Seven Open Source Business Strategies for Competitive Advantage

JOHN KOENIG

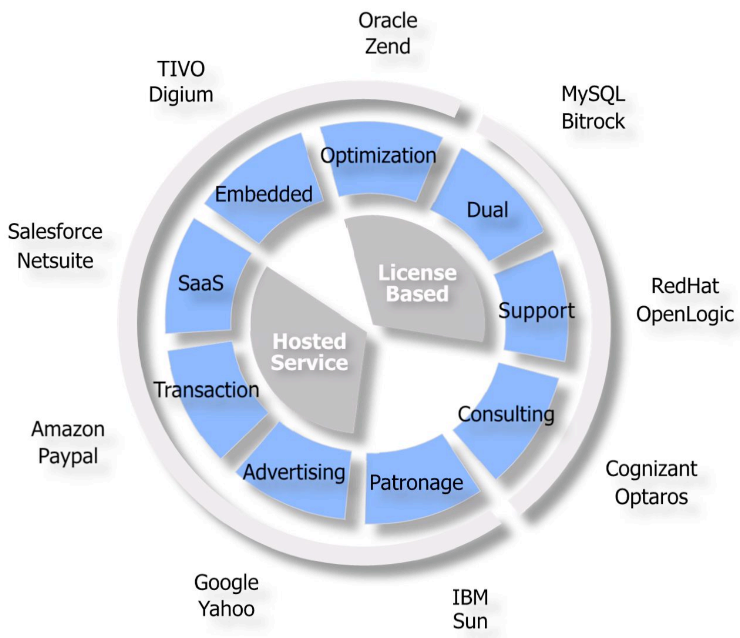
JOHN KOENIG is President at Riseforth and is responsible for research coverage of enterprise productivity solutions. He may be contacted at jkoenig @ riseforth.com.

Individual and enterprise users of software today have many options for satisfying their computing and networking needs. Open source software is one of them, and it is often selected because of the broader choices open source can deliver.

For instance, open source offers enterprises the opportunity to be more self-reliant through source code modification. It allows incremental project and upgrade schedules, free reign in integration decisions, and direct interaction with the user community. It creates the opportunity to implement projects in a way that is consistently mindful of enterprise goals, rather than the goals of a proprietary software vendor. Open source allows enterprises to select from a broader range of hardware and software vendors, and services providers than proprietary solutions. For these and other reasons, the pace of Linux and open source adoption continues to accelerate.

Open source presents a large potential competitive advantage for hardware and software vendors, and vendors of complementary or substitute services. Linux has contributed greatly to the adoption and success of open source. Companies like IBM, HP, Red Hat, Oracle, and recently. Novell, have invested in, and legitimized the use of Linux, for enterprise applications including data center operations. Linux related services deliver over a billion dollars in annual revenue to both IBM and HP. Oracle strongly promotes and likewise derives revenue from the Linux platform, with the "unbreakable Linux" guarantee. There are many strategies around open source platform applications and utilities aside from Linux or an open source solution stack. These strategies include substantial marketing and service alternatives that are creative and highly competitive.

An open source initiative for instance, may establish an industry standard. A relatively straightforward and simple open source marketing decision may reposition a company or product. For example, using the "patronage" strategy, IBM embraces and extends open source software with refinements that may help them pursue new markets or position themselves against established competitors more effectively. Likewise open source creates business challenges for many traditional software vendors. Companies like Sun, BEA and Wind River currently feel the impact of open source on their business as open source threatens to commoditize parts of their software portfolio.



The companies above illustrate some of the open source strategies being used to foster innovation, create product value, attract customers, and generate revenue. Each of these strategies are explained in greater detail in the following pages.

THE OPTIMIZATION STRATEGY

The optimization strategy is an open source manifestation of Clayton Christensen's *"law of conservation of modularity"*. In the open source application of Christensen's law, one layer of a software stack is *"modular and conformable"* allowing adjacent software layers to be *"optimized"*. The modular and conformable layers are commodities, and are unprofitable or only marginally profitable software businesses. The Linux operating system is an example. The disruption caused by a modular and conformable operating system such as Linux, serves to erode margins for other operating system vendors like Sun, Wind River, and Microsoft.

Winners under Christensen's law are the adjacent, interdependent layers of the software stack, the layers where applications are optimized to achieve greater value, and where correspondingly, better pricing power exists.

Oracle provides an excellent example of an optimized adjacent layer. In this example, Electronic Arts needed fast, reliable servers for its online version of the popular "Sims" game. Oracle proposed the Linux version of its Oracle 9i Real Application Cluster (RAC). Oracle has a long history of supporting multiple operating systems. In fact, open source platform portability was one of the early competitive advantages of Oracle. Portability created an implied assurance that customers would not get locked into a single hardware and operating system vendor.

Linux running on the typical Intel server lacks some features that are found in other operating systems and platforms. One such feature is the database clustering available on Sun Solaris. To compete on the demanding SIMs project at Electronic Arts, Oracle delivered a competitive database solution by porting and optimizing the Oracle RAC (Real Application Cluster) product to fun on commodity Linux x86 servers. By utilizing less expensive commodity x86 servers available from dozens of reputable vendors, including Dell, HP and IBM, Oracle could replace a more expensive bundled Solaris solution from Sun.

The lower cost of the software-based Oracle RAC clustering solution allows Oracle to price its software at a higher margin and still charge customers less than the Sun Solaris clustered Oracle solution. In the case of the SIMs project at Electronic Arts, hardware for the Oracle Unix (non-RAC) solution on Solaris was priced $2,000,000 more, with no better performance than Oracle RAC running on commodity Linux servers. According to Mainstay Partners, Oracle delivered the RAC solution on Linux at an $800,000 premium versus Oracle licenses for Sun Solaris, while still saving Electronic Arts over $1,300,000 dollars.

THE DUAL LICENSE STRATEGY

Under the dual license strategy, a software company offers free use of its software with some limitations, or alternatively offers for a fee, commercial distribution rights and a larger set of features. In the dual license approach, free use carries certain conditions; typically, any modifications that are distributed must also be made public in source code form, and companies cannot use the free version as a component of any product or solution they commercialize. This prevents third parties from developing improvements that would rival the original open source software.

The dual license approach is not typically one integrated license. It is a business policy that permits a customer to choose one of two licenses: either the commercial license or an open source license. So what is the incentive for dual license vendors to license software without charge? A free option facilitates new business in a number of ways, including improved customer awareness and faster adoption, stronger competitive positioning, and a large base of users to find bugs and recommend improvements to the software.

The dual license allows interested prospects a pain free path to application development and testing. Developers experience no business complications in exercising the software in a trial project. The right to use software internally for free, without disclosure of their modifications, is more than a money-back guarantee. Competitively this creates a wide advantage over highly supervised trial licensing practices of the past.

Any commercial license requires a metric by which the customer is charged. For the MySQL database, the commercial metric is a per-server fee. MySQL and Actuate with its BIRT product incorporate a tiered approach by charging higher fees for more functionality.

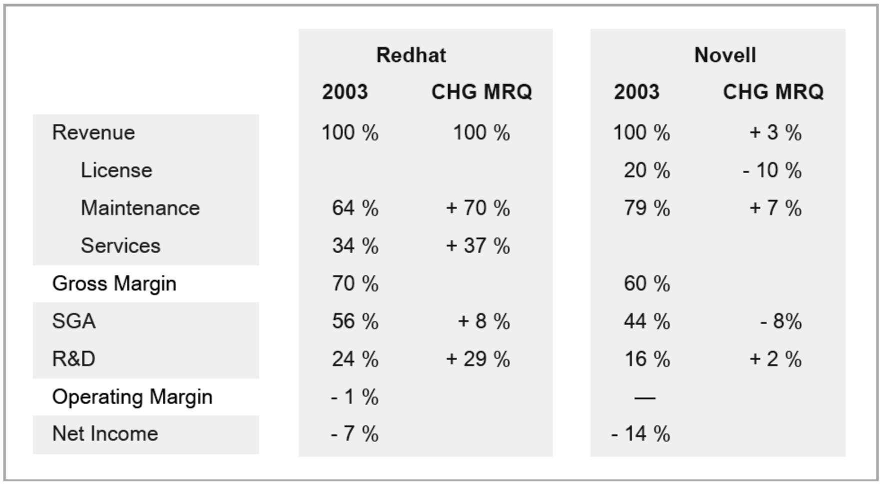
Many companies apply other strategies in combination with the dual license. For instance, Bitrock gains complementary maintenance and professional service revenue by offering a free version of its software to qualifying open source projects. Dual licensing helps MySQL capture a larger user base. Much of MySQL's massive adoption success is attributable to its simple installer features.

The dual license strategy provides a powerful tool to build a strongly defensible market position. Free software projects sometimes generate high numbers of downloads and broad awareness. By comparison, there have been, and still are, hundreds of software companies which have invested, in aggregate, billions of dollars, only to each gain small numbers of customers.

THE SUPPORT STRATEGY

According to analysts at Culpepper, *"revenues from services – both maintenance and consulting – increase in proportion relative to revenues from licenses... Move out to the 20 year mark, and the typical software company will have $2 of services for every $1 of licenses."*

The table below illustrates the support strategy being applied by many open source companies. It compares financial percentages of Red Hat (past 9 months) with Novell (past 12 months) from reports in Q1 2004 . It suggests why Novell acquired SuSE, a supplier of a popular distribution of Linux.



	Redhat		Novell	
	2003	CHG MRQ	2003	CHG MRQ
Revenue	100 %	100 %	100 %	+ 3 %
License			20 %	- 10 %
Maintenance	64 %	+ 70 %	79 %	+ 7 %
Services	34 %	+ 37 %		
Gross Margin	70 %		60 %	
SGA	56 %	+ 8 %	44 %	- 8%
R&D	24 %	+ 29 %	16 %	+ 2 %
Operating Margin	- 1 %		—	
Net Income	- 7 %		- 14 %	

Red Hat, also shown in the table, is rapidly growing its maintenance revenue for the Red Hat Linux distribution. Unlike the Netware software product from Novell, the Red Hat Linux distribution generates no license revenue for Red Hat. But clearly Red Hat maintenance revenue is increasing at a faster rate than Novell maintenance revenue. In the table, Red Hat "subscription" revenue reflects what most companies report as maintenance, as quoted from the Red Hat quarterly report:

> *"The base subscription entitles the end user to one year of maintenance, which entitles the end user to configuration support and updates and upgrades to the tech-*

> *nology, when and if available, during the term of the subscription[s] (sic).*

Below is how Red Hat defined services:

> *"Enterprise technology services are comprised of revenue for enterprise consulting and engineering services, and customer training and education."*

Aside from Novell and Red Hat, there are many other open source segments and markets being addressed using the support strategy. Bitrock for example, delivers and maintains custom and standard installers for applications that run on the popular open source combination known as LAMP (Linux, Apache, MySQL and PHP or Python and Perl).

THE CONSULTING STRATEGY

In an article he wrote in 1999, software industry pundit Clay Shirky said the following:

> *"30 years ago, every IT department in this country was in the business of building custom products, and the software industry grew up around that assumption... Now, open source suggests an almost pure service model, where the basic functionality costs nothing, and all the money is in customization."*

Indeed, an internal McKinsey Consulting study cited in 1999 suggested that enterprise solution fees are 30% license and 70% implementation. According to a 2000 US Department of Commerce report, not since 1962 has packaged software investment reached 30% of total software investment. So Linux or not, software licenses are earning a smaller portion of information technology (IT) investment, while consulting and services continue to rise.

According to Red Hat, the operating system comprises only 4% of the overall revenue of a Linux-based solution. Delivering a customer solution involves integration of hardware, software and maintenance: middleware integration earns high margin consulting fees. With increasing frequency, custom application consulting is performed by system integrators and value-added resellers (VARs), the vendors closest to the customers. These vendors have seen the advantages of open source, making existing VARs and resellers of Microsoft, BEA, and Oracle, prime converts to broad open source-based solutions.

Linux certification programs from Red Hat and Novell greatly reduce the support concerns that customers previously raised about Linux. As Shirky predicts, system integrators like Cognizant see the opportunity to remove nearly all licensing costs from a proposed solution, and create winning bids for customers, at both lower prices and higher margins. Cognizant operates an "open source center of excellence" in India as a resource for its system integration projects around the world.

THE PATRONAGE STRATEGY

Why would a company like IBM, or any company for that matter, contribute time, energy, developers, and code to an open source organization? There are a number of strategic reasons. IBM does it to drive standards adoption and crack entrenched markets. When a company contributes open source software to an independent organization, it anticipates that a de-facto standard and supporting community will converge around that contribution.

A company may also use the patronage strategy to commoditize a particular layer of the software stack, eliminate competitors that are extracting revenue from that layer. For example, IBM, as a major corporate patron of Linux, seeks to commoditize the x86 operating system, eliminating server fees for Microsoft Windows and Sun Solaris. This creates an opportunity for IBM to offer value higher up the stack through clustering, availability, provisioning, security, and management software.

To succeed with a patronage strategy, the patron must deliver more than just source code. They must also assure leadership and consistency. Mozilla is an example of a project that initially failed in this regard.

In January 1998, with 60% of the browser market, Netscape was losing market share to Microsoft. On April 1st, 1998, Netscape publicly released the source code to what ultimately became Mozilla. Clearly Microsoft had picked an easy target. The Mozilla project continued to deliver buggy, late releases, and by January 2004, Microsoft had gained 95% market share, with Mozilla falling to a mere 2% share of the browser market. Contributing software to the open source community alone was not sufficient to save the successor to the Netscape browser.

Another interesting case is the Apache web server. IBM dropped its own offering, which had loyal support within IBM but few synergies outside of the company. At the time, Apache had about 50% of the web server market, and Microsoft was steadily gaining share. By adopting Apache, IBM prevented another Netscape episode, where Microsoft seized control of the browser platform. Apache eventually accelerated in popularity to 70% of the web server market. IBM's patronage strategy successfully prevented a Microsoft monopoly.

When IBM open-sourced all of its Eclipse code with a contribution valued at $40 million, it rearranged the integrated development environment (IDE) landscape. Since Eclipse lets developers target Linux, Java, or Windows, it potentially replaces Sun or Microsoft with a standard cross-development framework in which IBM can better integrate its Rational tools.

Aside from IBM using Eclipse to develop its own software, Eclipse potentially levels the field for IBM across a large development community. By commoditizing the framework, IBM can add value higher up the development tool chain. IBM licenses tools that customers will purchase if integrated well in a development platform. Furthermore, since Eclipse is free, programmers are likely to learn Eclipse as part of their education. Once they are experienced with the Eclipse IDE, they are lifelong prospects for robust software tools from IBM's Rational product line. IBM might have pursued business development through university licensing programs. Instead it made a long-term $40 million dollar investment in open source software available to everyone in computer science and engineering education worldwide.

According to data from the Eclipse community group, there are more than 10,000 download requests per day and more than 450 Eclipse-related projects. From that effort will emerge a stream of prospects for Rational tools from IBM. But, like many other large software providers, IBM must carefully manage the potential open source competitive threat to its software franchises such as Rational, Websphere, DB2, and Notes. In the relatively near future open source will make inroads on those domains, and IBM and other independent software vendors (ISVs) will have to establish other areas to add value.

IBM has been very focused on where it applies its open source energies. The company has an Open Source Steering Committee that has approved many open source initiatives. IBM's open source initiatives are clearly vested in server strategies as opposed to the desktop. As a result of such focus, IBM has succeeded in commoditizing the Sun Solaris operating system and in slowing down Microsoft server adoption in the data center. It has made no headway yet however, in breaking up the Microsoft Office desktop monopoly.

Most major original equipment manufacturers (OEMs) and software providers have adopted the patronage strategy to some extent. Today HP supports more than 60 open source projects that provide tools, utilities, and solutions that make it easier for customers to deploy or customize their products. SGI supports numerous open source projects that are focused upon their high performance computing market.

THE HOSTED STRATEGY

In a January 2004 interview with Java Developer's Journal, Scott McNeely gave the following prediction:

> *"Software licensing and deployment models will be radically simplified. 2003 was the year we saw a bunch of companies finally get the service provider model right. Companies like Salesforce.com, eBay, and Google, are in the software business, but they don't sell their software, they let you use it or rent it."*

Similarly, at the March 2004, Open Source Business Conference, Tim O'Reilly discussed what he called the *"Open Source Paradigm Shift"*, advising companies to look for *"hidden service business models"*. He pointed out examples like "Google and Amazon, who's APIs treat web applications and their data as programmable components".

In looking at open source business strategies, it is apparent that service providers have much to gain from open source. They can use GPL licensed software internally without restriction and without the obligation of sharing their code modifications. This allows them to leverage open source, and incur little or no competitive risk.

The GPL license allows them to own and keep secret the intellectual property modifications they create, and as long as they don't distribute the software, they don't have to publicly share the modifications. Using open source allows them to lower costs, while delivering extremely reliable, enterprise quality services.

For example, in June of 2003, Salesforce.com revealed that they use open source Eclipse and Linux. Salesforce.com, a Software-as-a-Service (SaaS) provider customer relationship management (CRM) applications charges a monthly per-user fee. The SaaS software runs in a browser window, so no software installation is needed by the customer. Netsuite, another SaaS company delivers both financials and CRM applications, making heavy use of open source in its operations.

Consider also Amazon, through which billions of dollars of consumer transactions flow each year. Amazon is a large user of open source. CNET a few years ago discussed Amazon's SEC filing, where Amazon attributed millions of dollars in savings to *"migration to a Linux-based technology platform that utilizes a less costly technology infrastructure."*

Google even more impressively, bootstrapped its keyword advertising business using Linux and commodity servers, saving Google millions in server infrastructure costs. Sergey Brin, co-founder of Google, gave the 2002 keynote presentation at LinuxWorld, describing how Google runs Linux on over 10,000 servers, generating advertising revenue through a search service that is known for speed and relevancy. Google is now rumored to be running over 100,000 Linux servers and laying plans to leverage its server infrastructure in ways that extend far beyond search. Computer World reported in 2002 that financial services companies, often the leaders in IT adoption, were rapidly deploying Linux servers. One major example is E-Trade, a highly successful internet-based banking and securities trading service. What do these companies all have in common? They are hosted service companies using open source as a cornerstone to their IT platforms.

THE EMBEDDED STRATEGY

Linux is the operating system in over half of the embedded systems market. It has been used in consumer products such as TIVO and devices large and small, from servers to cell phones. Throughout the word, it is rapidly becoming the operating system of choice for many low cost communications products.

It is well known that hardware vendors adopting Linux gain advantages in terms of a platform that is functional, extensible, and quickly implemented with minimal capital outlay. A hardware vendor starting a new project should encounter few complications using Linux to get started with design and feasibility testing. And because Linux runs on generic hardware, engineering, prototype, and demonstration hardware costs are a minimum. For a hardware vendor, these advantages free up budgets for potentially better uses in creating value for the customer.

Michael Tiemann, CTO of Red Hat, offered a technical strategy in a May 2002 editorial in Linux Devices. The key, according to Tiemann, is viewing open source as a platform, not merely using Linux as a product to replace a proprietary operating system: *"The fact that Linux can be licensed free of charge... changes the equation almost not at all."*

Hardware vendors should utilize standards and commodities, including Linux, as a platform strategy, and move up the chain by developing software that actually creates value. Set top vendors, for example, might be more viable businesses today if they had pursued truly open platforms and standards. But companies continue to waste their development dollars on software functionality that is otherwise free and available through open source. They persist in buying third-party proprietary platforms or creating their own proprietary development platforms that deliver marginal product differentiation and limited value to customers.

In contrast, Linux and other open source software deliver great value in the embedded market. The inherent technical advantages of Linux for embedded systems include stability, small footprint, and networking. Through the Ipv6 implementation, Linux can address thousands of embedded devices. The Linux kernel is well known for its stability. Linux has relatively low latency, and is generally capable of driving hardware across the embedded device spectrum. Where embedded applications present a real-time performance challenge, the Linux kernel can run as a task under a real-time OS. Linux includes well-documented device drivers. A large support community exists, deeper and potentially more responsive than many proprietary vendors can field. Development tools for embedded Linux are improving.

For example, Linux running on commodity hardware allowed network device developer Neoteris to concentrate on creating value-added software. Neoteris delivered a product with more features, months before the competition and at lower price. The strategy clearly paid off in October 2003, when Netscreen acquired 3-year old Neoteris for $265 million in stock and cash.

Digium produces telephony cards that interoperate with the Asterisk open source PBX. Many telephony product and service vendors integrate Asterisk systems to deliver a broad range of telephony applications to customers.

FINAL THOUGHTS

There are a number of ways for vendors to chart successful open source business strategies. These strategies provide a powerful tool for getting a business on a faster revenue trajectory, for improving value, and for outmaneuvering the competition. Some of the strategies in this discussion parallel traditional commercial software; others invoke new services or methods. Examples like Amazon, Google and Neoteris, demonstrate that Linux and other open source can even help companies that are not strictly in the software business achieve tremendous growth and profitability in a relatively short time.

Business managers should understand open source strategies and determine whether any are useful for their companies to adopt. Investors should consider the strategies here when evaluating companies for their investment portfolios.