

Customer Fulfillment in the Digital Economy

Open Source

Digital Economy Alchemy

“The Open Source world behaves like a free market or an ecology—a collection of selfish agents attempting to maximize utility which produces a self-correcting spontaneous order more elaborate and efficient than any amount of central planning could have achieved.”¹

—Eric S. Raymond,
Open Source anthropologist

“While making money with free software is a challenge, the challenge is not necessarily greater than with proprietary software.”²

—Bob Young,
chairman, Red Hat Inc.

“Apache is exactly what people want because it’s made by the people who use it.”³

—Bjorn Borud,
WebBuilder reviewer



360 Adelaide Street W, 4th Floor
Toronto, Ontario, Canada M5V 1R7
Tel 416.979.7899. Fax 416.979-7616
www.digital4sight.com

Scorecard

B-web type

- Alliance

KEY PARTICIPANTS

Customers

- Corporate and retail customers; i.e. Developers; Webmasters; Internet users

Context providers

- Open Source program distributors (e.g. Red Hat)
- Open Source projects
- Open Source project owner
- Co-developer

Content providers

- Developers; publishers; standards organizations

Commerce services

- Collab.net, Cosource

Infrastructure providers

- ISPs

Offering

- Operating systems; Software applications; Software development tools; Protocols (such as HTTP, IP); Psychic rewards

CFN value proposition

- Stable and flexible software and source code by a self-organized b-web of thousands of developers around the world
- Efficient and robust, Internet-enabled model for value creation that capitalizes on the strengths of each contributor

Success indicators

- Linux: 10–15 million copies worldwide
- Apache: 56% server market share for all www domains and 79% of secure domains

URL

- www.opensource.org
- www.linux.com, www.linux.org
- www.apache.org,
- www.freesource.com
- www.sourceexchange.com

Open Source

In the digital economy, value creation depends on new ideas and solutions to complex problems. Open Source is a radical new way of producing software. It forms the nucleus of a growing proportion of commercial operating systems and software applications. Software developers in the Open Source community collaborate on the Internet to produce software. If the software is useful to enough developers, it will be maintained in a continual state of renewal and improvement. In Open Source the producer may be synonymous with the customer. New models of fulfillment are forming around Open Source to capture the highly integrated value emerging from the community.

The success of the Open Source development model in the software industry raises interesting questions: How can businesses leverage the Open Source development model to improve their value offerings? In what other industries could this model be successfully applied?

The term “open source” derives from the central tenet of the community: that the “source code” of computer software, the computer language instructions that make software work, should be open and freely available. In “closed source” (proprietary) software such as MS Windows, the source code is not accessible to anyone but Microsoft—users cannot access or modify the source code; they only see the working software.

Part-time hobbyists and corporate software developers around the world are using the Internet to create a new model for developing software. Developers use the Internet to download source code, collaborate with other developers, and share their improvements with the Open Source community.

Open Source defies the traditional paradigms of complex software development. Large projects like Linux demonstrate that developers can self-organize to produce high quality and robust software that exceeds the quality of most commercial equivalents. Linux distributors like Corel and Red Hat harvest the fruits of the community’s development to release their own versions or “distributions” of Linux. These companies understand that their core competency is not software development. Instead, these organizations compete further upstream in the customer fulfillment network—branding, distribution, service, and support. By commoditizing the computer’s platform, Linux eliminates the competitive advantage of proprietary operating systems like Microsoft Windows.

Although the term Open Source was coined in 1998 by community leaders, it became a part of mass media and

investment terminology in 1999. The best-known open source software, Linux, grew 50% in 1999, after a 212% rate of installation growth during 1998⁴. During 1999, three Linux-related companies made very strong stock market debuts: Red Hat, Cobalt Networks, and VA Linux. Responding to the US Department of Justice antitrust case, Microsoft declared that Linux is a significant threat to its Windows operating system. In addition to Linux and Apache, there are dozens, perhaps hundreds, of other Open Source projects in development worldwide.

Business context

SOFTWARE EVOLUTION: In the very early days of computing, academia and the military produced most software because they were the only institutions with an acute need for computers. As mainframes and minicomputers opened corporate doors to computers in the 1960s, companies like IBM and DEC created and bundled proprietary software with their hardware. The customer was locked into the vendor’s hardware and software: if customers required upgrades, they had to go to the same vendor. Around that time, academics and hobbyists began to create and freely share software for personal or academic use.

When IBM introduced its personal computer in 1981, Microsoft made the fortuitous deal to license their Desktop Operating System (DOS) software. This enabled two important computer industry developments: licensing DOS made Microsoft very rich, and DOS became the de facto PC standard operating system. A standardized operating system meant that software companies could develop one version of their software and run it on any hardware that conformed to IBM’s PC standard. Hardware prices steadily declined, the computer user base grew, and software blossomed to become what is now a trillion-dollar industry worldwide. Commercial software is generally proprietary and protected by copyrights, patents, and licenses. The prevailing business model in the software industry is software as protected-closed intellectual property.

OPEN SOURCE EMERGES: Frustrated by the fragmentation of the UNIX operating system due to diverging commercial interests, Richard Stallman, a legendary software hacker and free software evangelist, created the Free Software Foundation in 1984 to promote the concept of free and open access to software’s source code. He devised the General Public License (GPL) copyright, whose premise was that software source code should be openly shared regardless of corporate interest. Any software published under this trademarked license must remain open and accessible. Linux, like most Open Source projects, is published under this license. Based on the GPL license, the prevailing model of the Open Source software community is protected-open intellectual property.

OPEN SOURCE MATURES: The recent popularity and success of Open Source software projects raises interesting questions about the nature of collaboration by individuals on high value-integration projects. How does world class software materialize, almost magically, from a group of developers (most of whom have never met in person) collaborating using only the Internet? What are the implications for the nature of the traditional firm or organization? In what other areas of collaboration could Open Source development models be successfully applied?

Open Source software is enjoying healthy growth rates and market share. During 1999, Linux grew to about 25% of the server market, representing annual growth of about 50%.⁵ Apache is used in 56% of all World Wide Web domain servers and 79% of secure (e-commerce) domains.⁶

The latest release of Linux, version 2.4, represents a concerted effort by the Linux Open Source developer community to create a desktop PC-friendly version with support for common protocols and devices. Linux version 2.4 is scheduled for official release in Q1 2000, and it includes support for common PC interfaces like USB, a high speed communication standard for printers, scanners, CD-RW drives, and digital cameras. Version 2.4 also supports the PCMCIA card standard, the common credit card sized devices found in most laptops. In January 2000, Corel Corporation released a Linux distribution that supports the use of most Windows-based applications like Microsoft Word, Excel, and PowerPoint.

Value proposition

Open Source software offers three compelling value propositions to the end user: stability, customization, and price. To a developer, the Open Source community provides an additional benefit—a sense of purpose and belonging.

STABILITY: The Open Source process involves significant peer review to ensure that the best source code rises to the surface. With a better code there are fewer bugs, and fewer bugs means less crashing. According to Ransom Love, President of Caldera, a Linux distributor, “Stability is key. And that’s where Linux is actually better than just about any other commercial OS that is shipping. Windows NT is not stable—Microsoft has created a void that Linux is ideally suited to fill and is filling.”⁷ In October 1998, three Microsoft employees anonymously leaked a series of documents to the public that described Microsoft’s internal research of Open Source. In these “Hallowe’en Documents”, as they were dubbed, author Vinod Valloppillil states, “Linux and other OSS advocates are making a progressively more credible argument that OSS (open source software) is at least as robust—if not more—than commercial alternatives.”⁸

CUSTOMIZABILITY: Sophisticated users can modify the source code of Open Source software for their own unique purposes. Systems administrators might tweak an Apache function to optimize the software for their particular network or rewrite a device driver for Linux. In Open Source software, all users have access to the source code to make their own modifications. These same users can share the modified code with the Open Source community. Users of proprietary operating systems like Windows NT or Sun’s Solaris must go to the manufacturer for any special modifications or features and likely pay substantial premiums for these changes.

LOW COST OF OWNERSHIP: The cost of downloading Open Source software to the end user is zero. Distributors like Red Hat and Corel make a modest profit by packaging an easy to install version of Linux with installation support for \$30 to \$100. Advanced technical support is purchased separately. This contrasts sharply with the \$4,000 price tag for Microsoft Windows NT. In a typical corporate environment the real costs of any system depend on setup costs, support, maintenance, and downtime. Linux has received accolades for its relative stability over most commercial operating systems such as Windows NT.⁹ A stable OS will crash less frequently and ultimately cost less in terms of downtime and labor.

NEEDS FULFILLMENT: Those involved in Open Source projects become part of an affinity group—hacking is the common bond among participants. Those who frequently contribute quality code generate respect and positive reputations in the community because of their talent. The hacker world is a staunch meritocracy, and in many respects, elitist. The standards are high, and belonging to the club is a prestigious honor. For some developers, the desire to scratch an altruistic itch is a primary motivation; they contribute time and energy in the service of a greater cause—the creation of better software.

CFN strategy

Open Source projects represent a self-organizing “shop” of developers collaborating in the common space of the Internet. Without the Internet, large Open Source projects couldn’t exist. As Vinod Valloppillil observes, “There is a direct correlation between the size of the projects that OSS can tackle and the growth of the Internet.”¹⁰ The Internet enables developers around the globe to download source code, test the software, modify it, and contribute these improvements to the community. The World Wide Web pages give Open Source projects a home by consolidating all Open Source project information, complete with hyperlinks, in a single, efficient interface.

In traditional commercial software development, removing software bugs before the official release date is of paramount importance. Testing and debugging is a serial process that results in long intervals between releases. Open Source undermines the traditional software development paradigm. Collaborating over the Internet enables Open Source developers to design, develop, test and debug the software in parallel. This massive parallelism leads to rapid development and continual improvement.

Most open source software evolves from a software developer's need, and the most successful projects are those adapted from existing software. The developer, now the "owner" of the project, tries to galvanize support and other developer resources by posting information about the project on related Internet newsgroups. If other users share the same need, they may join the project as users, developers, and testers. They may tell their colleagues about the project who may also become users and developers. The Internet speeds up adoption as developers frequently collaborate with colleagues around the globe. The project owner communicates regularly with the developers via e-mail. Developers and users simply sign up to e-mail lists to receive regular

communications. To concretize the project, many owners build a virtual homestead for the project in the form of a portal-like Web page. The Web page usually includes a project description, Frequently Asked Questions (FAQs), e-mail list sign-up, news, and File Transfer Protocol (FTP) access to the latest version of the source code. The Internet and World Wide Web enable the project owner to centralize fulfillment and communications, and co-ordinate the efforts of a distributed group of development resources.

Most Open Source software is not commercial. An individual can't buy the software in a store or download it from a vendor. The end user, most likely a frequent project contributor, must download the source code and compile the code to create the working software. The compiling process often requires significant technical proficiency. Often the only technical support available to the end user is a FAQs archive maintained by the project owner. Most Open Source developers are more concerned with creating cool software than catering to the needs of novice users. But over time, some Open Source projects like Apache, Sendmail, and Linux have caught fire with users, fueled by needs in the marketplace for which they are either the only solutions or the best solutions for the task.

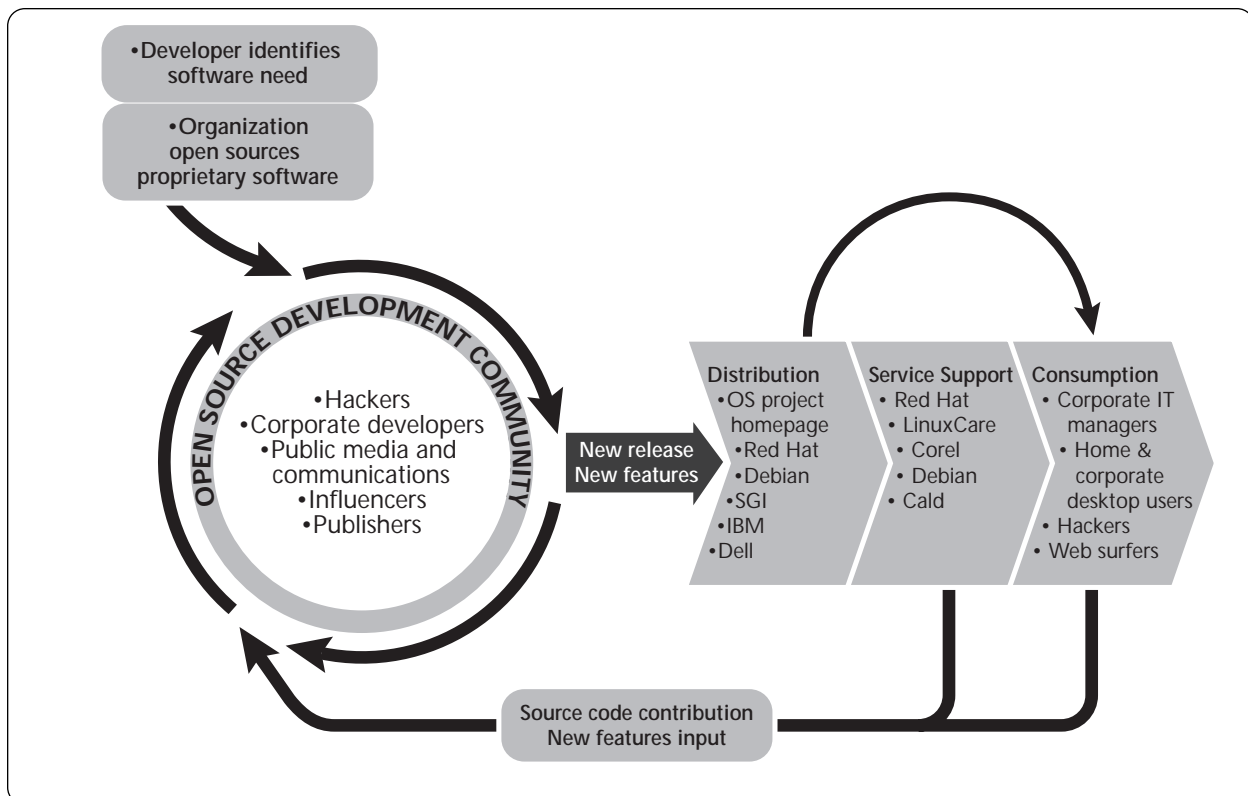


Figure 1. Open Source value chain.

The growing Linux market for novice and corporate customers opened an opportunity for companies like Red Hat, Caldera, and Corel to earn revenue by selling easy-to-use versions of Linux bundled with service and support. Now users have the choice of downloading, compiling, and installing Linux source code from the Internet (for experienced users only), or purchasing an easy to use Linux CD-ROM from a distributor.

The Open Source community and the distributor community have evolved to become a symbiotic software fulfillment network (see figure 1). The Open Source project community is the core value creation engine. Distributors add value by making the end user experience more convenient. They create easy-to-use versions of Linux, establish the sales and distribution channels, and provide technical service and support to customers. The distributors often contribute developer resources and source code to the Linux community as a show of goodwill.

Red Hat, one of the most successful Linux distributors, is particularly adept at integrating itself into the developer community. Red Hat designates developers whose full time jobs are to create Open Source code for Linux. It also maintains a central developer home page

for Linux users and developers and provides the most publically recognized brand of Linux. In January 2000, IBM announced that it will dedicate 200 developers to work on Linux for its enterprise servers and desktops. Any new code that it develops will be shared back with the Linux community.

Business processes and applications

Perhaps the best way to understand what makes Open Source tick as a development model is to walk through the process of contributing to Linux, the largest Open Source software project.

A Linux developer must first become familiar with the software itself. Contributors are very frequent users of Linux. Linux is available through about twenty different distributors (and growing monthly) including Red Hat, Corel, and Debian, and raw versions of Linux are available for download from the kernel.org Web site.

There are three main forms of source code contribution: identifying bugs, fixing bugs, and adding new features. This process of peer review is one of the most powerful processes in the Open Source community. With hundreds or thousands of high-end software users pouring over the software code, bugs and solutions are

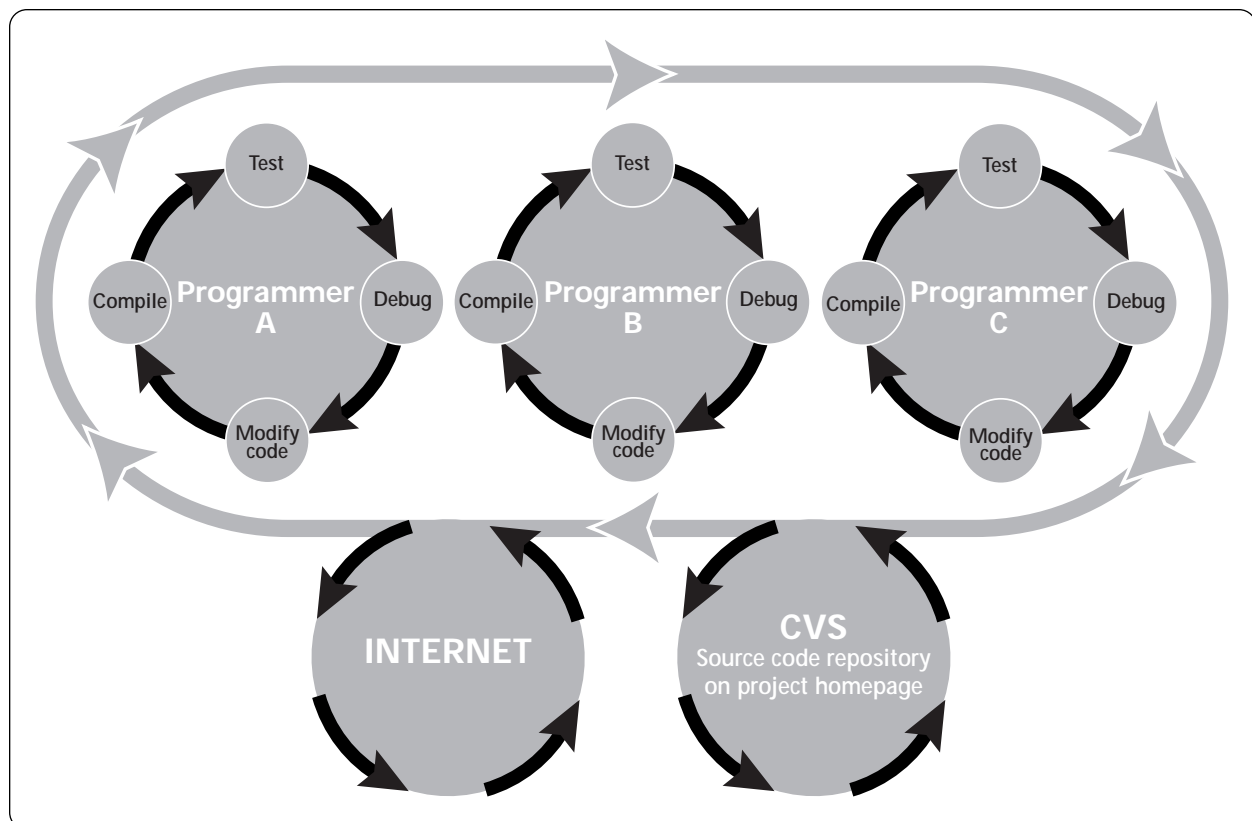


Figure 2. Parallel programming using CVS.

often obvious to someone. This is the same concept as academic peer review, but on a much larger scale. And in the meritocracy of Open Source, the most efficient code becomes integrated with the software. This enables parallel development and the rapid development cycles in Open Source.

Often, the person who identifies a bug is not the person who fixes it. Linus Torvalds articulates the process: “Somebody finds the problem and somebody else understands it. And I’ll go on record as saying that finding it is the bigger challenge.”¹¹

Each Open Source project with a home page will usually have a bug reports section where users have identified the bugs in the source code. Other developers may search the reports and post fixes. If the problem is a show-stopper, the project leader may encourage the developer community to focus its efforts on finding a fix to this bug.

To contribute a bug report or a fix, the developer joins one of the existing mailing lists for Linux, such as “linux-kernel-patch” from the linux.org Web site, and submits a report or fix. Alternatively, the developer may submit a contribution via e-mail to the module owner for peer review.

If the developer consistently delivers valuable contributions to the module, that module owner may grant special membership privileges to the developer. In Apache, this means full membership; in Linux it means frequent contributor status. In both cases, the developer now has access to the project’s Concurrent Versioning System (CVS) (see figure 2).

CONCURRENT VERSIONING SYSTEM (CVS): CVS is a code management tool that allows multiple developers from around the world to “concurrently” contribute to an Open Source project. Source code files for Open Source projects are stored in a central repository. When developers want to modify a file, they retrieve a local copy by FTP over the Internet to the CVS. Traditionally, if a central file is modified by one user, that file is locked out or inaccessible to other users. When the developer saves the modified file back to the CVS, the system saves only the change, without overwriting the original. Each change is inextricably linked to a specific developer.

Individuals with CVS access may check out all the source code files from the project’s CVS server, with up-to-the-minute code. Developers modify the code to their liking and verify that this improved version functions properly. When checking the file back into the CVS, the system returns modifications by other developers made since the first developer checked out the file. If the developer determines that the other

added code won’t interfere with their modifications, he then checks the file into the CVS. The CVS stores the developer’s modifications and contact information so that any other developer may contact him about further modifications.

DISTRIBUTORS: Linux distributors such as Red Hat assemble all the components of Linux source code and create an Install wizard for the end user. Distributors often add, modify, or delete features that access the core or kernel of the operating system. Feature modifications are another competitive differentiator for Linux distributors, in addition to service, support, and brand.

LICENSING: Unlike free software, Open Source software comes with conditions attached. Licenses ensure that Open Source code remains Open Source—it prevents proprietary modifications by commercial interests and discourages code forking. Licenses often stipulate that any code improvements to an Open Source project must be shared back with the originator or owner of the project. The most popular license is the GNU General Public License initiated by Richard Stallman. GPL incorporates the “copyleft” scheme—programs are copyrighted with the condition that the code remains open despite modifications and redistribution.

B-web organization

The core Open Source community consists of thousands of developer-users who work on dozens of independent software projects. Projects vary widely in size: a small project might only have five to ten developers, while the Linux developer community numbers in the thousands.

Each Open Source project constitutes an alliance of developers collaborating in the common space of the Internet. Open Source project participation spans a broad spectrum: from frequent, often daily, contributions to sporadic or single contributions. The barriers to entry for participation are very low, which results in rapid adoption of developers—but only if the project satisfies a common need. In the case of Apache, even non-technical users are encouraged to volunteer. They help organize local user groups, volunteer at conferences, or help with organization administration.

In the case of Linux, the b-web has evolved from an alliance of independent developers contributing in their spare time to an alliance of corporations contributing to a common standard. In his influential essay, *The Cathedral and the Bazaar*, Open Source proponent Eric Raymond points out that Linux’s early success was based, in large part, on the participation of self-selected developers “scratching a personal itch.”¹² Operating system software is very complex, and code contributions and bug fixes came from the cream of the developer crop.

Today, most participants contribute to the Linux community as part of their full time jobs at companies like IBM, Compaq, SGI, Corel, and Red Hat. Does this inhibit the quality of the software? Arguably no. Today, much Linux development focuses on adapting the software to support various hardware platforms and end user applications. These tasks are well suited to developers from the corporations who make the hardware and applications.

OPEN SOURCE PROJECT ORGANIZATION: Each Open Source project has a project owner, someone who leads the effort. In the case of Apache Webserver, the owner is a committee of members. Owners are responsible for making decisions about new features, allocating resources, resolving conflict, rallying the troops, and preventing the project from “forking.” This dangerous practice occurs when two or more project threads diverge.

The prevailing model of Open Source leadership is “benevolent dictator.” Although Open Source community leaders welcome input from all participants, the leader makes the final decision. A project leader may or not be the same person as the project founder.

In large projects like Linux and Apache, development is split into many modules, each with its own owner. Linus Torvalds is the undisputed benevolent dictator of the Linux developer community. He delegates responsibility and ultimately decides when the source code is ready for official release. Alan Cox is the owner of the core Linux module, the stable kernel. Approximately 200 developers are regular contributors to the Linux kernel. Within kernel development there are at least twenty different modules, each with owners and contributors. Outside the kernel, there are dozens of modules, each with an owner and developer base. A full list is available on <http://www.linux.org>.

Linux has proven that Open Source projects are scalable, both in the complexity of the project and the number of developers. Modularity is key. In this regard, Linux consists of a hierarchy of activities that all eventually feed into an official release of the software.

The Apache Foundation, on the other hand is a highly structured Open Source community. It has a central home page, the organization is incorporated, and there are strict bylaws and rules for membership. Decisions are made by a nine-member Board of Directors, presided over by Brian Behlendorf, one of Apache’s founders and now CTO of Collab.net. As of January 2000, Apache had 37 full members. Membership is by invitation only, and is based on consistent and quality contributions to the Apache

Foundation. Although hundreds of other individuals have contributed to Apache, membership is for the privileged few who have demonstrated commitment. Membership grants full CVS access to the project source code and gives voting power to the individual. Although members collaborate almost exclusively over the Web, they do meet annually to resolve outstanding issues in person.

Open Source represents an important new development model for collaborative value creation. The Internet enables skilled developers from around the world to create world class software such as Linux and Apache. For most Open Source projects, the end user is also a producer of the product. Linux has outgrown its hacker origins—many distributors sell Linux software to non-developer end users, complete with technical service and support.

Key Lessons

Open Source communities are pure alliances and the core value creating engines of larger fulfillment networks. Competition among Open Source software distributors is not about proprietary intellectual property. Instead, Open Source community practices ensure that the software remains a commodity. Distributors derive competitive advantage from how well they serve the needs of corporate and retail customers, while maintaining goodwill in the Open Source community.

Linux originally came from Linus Torvald’s need for a UNIX-like operating system for the PC. Previously, the UNIX standard had fragmented due to diverging or forking commercial offerings. This fragmentation crippled UNIX’s potential as a dominant PC standard. Linux distributors are very careful not to add or delete critical features that diverge from the path of the Open Source community, for fear of forking the project. Continual participation in the open source community by distributors like IBM, Red Hat, and Corel is critical for maintaining a common standard that will continue to grow the overall user base.

OPEN SOURCE MARKETPLACES: SourceXchange, Cosource, and Free Software Bazaar are online markets which match organizations that need software with developers who perform the work. In 1999, US corporations spent \$180 billion on contract and salaried software development. “If we can get a fraction of one percent of that, I’ll be happy,”¹³ says Brian Behlendorf, CTO of Collab.net, parent of sourceXchange. SourceXchange offers value-added services for its marketplace members, such as contract preparation, payment collection, and web-based project collaboration tools.

These organizations represent the deliberate seeding of Open Source projects. Aside from ensuring that the code is “open,” there’s not much else Open Source can do about these exchanges. Most of the projects from these sites break a fundamental tenet of Open Source development—successful Open Source projects thrive when developers identify and state a common need. These are really another way for organizations to contract software development.

APPLICATIONS OF OPEN SOURCE DEVELOPMENT IN OTHER INDUSTRIES: To understand the implications of the Open Source development model for other industries, it’s helpful to understand the environmental conditions in which Open Source thrives:

- Users share an important common need
- Users are producers of the end product—they are “prosumers”
- Barriers to participation are low; in Open Source, one needs only an Internet connection and some programming skill
- Meritocratic value system: the best contributions that address the needs of the project “win”

For complex technical products (such as operating systems) the following conditions are also critical:

- Presence of a critical mass of exceptional user-developers
- Sophisticated leadership to resolve conflicts, make resource allocation decisions, enforce rules
- Peer Review process to enable meritocracy and ensures robustness
- Parallelism: the product must be divisible, enabling parallel work by multiple teams.

As Open Source software gains greater customer market share, other industries are likely to try Open Source development models. Open Source isn’t just about customizing software, it’s about many customers using the Internet to continually improve a product. In Open Source, the origin of customer fulfillment is other customers. The collaboration aspect of Open Source is present in other industries. Cisco Connections is an online community where network design engineers post questions and collaborate with one another to formulate solutions. On health care sites like WebMD and MediConsult, members frequently seek and give advice in the online communities; the common need linking members is better health. Language, music, and government democracies are other examples of areas where Open Source-like alliances come into play.

—Chris McRaid

1. Eric S. Raymond, “The Cathedral and the Bazaar,” revised 8 August 1999, <<http://www.tuxedo.org>>.
2. Personal interview with Bob Young, Chairman, RedHat Inc., December 1999.
3. Bjorn Borud, “Apache gets down to business,” *WebBuilder*, 24 September 1998.
4. “Linux market share leaps by 212%,” *ABCNews online citing IDC*, 16 December 1998. <http://abcnews.go.com/sections/tech/CNET/cnet_linux981216.html>.
5. http://www.amcity.com/journals/high_tech/doc/2000/03/06/triangle_story4.html.
6. NetCraft Web Survey, December 1999, <http://www.netcraft.com/survey/>.
7. Personal interview with Ransom Love, President, Caldera, December 1998.
8. Vinod Valloppillil, “August 1998, V. 1.00,” <<http://www.opensource.org/halloween/halloween1.html>>.
9. <http://www.deja.com>.
10. Vinod Valloppillil, *ibid*.
11. Raymond, *op. cit*.
12. Raymond, *op. cit*.
13. Justin Hibbard, “The Trailblazer,” *Red Herring*, December 1999.

 DIGITAL 4SIGHT
360 Adelaide Street W, 4th Floor
Toronto, Ontario. Canada M5V 1R7
Tel 416.979.7899. Fax 416.979-7616
www.digital4sight.com