

# Free and Open Source Software: Business Aspects

By: Dr. Salah Mubarak Elhaggaz

[salah@elhaggaz.net](mailto:salah@elhaggaz.net)

## ABSTRACT

Many of today's most innovative products and solutions are developed on the basis of free and open source software (FOSS). Most of us can no longer imagine the world of software engineering without open source operating systems, databases, application servers, Web servers, frameworks, and tools. Brands such as Linux, MySQL, Apache, and Eclipse have shaped product and service development. They facilitate competition and open markets as well as innovation to meet new challenges. De facto FOSS standards such as Eclipse and Corba simplify the integration of products, whether they're all from one company or from multiple suppliers. This article has assembled this theme section to provide a brief yet practical overview of where FOSS is heading from a pure business perspective.

## Introduction

The open-source model has a lot to offer the business world. It's a way to build open standards as actual software, rather than paper documents. It's a way that real world industry and individuals can collaborate on a product that none of them could achieve alone. It's the rapid bug-fixes and the changes that the user asks for, done to the user's own schedule. The open-source model also means an enhanced security features; because code is in the public view it will be exposed to extreme scrutiny, with problems being found and fixed instead of being kept secret until the wrong person discovers them. And last but not least, it's a way that the little guys can get together and have a good chance at beating a monopoly.

Among all features affiliated to good quality software the most fundamental ones are the enhanced reliability measures. And if that's too abstract for you, you should think about

how closed sources made the Year 2000 problem worse and why they might have very well killed your business.

## **The Reliability Problem**

The reliability of most software has been atrociously bad. The foundation of the business case for open-source is high reliability. Open-source software is peer-reviewed software; it is more reliable than closed proprietary software. Mature open-source code is as bulletproof as software ever gets, as it combines the input from different business perspectives.

Until recently this was a radical idea to many business people and organizations; many had a belief that open-source software is necessarily not "professional," that it is carelessly made and more likely to fail than closed software. But the Internet's infrastructure makes the best possible disclaimer. Since OSI was founded in 1998 many people have been paying attention to this field.

Considering some examples of popular industry and business use such as; DNS, sendmail, the various open-source TCP/IP stacks and utility suites, and the open-source scripting languages such as Perl that are behind most "live" content on the Web. These are the running gears of the Internet. These open-source programs have demonstrated a level of reliability and robustness under fast-changing conditions (including a huge and rapid increase in the Internet's size and utilization) that, considered against the performance record of even the best closed commercial software, is nothing short of astonishing.

You can read an extended technical argument for the superior reliability of general open-source software in "<http://www.catb.org/esr/writings/cathedral-bazaar/>". This paper was behind Netscape's pioneering decision to take its client software open-source. It describes a bazaar or a marketplace style of managing software development that depends on open source and leads to high reliability and quality standards. The real-world evidence backs this up. In an independent head-to-head reliability test, open-source UNIX systems and utilities were less fragile - crashed or hung less often - than their proprietary counterparts. .

The business implication of this technical case is clear. Eventually, bazaar-mode peer review will come to be considered a necessary condition for highest quality. In many market niches, software that has not been peer-reviewed simply won't be perceived as good enough to compete.

## **The Payoff for Software Producers**

Bazaar-mode development seems to reverse our normal expectations about software development; more programmers are better (at least, as long as the capacity of the project leader or project core group to handle integration isn't exceeded). Even a small open-source project can gather more brains to improve a piece of software than most development shops can possibly afford. Under the open-source model whether you're producing software for internal use or for resale, the following benefits are likely to be evident:

### ***Development Speed***

The bazaar-mode development should be able to grab, and keep, a substantial initiative advantage over those that don't. The first commercial developer in a given market niche to switch to this mode may gain substantial advantages over those who join later.

This is because the pool of talent available for bazaar recruitment is limited. The first bazaar project in a given niche is more likely to attract the best co-developers to invest time in it. Once they've invested the time, they're more likely to continue with it.

### ***Lower Overhead***

Switching to the open-source model should also be good for a significant overhead reduction in per-project software production costs. The open-source model allows software shops to (in effect) outsource some of their work, paying for it in values less tangible than money. This means smaller shops will be able to handle bigger projects.

## **The Payoff for Software Merchants**

For those who produce software for sale, they are most likely to see two more advantages:

### ***Closeness to the Customer***

One of the most often-repeated pieces of management advice to marketing led organizations is "Stay close to the customer." In today's fast-moving, short-product-cycle business climate it's more important than ever to be close to customers and to understand what are the customer needs and be able to rapidly respond to those needs.

It's worth pointing out that the open-source, bazaar-method resembles the way many successful Japanese and Indian companies have done consumer product development; get a product to market that works but is not perfect, and iterate quickly based upon customer feedback to reach the combination of features that the customers need and want. This has turned out to be especially valuable for high technology products (laptops, personal assistants, cellphones, network management systems etc) where users don't exactly know the functionalities and features that they need.

### ***Broader Market***

An important side-effect of the open-source model will be a much wider platform range for your product. Open-source authors frequently find themselves receiving, for free, port changes for operating systems and environments they barely know exist and can't afford developers to support. This is a very vital and important gain for any business environment. Each such port, of course, widens the market appeal of the product.

## **The Payoff for Entrepreneurs**

For an entrepreneur or start-up software producer, taking the open-source approach is a way to grab mind-share. The best new concept in the world won't make money unless

people know it's interesting. Whether this makes sense as a strategy depends on how we vision the main value proposition, whether it is in the software itself or in service and the expertise associated with the software. A normal answer from a business perspective is that the value is actually in service and integration.

## Four Ways to Win

Now from an investor's point of view. We can least identify four known business models for making money with open source:

1. **Support Sellers (otherwise known as "Give Away the Recipe, Open A Restaurant"):** In this model, you (effectively) give away the software product, but sell distribution, branding, and after-sale service. This is what (for example) **Red Hat** does.
2. **Loss Leader:** In this model, you give away open-source as a loss-leader and market positioner for closed software. This is what Netscape is doing.
3. **Widget Frosting:** In this model, a hardware company (for which software is a necessary but strictly a cost rather than profit center) goes open-source in order to get better drivers and interface tools cheaper. Silicon Graphics, for example, supports and ships **Samba**.
4. **Accessorizing:** Selling accessories - books, compatible hardware, complete systems with open-source software pre-installed. It's easy to trivialize this (open-source T-shirts, coffee mugs, Linux penguin dolls) but at least the books and hardware underlay some clear successes: **O'Reilly Associates**, and **SSC** are among them.

The open-source culture's exemplars of commercial success have, so far, been service sellers or loss leaders. Nevertheless, there is good reason to believe that the clearest near-term gains in open-source will be in widget frosting.

For widget-makers (such as semiconductor or peripheral-card manufacturers), interface software is not even potentially a revenue source. Therefore the downside of moving to open source is minimal.