

Adopt Open Source Software or Die

“Business Strategies for Free and Open Source Software Companies and Communities”

By **David Pool, MBA**
Naked Ape Consulting
<http://naked-ape.com>

“Open source heralds a global paradigm shift in social and economic value creation of enormous proportions, the extent of which is almost completely unappreciated.” - Mitch Kapur¹

Introduction

Open source software (OSS) is turning the IT industry on its head. By using the Internet to collaborate, programmers have managed to develop high quality software outside of the bounds of traditional software corporations.

The Linux operating system is one of the best known examples. In the 1990's, IBM lost an operating system fight with Microsoft in which each side tried to control the operating system space. By the late 1990's IBM lent its support to Linux in an effort to establish a level playing field for itself and others. Large players like Oracle, HP, and Dell helped out, perhaps recognizing that “community property” is better for the ecosystem than domination by a single corporation.

This isn't to suggest that IBM led the Linux fight early on, far from it. Linus Torvalds started work on Linux all by himself in his spare time while attending the University of Helsinki. His efforts gained support from others via the Internet.

These dynamics are important to keep in mind, even now that OSS has become popular. Large corporations are not the historical source of OSS innovation. Small teams, working intimately with the code and problem space are better sources of entrepreneurial innovation. Another lesson is that several companies can work together for the good of their ecosystem as a whole.

2005 – The end of the beginning

At an the Angel Oregon conference in Portland recently, VC Dave Chen pointed out that many of the obvious OSS businesses have been done.

¹ <http://blogs.osafoundation.org/mitch/000815.html>

MySQL and PostgreSQL are strong OSS database products. The Linux operating system is well accepted. The Apache web server is closing in on 70% market share. Firefox is a fantastic web browser and Open Office has matched MS Office feature by feature.

Throw in mature development tools like vim and emacs, OSS languages like Perl, PHP and Python and you have targeted most of the low hanging fruit.

From a game theory perspective, we're talking about an early phase in the game where the chess pieces are carved, a board assembled, and the rules are worked out. Players are still wrapping their heads around how the knights and bishops move.

What Chen called derivative or second order business models would take the foundation provided by the early OSS pioneers and build another industry on top. When asked, Mr. Chen confessed he had no idea what the innovative business models were that would usher in the next phase of OSS development. To torture the analogy, it's as though the industry has been playing checkers, and only recently swapped out some of the pieces for chess pieces. No one's quite sure how the new game is going to be played.

Today I want to explore some of the dynamics of how those second order business models might work. I believe that there are a lot of opportunities, both within the IT industry and outside it. But first, I'd like to do a brief exploration of some business strategy concepts.

Sustainable Competitive Advantage

Harvard's Michael Porter practically founded the field of modern business strategy in the early 1980's with his books Competitive Strategy and Competitive Advantage.

He concluded that there were basically only three fundamental strategies: Cost leadership, Differentiation and Focus. Put simply, he is saying that to compete you have to be either cheaper, better or narrowly focused on meeting the needs of a niche market.

Free, Better and Niche

Toyota used Total Quality Management to achieve the triple combination of higher quality, lower costs and a defensive lock on several niches from which it could safely launch offensives. It's a sweet spot to be in, and it's what OSS is doing to the proprietary ecosystem.

The OSS technology stack provides a clear competitive advantage on cost by:

- 1) Providing free components such as operating system, database, programming language, development tools, software modules, web servers, file servers, and email servers.
- 2) Tapping into a worldwide developer pool and so dispersing development costs among a wider number of firms.
- 3) Support venues like the PLUG (Portland Linux Unix Group) list where technical support can be obtained for free.
- 4) Lower maintenance costs, especially downtime and viruses.

That adds up to a clear cost advantage for the OSS ecosystem.

OSS is also dominating because of its higher quality. Its higher quality is derived from some of the same tricks that Toyota used. For example, OSS reduces inventory – changes to the Linux kernel are in use within weeks or months compared to years for a Windows OS release.

OSS style development also allows a community of the best minds in the world to work openly on solving problems on the cutting edge. This is the approach to excellence taken by science and academia. The Internet enables OSS developers to scale the approach up to global participation.

OSS also allows companies to expand into new niches because there are no boundaries to where they can go. While the proprietary stack is composed of layers of opaque code from different competing companies, the OSS stack is transparent. Any barrier can be addressed.

You don't see companies like Google lashing together hundreds of thousands of windows machines with Visual Basic. Amazon didn't use Perl just because it was cheaper than ColdFusion. Perl lets companies like Amazon and Yahoo and E-bay go wherever they want to. Innovative young companies don't want to go through Microsoft to answer the question "Where do you want to go today?". They just want to follow their curiosity and business model wherever it leads.

Cheaper, Better Niche is a nice place to be, but if you can't make any money selling free software, what's it get you? Well let's take a look at what are becoming some standard business models for OSS.

10 Ways to Make Money with free Software

John Koenig presented 7 OSS business models in his ground breaking paper² presented at O'Reilly's 2004 OSCON Convention in Portland last year. We've taken his start and added three more to round out an even 10.

Publishing

O'Reilly itself is a good example of a profitable OSS business model. O'Reilly sells books and holds conferences around OSS. By doing so they contribute valuable knowledge to the ecosystem, support developers who are also contributing to the ecosystem, and position themselves out at the head of the learning curve – ready to take advantage of new business opportunities as they arise – nice business if you can get it.

Optimization

Koenig's first model builds on a lower cost infrastructure to optimize the value delivered higher up the software stack. Koenig's description Oracle's strategy makes it sound a lot like Cost Leadership:

“Oracle leveraged its database solution with commodity Linux and server hardware, optimizing the Oracle RAC product for Linux clusters, and thereby allowing Oracle to price its software at a higher margin. Hardware for the Oracle Unix (non-RAC) solution would have cost \$2 million more, with no better performance than Linux on commodity hardware. Oracle could deliver the Linux RAC solution at an \$800,000 premium, while still saving Electronic Arts more than \$1.3 million.”

Dual License

Dual license is a fairly clever concept. The main idea is to provide a free open source license to anyone but also requires businesses that build on it to open source the software they develop. However, businesses can pay to get a different license that allows them to sell their software without open sourcing their own code.

Part of what I like about this strategy is that it is a surcharge on proprietary software. The proprietary ecosystem is used to paying these price premiums. In this approach the proprietary system subsidizes the OSS ecosystem. Sweet.

2 <http://management.itmanagersjournal.com/management/04/05/10/2052216.shtml>

Consulting

If one can make a living as a proprietary software based consultant, one can make the same living using OSS tools. The ability to provide access to free solutions provides a cost structure advantage.

Three main areas for consultants to earn money with OSS are:

OSS tools – installation and support of OSS solutions.

Development – developing new software or adding to existing software.

Migration – moving customers off proprietary systems.

Subscription

Subscription is a recurring fee that companies pay to an OSS firm to provide software updates, configuration support and support services. Red Hat is a Linux vendor who earns revenues under this model. It recognizes that software development is going to be an ongoing process for businesses that want to remain competitive and that keeping systems updated is often best done by professionals who can keep up to date on issues as they arise.

Patronage

This model recognizes that some corporations can sponsor someone to work on a software project because it benefits the ecosystem which in turn benefits the business. Think of IBM sponsoring programmers to work on the Linux kernel. But also consider any business that allows it's employees to participate in OSS projects. In a way, patronage can be thought of as a form of pure research – enabling the global IT community to work on issues that interest them.

Hosted

The Hosted strategy plays off of a “loophole” in the GPL. One only has to distribute the source code of an application if one sells it. By providing services over the Internet companies like Google, E-bay and Amazon can make money off of their improvements to GPL code without having to give away those improvements.

Each of our strategies has their strengths and weaknesses. While the hosted strategy provides a defense against competitors free loading off one's work, it also discourages the development of a OSS style collaboration on the source code. Hosting can also provide a strong customer “lock-in” of data and skills which can scare away potential customers.

Embedded

The embedded strategy makes use of a different “loophole”. If you sell hardware that happens to have software embedded in it, you're not selling software so don't have to give away the code. TiVo runs on Linux as do several cell phones.

The embedded strategy is also a market niche that draws the same benefits of lower cost and better tools as other OSS plays. Competing with a lower cost structure, the OSS embedded companies can cooperate to commoditize the lower portions of the software stack and optimize higher up.

Both the embedded segment and the super computer segment are obvious areas where the OSS “no barriers” advantage plays a key role. Anyone is free to take Linux wherever their curiosity or business model takes them. IBM used Linux to build the world's most powerful computer³ while Motorola, NEC and Samsung have put it in their cell phones⁴

Hardware

IBM and HP earn a lot of money selling hardware. By removing a layer of the cost structure (the operating system) they are able to divert those dollars to pricing discounts, profits, or investments in marketing, product research etc.

Training

The Portland area is lucky enough to serve as the home of Stonehenge Consulting Services⁵. Stonehenge has made a nice living by providing Perl training for many years. This is still the tip of the iceberg. There are lots of people trained in proprietary software tools but a much smaller number in OSS. This isn't as true as it was five years ago, but there is still enormous opportunity in this area. And it's not just in the hardcore geek stuff, there are markets to be developed in training Open Office, the GIMP, Evolution and other end user OSS applications.

Thanks to the K-12LTSP project⁶, local Portland schools are turning out students who are trained in OSS applications. Multnomah ESD students are creating a workforce capable of being productive on the Linux desktop, using free applications and delivering a lower cost structure to

3 <http://www.top500.org/>

4 <http://www.linuxdevices.com/articles/AT9423084269.html>

5 <http://www.stonehenge.com/>

6 <http://k12ltsp.org/contents.html>

prospective employees.

OSS Succeeds

- Companies that create the future say to themselves “With all this potential customer benefit, there must be a way to make some money in here somewhere.” *Competing for the Future* by Gary Hamel and C.K. Prahalad

The point is that OSS has many ways of surviving and thriving. There are lots of ways to make money around OSS besides selling binaries. The ten models listed above aren't exhaustive and none of them require a frontal assault on a heavily fortified position. History advises against fighting on foreign soil, against superior forces while attacking a well fortified position. Think “Charge of the Light Brigade”.

OSS doesn't need to sell binaries, especially desktop operating systems to succeed. Conversely, because OSS can succeed in so many other ways, it can defeat those Goliath like forces without having to engage in direct conflict.

Governments and nonprofits also have a stake in growing the OSS ecosystem. In Oregon alone city, county and state governments are all using OSS. International governmental support for OSS is even stronger.

Good for the OSS IT Industry, But So What?

- The dominant new ecosystems will likely consist of networks of organizations stretching across several different industries, and they will joust with similar networks, spread across still other industries.
 - *The Death of Competition* by James F. Moore

So far we've only discussed where we are as we shift from the childhood of OSS to its adolescence. A small number of companies have learned how the pieces move and have established winning games for themselves. But suppose you're not in the IT industry. Does OSS really matter?

I'd argue that it matters a great deal. I'd argue that the adolescence of OSS isn't going to be so much a fight within the IT industry as the OSS industry is going to be the arms dealer in the battles that go on in almost every other industry in a fierce global competition. I'd argue that OSS is going to be the “Intel Inside” of successful industry clusters as they feast on the lunches of slower moving and less efficient

independent companies.

The good news is that we're still very early in the next stage of the OSS revolution. Most of the business models addressed so far are first order businesses. The second order businesses are still forming, learning how the pieces move and seeking teammates.

In a way Google, Yahoo, Amazon and E-bay straddle the two generations. They built their businesses on the cost and capabilities of OSS. However, none of them have used OSS style collaboration to drive the IT innovation at the core of their business. Each guards their IP assets secretively.

Through the Looking Glass

So far we have looked at how to use OSS to achieve a Sustainable Competitive Advantage over the competition. That's a fairly traditional approach. However cooperation is at the heart of open source. What's needed is an approach that builds Sustainable Cooperative Advantage.

The ten business models we examined above are still fairly traditional models. Most of them have a mechanism to create an edge so they can compete successfully. The OSS businesses that are on the cutting edge are those that are trying to learn how to cooperate successfully.

Studying Cooperation

Game Theory reduces cooperation to it's bare minimum in a game called Prisoner's Dilemma. The "game" is played by two participants who each face an identical choice, cooperate or defect. Each player's score depends on the combined out come. If they each cooperate they get 5 points each, if they both defect they each get nothing. If one cooperates while the other defects the defector wins 10 points while the cooperator loses 10 points.

Prisoners' Dilemma

Prisoner B	Prisoner A	
	Cooperation	Defect
Coop	A + 5 B + 5	A + 10 B - 10
Defect	A - 10 B + 10	A 0 B 0

In *The Evolution of Cooperation* by Robert Axelrod explored cooperation through computerized tournaments of this game. Cooperating successfully requires players to adopt strategies that help both sides stay in the upper left corner despite the temptation of each side to defect and earn exceptional profits in the short term.

In addition to insightful studies of how evolution can evolve in a population and of ecologically stable systems (both cooperative and non-cooperative), Axelrod derives five suggestions for promoting cooperation.

1) Enlarge the shadow of the future

The more likely one is to face the other player in the future, the higher the incentive to cooperate for fear of retaliation. In the business of OSS, this results from the time and energy it takes to learn and contribute to OSS software. Since participants are committing to the use of shared code, it follows that developers and customers have a disincentive for stabbing each other in the back.

To quote Axelrod "The principle is always the same: frequent interactions help promote stable cooperation."

2) Change the payoffs

The GPL license is one way that OSS has helped promote cooperation. While one can still fork a project, the GPL removes much of the monetary incentive for doing so.

3) Teach people to care about each other

This may seem naïve from a business perspective, but building good relationships is a sustainable cooperative advantage. Good relationships make good business sense.

4) Teach reciprocity

As nice as cooperation sounds, prisoner's dilemma tournaments weren't won by always cooperating. It's too easy to be exploited. The strategy that won most was called TIT FOR TAT which started by cooperating and then reciprocated whatever the other player did. In OSS, punishing defectors is often accomplished by shunning and avoiding players who are exploitative.

5) Improve recognition capabilities

Shunning and avoiding are only possible if a player can recognize another player the next time they meet. In OSS circles reputation is the coin of the realm. Building a good reputation and working with others with strong reputations encourages cooperation.

Importantly, as information flow approaches the ideal of perfect information, cooperative strategies do better while competitive strategies fail. For example, strategies that seek to create barriers to exit by trapping customers in proprietary data formats become less attractive as customers learn about the strategic trickery. A cooperative strategy that provides open data formats which are in a clients' best interest, become more attractive to clients as they know more.

A great acid test for developing successful strategies is how it performs under an assumption of perfect information. In contrast to many competitive strategies, cooperative strategies succeed more as more players adopt them. As Brandenburger and Nalebuff say in their book *Coopetition*, "Where conventional wisdom goes wrong is in ignoring the possibility of win-win strategies. Not a surprise, given the conventional mindset of business-as-war. With win-win, imitation is healthy."

The evolution of a cooperative business ecology

In James F. Moore's book, *The Death of Competition*, he staked out five stages in the evolution of a cooperative business ecology.

- I) Pioneering the terrain of opportunities
- II) The revolution spreads
- III) Defending the revolution
- IV) The Red Queen effect
- V) Renewal or Death

While OSS tools and technologies are reasonably mature, the business models around those tools are still in stage I. Here Moore explains

“During this period, entrepreneurs struggle to form embryonic ecosystems that, while hardly mature, are at least complete enough to fulfill the needs of initial customers. In this stage, they attempt to establish “proof of concept” – in other words unmistakable evidence that here is a viable and exciting alternative to the status quo.”

This new world of cooperative business ecosystems is where the future of OSS business models lies. Right now, we are still in the wild west stage – opportunities are everywhere for those with the fortitude and enthusiasm to pursue them.

This may make the shift to OSS seem risky, but doing nothing is risky too. You need to consider the possibility that some other business is going to pair OSS's competitive advantages with its cooperative advantages and arrive in your competitive space. If your business hasn't even figured out how the pieces move by that time, you may never have the time to learn how the game is played before it's over for you.

“Allowing a competitor to take the lead because one is absolutely certain the competitor is overcommitting financially or investing prematurely is one thing; surrendering leadership to a rival simply because one doesn't have a point of view about the future is something entirely different.”

Gary Hamel and C.K Prahalad – Competing for the Future

Strategies in a Cooperative Ecosystem

Designing a strategy for success in a cooperative ecosystem is a new challenge. Some models for how to achieve this follow.

Cooperate Globally, Compete Locally

One ecologically stable strategy is to focus on meeting the needs of customers in a local geographic region while cooperating with others who are solving the same problems but in their own region. As each player contributes their small piece to solving customer problems through improved software, the ecosystem grows richer for all.

Cooperate Locally, Compete Globally

In a flip flop, local businesses can cooperate on software that allows them to compete successfully abroad. Options to avoid local skirmishes include options like working across industries, market segments or focusing on rapidly expanding markets.

Focus on Core Competencies (Cooperate Locally, Cooperate Globally)

The core strength of any business is the ability to apply intelligence to the information at the heart of the business. If the software that's managing the information at the core of your business isn't evolving, your business isn't evolving.

OSS style collaboration on improving that software is a double edged sword. On the one hand, it provides an advantage over businesses that aren't cooperating while at the same time it levels the playing field for businesses that are sharing the software. On a level playing field, you may need to be better to succeed.

In that scenario, advantage comes from having a head start in learning and continuing to learn at a faster pace than potential rivals. This is what Moore was referring to as the Red Queen effect. The ultimate advantage comes from being able to learn how to improve faster than other players. Like the Red Queen in Alice in Wonderland, one ends up having to run faster just to stay in one's place.

The good news is that most businesses haven't even heard about the new game, while you have a head start on learning how it's played.

Take away points

- 1) Seriously question a strategy of waiting for someone else to come prove to you that OSS can provide competitive advantage in your industry. That may prove to be an expensive education.
- 2) Consider entrepreneurial ventures. OSS is opening up opportunities to overturn entire industries. Established businesses are at a disadvantage because they're locked into proprietary systems.
- 3) Develop a plan to derisk the exploration of OSS within your organization. Get to know how the pieces move now while there's still time to do that cheaply.
- 4) Explore ways of networking with other businesses who are interested in achieving cooperative advantage.

Recommended Reading

Axelrod, D. *The Evolution of Cooperation* (New York: HarperCollins – Basic Books, 1984).

Brandenburger, A., and Nalebuff, B. *Co-opetition* (New York: Currency

Doubleday, 1996).

De Geus, A., *The Living Company* (Boston:Harvard Business School Press, 1997).

Dixit, A., and Nalebuff, B. *Thinking Strategically* (New York: Norton, 1991).

Dorner, E. *The Logic of Failure* (New York: Metropolitan, 1996).

Hamel, G., and Prahalad, C.K. *Competing for the Future* (Boston:Harvard Business School Press, 1994).

Hawken, P. *Growing a Business* (New York: Simon & Schuster, 1987).

Mintzberg, H., Ahlstrand, B., and Lampel, J. *Strategy Safari* (New York: Free Press, 1998).

Moore, J., *The Death of Competition* (New York: HarperCollins, 1996).

Porter, M., *Competitive Advantage* (New York: Free Press, 1985).

Senge, P., *The Fifth Discipline* (New York: Currency Doubleday, 1990).

Wells, S., *Choosing the Future* (Boston: Butterworth-Heinemann,1998).

Licensing

This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/2.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.