

Open Source Software and the Digital Divide

K**Heidi L. Schnackenberg***State University of New York at Plattsburgh, USA***Edwin S. Vega***State University of New York at Plattsburgh, USA***Michael J. Heymann***Nazareth College, USA*

INTRODUCTION

Open source is a philosophy and a methodology associated with free and collaborative, creation, modification and use of software applications and operating systems. The term “open source” was coined in 1998 at the “Open Source Summit” in Palo Alto, California in an effort to standardize both the idea and a word or phrase for it. Shortly thereafter, the “Open Source Initiative” (OSI) was formed and sought to promote the idea of free applications (specifically, free source code) to the public (Tiemann, 2009).

Open source is characterized by a particular creation process associated with the development of software and/or operating systems. It is protected by specific licenses which allow users and developers the ability to freely modify, upgrade, use and distribute the products as they wish (von Krogh & Spaeth, 2007). Although not relegated to a hierarchical structure, a project manager generally emerges and leads the initiative by keeping things organized and moving forward while volunteers offer their talents by designing, coding, debugging, beta testing, and utilizing the product (Crowston & Howison, 2006). Volunteers can also offer feedback and provide technical support, either in-person, via phone, or in online user forums. This process can end in the free proliferation of an open source product or, at times, a profitable venture if the software is sold to a company and marketed commercially. O’Neill (2012) notes that open source can be viewed as a change agent. It has altered the way software and operating systems are produced and marketed, and has even modified the way knowledge is created. In this way, open source has

also redefined business and marketing models around the world (Bonaccorsi, Giannageli, & Rossi, 2006; von Krogh & Spaeth, 2007).

BACKGROUND

According to Bonaccorsi and Rossi (2003), the open source movement was not motivated by profit, yet gained momentum in an environment that was dominated by proprietary regulations. Conversely, the way that commercial software is conceptualized, produced, and marketed is quite different from the open source process. Proprietary software development is often highly secretive, thus creating a very insular process where only a certain number of people can be involved in the evolution of the product. In a global society and market, where knowledge and information aren’t held by just a few, and communication is constant through social networks and real-time, interactive video applications, the proprietary software creation process seems somewhat antiquated, often resulting in a product that has “bugs.” Although open source products are far from perfect themselves, the consumer isn’t paying for those imperfections like they are with commercial products (Schnackenberg & Vega, 2010).

The success of open source demonstrates that society possesses strong collaboration ability, both virtually and in person. De Florio and Blondia (2010) call this cooperation potential *social energy* and assert that it can be used to overcome the technological accessibility issues of certain groups of users (Mehra, Merkel & Bishop, 2004), such as individuals with disabilities

DOI: 10.4018/978-1-4666-5888-2.ch456

or impairments. In both of their 2010 articles, Sun, DeFlorio, Gui, and Blondia (2010a, 2010b) illustrate this phenomena in the creation and use of Ambient Assisted Living (AAL) programs, where human interaction (i.e. social energy) is the fundamental element in this “mutual assistance community” where both elderly and young people participate and assist each other with problems and challenges. The creation of use of a platform such as ALL precisely exemplifies the spirit of the Open Source Initiative.

In many senses, open source is a prevalent common practice of the hacker community, and a strongly held belief of technological idealists (Lakhani & Wolf, 2003). It is perhaps a type of high-tech grassroots movement. Open source products and applications are a manifestation of these practices and beliefs from which lay people can benefit. In his book, *The Cathedral and the Bazaar*, Raymond (2005) famously likens the culture and the creation process of open source to a bazaar, or an open market, where everyone has some unique goods to offer that all interplay for the benefit of the whole. Raymond contrasts this to the very hierarchical way in which ancient cathedrals were built. Despite its philosophical underpinnings and model of operations running entirely counter to conventional wisdom about how knowledge is created and how traditional business models function, the open source movement has taken a strong foothold in our technological culture (Weber, 2005).

ISSUES, CONTROVERSIES, PROBLEMS

Several challenges with open source and open source products exist with varying degrees of frustration for developers and users. The following is a description of some of the most prevalent issues with open source.

Confusing licensure: For the most part, OSS is free to modify for individual user needs. *Copyleft* is a term commonly associated with open source. It is a form of licensing that makes use of copyright law to ensure that a software, and its subsequent upgrades, can always be freely modified by users (van Holst, 2013). Different open source products, or different versions of open source products, can carry varying restrictions and usage policies. It is difficult to tell which software you can modify in what ways because there

is no consistency in the licensing policies. At times, it could be risky to utilize certain software in a particular way, or combine other software for a specific function, or to even create an entirely different product, for fear of violating licensing restrictions. To read all of the varying agreement policies that a developer or group of developers can attach to open source software can be extremely time-consuming and frustrating.

Collaborative creation process doesn't necessarily mean that the product will be better: One of the great attractions of the open source movement is that products can be modified by a developer or group of developers (Tapscott & Williams, 2006). The idea of software creation being a community process is novel to the computer application industry (von Hippel, & von Krogh, 2003). While most commercially produced software is highly propriety and the source code and creation process quite secretive, open source products are a highly collaborate event. Admittedly, the idea of making software free and open to the public is an enormous, and attractive, economic and philosophical shift. However, sometimes too many individuals on a project can make the process inefficient and the product inferior. This phenomena is commonly referred to as “Brook's Law” (Brooks, 1886) and is particularly applicable when it comes to working on source code. Computer coding is a very detail-oriented, careful skill and often if one programmer makes an error when coding, it is difficult for another programmer to find and fix it. In this way, a piece of software can become either unfriendly to use or at worst, dysfunctional. Indeed, sometimes adding more software developers to a project doesn't necessarily make the product better.

Not everyone knows about OSS: Comino and Marenti (2005) point out that only some in society know about open source while others don't. They state that open source developers and producers have little incentive to advertise and raise awareness, and recommend that government intervention, mandating adoption of open source by schools, organizations, etc. would raise awareness. The fact that more individuals don't use or even know about the open source initiative indicates it is still perhaps more of an underground movement than its proponents would like it to be, thereby creating a divide in society of those informed and able to acquire the products, and those who are not aware of a free option when it comes to software. However, while promotion of the open source concept may still

6 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/open-source-software-and-the-digital-divide/112907

Related Content

Classification of Sentiment of Reviews using Supervised Machine Learning Techniques

Abinash Tripathy and Santanu Kumar Rath (2017). *International Journal of Rough Sets and Data Analysis* (pp. 56-74).

www.irma-international.org/article/classification-of-sentiment-of-reviews-using-supervised-machine-learning-techniques/169174

Factors Impacting Defect Density in Software Development Projects

Niharika Dayyala, Kent A. Walstrom, Kallol K. Bagchi and Godwin Udo (2022). *International Journal of Information Technologies and Systems Approach* (pp. 1-23).

www.irma-international.org/article/factors-impacting-defect-density-in-software-development-projects/304813

Two Rough Set-based Software Tools for Analyzing Non-Deterministic Data

Mao Wu, Michinori Nakata and Hiroshi Sakai (2014). *International Journal of Rough Sets and Data Analysis* (pp. 32-47).

www.irma-international.org/article/two-rough-set-based-software-tools-for-analyzing-non-deterministic-data/111311

Cyber Security Protection for Online Gaming Applications

Wenbing Zhao (2018). *Encyclopedia of Information Science and Technology, Fourth Edition* (pp. 1647-1655).

www.irma-international.org/chapter/cyber-security-protection-for-online-gaming-applications/183880

The Importance of Systems Methodologies for Industrial and Scientific National Wealthy and Development

Miroljub Kljajic (2010). *International Journal of Information Technologies and Systems Approach* (pp. 32-45).

www.irma-international.org/article/importance-systems-methodologies-industrial-scientific/45159