Chapter 1.3
Open Source Software Basics:
An Overview of a Revolutionary Research Context

Eirini Kalliamvakou
Athens University of Economics and Business, Greece

ABSTRACT

The open source software (OSS) development area of research presents a fresh and generous domain for analysis and study. In line with this, it is important to have a high-level understanding of the "open source phenomenon" before being able to delve deeper into specific niches of research. OSS presents a rich picture and because of that, both academics and practitioners have shown intense interest in producing high-quality literature. This chapter provides an initial understanding of what OSS is and how it has come to be the exciting research platform that it is today, attracting attention from various sources. In addition, we take an overview of the research streams that have formed in recent years, and the basic findings of attempts made to transfer lessons from OSS to other research areas.

OPEN SOURCE SOFTWARE AT A GLANCE

Open source software (OSS) has received growing attention in recent years from various perspectives. The thriving numbers behind OSS adoption and contribution have captured the attention of academic research that, in the past years, has been trying to decipher the phenomenon of OSS, its relation to already-conducted research, and its implications for new research opportunities.

OSS has a definition that focuses on specific characteristics that software has to serve in order to be labeled as "open source." The Open Source Initiative (OSI) is a nonprofit corporation dedicated to managing and promoting the OSS definition for the good of the community; thus, acting as the official organization behind OSS. Based on the OSS definition provided by OSI, any software that has the characteristics listed below is considered to be OSS, and vice versa:

- Free redistribution
- Access to source code
- Derived works allowed under the same license
- Integrity of the author's source code
- No discrimination against persons or groups
- No discrimination against fields of endeavor
- Distribution of license
- License must not be specific to a product
- License must not restrict other software
- License must be technology-neutral

The current OSS landscape presents a very interesting picture. Although the idea behind OSS dates back to the 1960s and the UNIX era in the 1980s, the official term of OSS was coined in 1998 and, at the same time, the OSI was created. Since then, the OSS movement has evolved at a very fast pace. Prime examples of successful OSS projects include operating systems (Linux, FreeBSD, OpenBSD, NetBSD), Web browsers (Firefox, Konqueror), graphical environments (KDE, Gnome), productivity applications (OpenOffice), programming languages and infrastructure (Apache, MySQL), and development tools (GNU toolchain, Eclipse). These widely accepted OSS endeavors show that, today, a wide range of OSS applications are available and they present a viable and robust alternative to proprietary software solutions.

In addition to the presence of prime examples in the OSS environment, the plethora of OSS projects is impressive. Project support sites are online environments that provide tools for listing and managing OSS projects while supplying information such as developer teams, maturity stage, latest versions, and so forth. Two of the biggest and most well-known support sites, *Sourceforge.net* and *Freshmeat*, have reported to have more than 125,000 and 40,000 listed projects in the summer of 2006, respectively. Although many of these projects are still in designing stages, these numbers reveal the dynamics behind OSS and its evolution/adoption. Furthermore, this striking progress of OSS provides an intricate motive for conducting research in such a context.

OPEN SOURCE SOFTWARE AS A RESEARCH CONTEXT

Open source software is developed in a way different than proprietary software. Development is done inside communities of developers that work on code for their personal satisfaction or need. However, lately, a trend has formed inside large companies, that pay their employees to contribute to OSS projects, using this as a platform that enables them to affect the introduction of new software features so that the final OSS product is better aligned with the company's interests and needs. An open question remains as to how expanded this trend currently is, and whether the "paid volunteers" are involved in the process of developing OSS software primarily out of their own satisfaction or are assigned to it exclusively by their employers.

Independent of this fact, the community-oriented development leads to the efficient production of high-quality software available for use by anyone interested. This is an important element of OSS, the prime motivation for developers is not to "make software" as requested by clients or employers, but mainly to satisfy their own software needs, which cannot be fulfilled by vendor-supplied software. After the software is prototyped, it can be made public for anyone who wishes to use, modify, and redistribute.

OSS communities have significant similarities with professional software engineering teams utilized by software houses. However, these two organizational structures also portray critical differences that show that they stand quite far apart. It is important to note that since not all companies organize their software development efforts in the same mode, and also, there is a wide variety