Making Money with Open–Source Business Initiatives

Paul Benjamin Lowry
Brigham Young University, USA

Akshay Grover
Brigham Young University, USA

Chris Madsen
Brigham Young University, USA

Jeff Larkin
Brigham Young University, USA

William Robins
Brigham Young University, USA

INTRODUCTION

Open-source software (OSS) is software that can be used freely in the public domain but is often copyrighted by the original authors under an open-source license such as the GNU General Public License (GPL). Given its free nature, one might believe that OSS is inherently inferior to proprietary software, yet this often is not the case. Many OSS applications are superior or on par with their proprietary competitors (e.g., MySQL, Apache Server, Linux, and Star Office). OSS is a potentially disruptive technology (Christensen, 1997) because it is often cheaper, more reliable, simpler, and more convenient than proprietary software.

Because OSS can be of high quality and capable of performing mission-critical tasks, it is becoming common in industry; the majority of Web sites, for example, use Apache as the Web server. The deployment of OSS is proving to be a productive way to counter the licensing fees charged by proprietary software companies. An organized approach to distributing cost-effective OSS products is intensifying as companies such as RedHat and IBM co-brand OSS products to establish market presence.

From a business perspective, the entire OSS movement has been strategically anti-intuitive because it is based on software developers freely sharing source code—an act that flies in the face of traditional proprietary models. This movement raises two questions this article aims to address: (1) why would individuals write software and share it freely? and (2) how can software firms make money from OSS? Before fully addressing these questions, this article examines the historical development of OSS.

OSS HISTORY

A strategic irony of the software industry is that its foundation rests primarily on OSS principles. Software development in the 1960s and 1970s was steered primarily by government and academia. Software developers working in the field at the time considered it a normal part of their research culture to exchange, modify, and build on one another's software (Von Krogh, 2003). Richard Stallman, a professor and programmer at MIT, was a strong advocate and contributor to this culture of open, collaborative software development. Despite Professor Stallman's influence, MIT eventually stopped exchanging sourcecode with other universities to increase its research funding through proprietary software licensing. Offended by MIT's decision to limit code sharing, Professor Stallman founded the Free Software Foundation in 1985 and developed the General Public

License (GPL) to preserve free code sharing (Bretthauer, 2002).

In the formative years of the software industry, Stallman's free software movement grew slowly; in the early 1990s, however, the concept of code sharing grew more rapidly for a couple of reasons. First, "free software" was renamed "OSS," a name that spread rapidly throughout the code-sharing community (Fitzgerald & Feller, 2001). Second, the OSS movement received a boost from the advent of the World Wide Web (WWW). The Web provided an opportunity for Internet users to quickly and conveniently share their code.

WHY DEVELOPERS WRITE OSS

The majority of OSS software developers fall into one of the following three categories: freelancers, software enthusiasts, or professionals. Freelancers enjoy the challenges associated with developing OSS and providing services to the OSS community to further their own careers. When freelancers create modules of code, they often include their contact information inside the modules (Lerner & Tirole, 2002). This allows businesses to contact the developers to request their future services.

Software enthusiasts are people who contribute to OSS simply out of the joy and challenge of doing so, with little regard for professional advancement. Enthusiasts are often university students who want to participate in the development of free software and who receive personal gratification from participating in real-world OSS development projects and gaining the respect of the OSS community.

Even though OSS is "free" software, many companies hire professional developers to work on improving OSS code. RedHat, a Linux support company, hires developers to fix bugs in OSS code and to create new applications (Lerner & Tirole, 2002). Other companies hire OSS developers because their systems run OSS applications and they need developers to customize the code for specific business purposes. Table 1 summarizes the different motivations for joining OSS projects and shows them on a spectrum of intrinsic and extrinsic motivations.

SOFTWARE DEVELOPMENT ECONOMICS

Proprietary software

The strategic motivation behind the creation of proprietary software is to set up high switching costs for consumers. For such companies their developers' resulting source code becomes the company's intellectual property and an unshared key company asset. Once customers purchase proprietary software, they must pay for updates continually to keep the software current, and often to receive full customer support (Delong & Froomkin, 2000). Most customers will pay these fees because of the lock in that occurs from the often costly prohibitive tradeoff of implementing a completely new system.

Microsoft is an example of a company that has succeeded in proprietary software, largely because they have a focused strategy of selling complementary products and services to their installed base of Windows users (Shapiro & Varian, 1998): Offering

Table 1. Developer motivations

Enthusiast	Freelancer	Professional
• Learn • Earn respect	• Challenge of developing code • Receive future job opportunities	• Programming income • Customize OSS
Intrinsic		Extrinsic

Related Content

Content-Based Multimedia Retrieval
Chia-Hung Weiand Chang-Tsun Li (2005). *Encyclopedia of Multimedia Technology and Networking (pp. 116-122).*
www.irma-international.org/chapter/content-based-multimedia-retrieval/17236

Review of Fuzzy Image Segmentation Techniques
Gour C. Karmakar, Laurence Dooleyand Mahbubhur Rahman Syed (2001). *Design and Management of Multimedia Information Systems: Opportunities and Challenges (pp. 282-313).*
www.irma-international.org/chapter/review-fuzzy-image-segmentation-techniques/8111

Mobile Device Selection in Higher Education: iPhone versus iPod Touch
C. Brad Crispand Michael L. Williams (2011). *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts (pp. 1213-1225).*
www.irma-international.org/chapter/mobile-device-selection-higher-education/50649

Video Surveillance System Applications
 (2014). *Video Surveillance Techniques and Technologies (pp. 311-333).*
www.irma-international.org/chapter/video-surveillance-system-applications/94148

Application of Error Control Coding for Multimedia Watermarking Technologies
Mehul S. Raval (2010). *Advanced Techniques in Multimedia Watermarking: Image, Video and Audio Applications (pp. 407-424).*
www.irma-international.org/chapter/application-error-control-coding-multimedia/43480