee, but apply at least one restriction, typically that the redistribution of the software, whether modified or unmodified, must be under the same license. Licensors of propriety licenses retain their copyright and only grant a few rights under

copyright, typically only the rights to perform and display. The following table, where the BSD license is used as an example of a Non-Protective Open Source license and the GNU General Public License as an example of a Protective Open Source license.

Non-Protective Open Source licenses include: Academic Free License v.1.2; Apache Software License v.1.1; Artistic; Attribution Assurance license; BSD License; Eiffel Forum License; Intel Open Source License for CDSA/CSSM Implementation; MIT License; Open Group Test Suite License; Q Public License v.1.0; Sleepycat License; Sun Industry Standards Source License; University of Illinois/NCSA Open Source License; Vovida Software License v.1.0; W3C Software Notice and License; X.Net, Inc. License; zlib/libpng License; and Zope Public License v.2.0.

Protective Open Source licenses include: Apple Public Source License v.1.2; Artistic License; Common Public License v.1.0; GNU General Public License v.2.0; GNU Lesser General Public License v.2.1; IBM Public License v.1.0; Jabber Open Source License v.1.0; MITRE Collaborative Virtual Workspace License; Motosoto Open Source License v.0.9.1; Mozilla Public License v.1.0 and v.1.1; Nethack General Public License; Noika Open Source License v.1.0a; OCLC Research Public License v.1.0; Open Software License v.1.1; Python License; Python Software Foundation License v.2.1.1; Ricoh Source Code Public License v.1.0; and Sun Public License v.1.0.

Some Open Source licenses of both types include other provisions, such as restrictions on the use of trademarks, express grants of license with respect to applicable patents, disclaimers of warranties, indemnification of copyright holders in commercial distributions, and disclaimers of liability. However, none of these provisions are as fundamentally important as the obligations/restrictions that are imposed on redistribution rights under the Protective Open Source licenses, and it is with those restrictions on redistribution that we next focus.

VII. Conclusion

IBM joined the Open source world, Intel and Netscape have invested in Red Hat. Research laboratories have adopted the open source model because the sharing of information is essential to the scientific method, and Open Source allows software to be shared easily. Businesses are adopting the open source model because it allows groups of companies to collaborate in solving a problem without the threat of an anti virus lawsuit. To what extent do open source projects have a greater or shorter effective life span than traditional projects? One of the arguments offered by open source advocates is that because their source code is publicly available, and at least some contributions will continue to be made, its software will have a longer duration. But another argument is that the nature of incentives being offered open source developers, lead them to work on highly visible projects—might lead to a "too early" abandonment of projects that experience a relative loss in popularity. Some large corporations have adopted Open Source as a strategy to combat Microsoft and to assure that another Microsoft does both come to dominate the Computer Industry.

References:

1. Open Source Initiative,1999, "Open Source Definition", http://www.opensource.org/osd.html.
2. Di Bona, Chir, Sam Ockman and Hark Stone, editors, 1999, Open Sources :Voice from the Open Source revolution, Sebastopol, California: O' Reilly
3. Josh Lerner and Jean Tirole, 2002 " Some Simple Economics of Open Source ", Journal of Industrial Economics, Volume 50, number2, PP. 197.234
4. http://www.linuxworld.com/article/8689
5. http://www.opensource.org/docs
6. Alexander Hars and Shaosong Ou, 2001. "Working for free? - Motivations of participating in Open Source projects," Proceeding of 34th Annual Conference on System Sciences. IEEE computer Society Press. Pp. 2284-2292.
7. Tim O'Reilly, 1999. "Lessons from Open Source Software Development," Communications of the ACM, Volume 42, Number 4, pp.32-37.