Open-Source Software: In Search of a Business Model

Businesses and governments increasingly use open-source software. Part of that growth comes from an expanding base of software that meets their needs. Part comes from cost savings, as proprietary software prices continue to increase at several times producer price indices. Perhaps more important, open-source software firms now provide the same services provided by commercial software firms without the higher costs and license limitations.

The Benefits and Risks of Open-Source Software

Users of software want:

- Stable and reliable software that meets their functional needs. The broader requirements, often unstated, are to improve service and reduce costs through the use of information technology.
- Support if the software fails to do what it should do or, more important, if a software failure interrupts service to customers, suppliers, distributors or firm staff.
- Ability for constant change in function, integration with other internal and external systems, and compliance with changing regulations.

Historically open-source software was developed by a few who were interested in “building” a solution to a problem they had identified. Some of the first open-source software—years before that name—came from colleges and universities and was available to others without documented ownership, licenses, or documentation. The quality of the software was judged on the reputation of the developers themselves. There are early successes. The University of Chicago and North Carolina State University developed statistical packages later known as SPSS and SAS. These subsequently became commercial products of SPSS Inc. (acquired by IBM Corporation in 2009) and SAS Inc.

The differing characteristics of the original open-source software development and what is emerging as a business model is shown in Table 1. These differences are reflected in the various business models that are being adopted.

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Historical Community</th>
<th>Emerging Open Source Business Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Culture</td>
<td>Innovation</td>
<td>Service</td>
</tr>
</tbody>
</table>

Note: Table 1 has 56 words and table 2 has 108 for a total of 164 words included in the word count. 300 words are reserved for the graphic.

1 In the U.S. the rate of price increase in intermediate goods has increased from 6.2% in January year-over-year to 11.6% in July primarily because of the cumulative increase in energy costs. In July the broader Producer Price Index increased to 7.2% over the previous year. Review of the annual cost of software maintenance for proprietary software is between 22 and 25%.

2 Tables 1 and 2 are updated from those in the author’s “Towards Sustainability of Open Source Software in Higher Education,” Chapter 2 in “Sustainability Study: A case study review of open source sustainability models” published by the University of Oxford.
Table 1 – Characteristics of open source software developed historically and developed by an emerging open-source software firm.

The preferences of U.S. firms for software license terms, shown in Figure 1, give an edge to open-source software because of perpetual rights to the software. Commercial software firms are hesitant to provide perpetual usage rights, since this jeopardizes their annual maintenance revenue. Users want perpetual licenses so a software vendor cannot terminate their use of the software.

In summary, there are many software development projects. A few become successful software products. The business models begin to emulate commercial software firms. Open-source software has an advantage—lower or no cost of marketing.

Figure 1 – Licensing terms preferred by U.S. corporate users

<table>
<thead>
<tr>
<th>Value proposition</th>
<th>Features</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>User trade-off</td>
<td>Functionality</td>
<td>Reliability</td>
</tr>
<tr>
<td>Focus</td>
<td>Technology</td>
<td>Use</td>
</tr>
<tr>
<td>Success</td>
<td>Peer recognition</td>
<td>Market Share</td>
</tr>
<tr>
<td>Documentation</td>
<td>Wiki entries, code comments</td>
<td>Professional book, eBook, and supplier documents</td>
</tr>
<tr>
<td>Software releases</td>
<td>Whenever</td>
<td>Predictable</td>
</tr>
<tr>
<td>Quality</td>
<td>Acceptable</td>
<td>Fully tested</td>
</tr>
<tr>
<td>Support</td>
<td>Developer lists</td>
<td>24/7 telephone, email, instant messaging, on-site consultants</td>
</tr>
</tbody>
</table>

Table 1 – Characteristics of open source software developed historically and developed by an emerging open-source software firm.
A recent Forrester-Dr. Dobb’s Developer survey reports; “Management is aware of the use of open source and, based on their experience, now 74% of software developers use open source to accelerate projects; 71% to reduce costs.” The report continues: “There's so much agita [anxiety] over commercial software licensing that we had to ask in our survey about the open source alternative. It's a measure of how far open source has come that 41% of respondents don't have a preference either way.”

Open-source software often has a cost advantage. Open source does marketing by providing materials, including code and online demonstrations, for potential users to evaluate the software rather than marketing staff. Potential users usually become aware of the open-source software product through comments of others in the same industry and from references in trade journals, not from advertising.

Assigning market costs only of the original software license, marketing cost is estimated to be 60% of the original software license. For commercial software firms the renewal rate for annual maintenance varies from 90% to 98% with insignificant cost of sales. The primary source of revenue is annual maintenance fees, not the original, frequently discounted, software license.

Some commercial software firms are using annual licenses so failing to pay the annual fee terminates use of the software.

Comparisons of the earlier open source software projects and the emerging open source business are summarized in Table 2.

---

3 The Forrester-Dr. Dobb's Developer Technographics Survey taken in the second quarter 2011 was reported in the September 5, 2011 issue of Information Week.
4 These data are based on Oracle Corporation and have been consistent for several years. Other public firms have similar results.
Processes | Historical Community | Emerging Open Source Business Models
---|---|---
Scope of features and functions | Participant defined | Industry criteria for ‘purchase’ and advisory committees
Documentation | Limited | Professional (such as an O’Reilly publication
Technical support | Community with developers | 24/7 telephone and e-mail support, tiered escalation, user-accessible database of errors, code corrections, configurations, and work-arounds
Training | Informal | Formal training programmes with professional materials; often on-line eLearning
Evolution of the software | Participant contributions | Software development teams following a software development methodology and defined architecture
User organisation | Participant meetings and conferences | Formal and sometimes subsidised user meetings
Marketing | Informal by participants | Community representatives; limited marketing materials, on-line demonstrations and code available to install and test

Table 2 Comparative business processes for early project and emerging open source products.

**Open Source Software, enforceable “not for sale”**

MIT Professor Richard Stallman believes open-source software should never be used in commercial software or sold. In his March 1985 GNU Manifesto he wrote:

> Arrangements to make people pay for using a program, including licensing of copies, always incur a tremendous cost to society through the cumbersome mechanisms necessary to figure out how much (that is, which programs) a person must pay for. And only a police state can force everyone to obey them.

The GNU Public License contains one clause that creates limitations on derivative works:

> You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

Stallman’s “business model” is individuals creating open-source software. It continues to be a practice in universities. If a derivative work is created then it must be licensed under the terms and conditions of the GNU Public License—that is, no charge for a license.

This is a risk for subsequent users who either modify a program that includes GNU licensed computer code or wish to sell the resulting software as a commercial product.

**Competition and cooperation**
The Apache web server is one of the best known open-source products. It is used in 58% of U.S. firms as compared to a market share of 6% for IBM’s Websphere and 5% for Oracle’s Weblogic servers. Worldwide Apache has a market share of 60% for the 255 million Internet servers.

Apache began as a reference installation for HTTP (Hypertext transfer Protocol) that makes the Web work communicating web pages between servers and browsers. Eight software developers began work on the Apache server in 1994; three years after the Internet Engineering Task Force had first defined the protocol. The Apache Foundation was organized in 1999 to continue this work and support similar projects.

IBM and other firms developing web servers elected to use the Apache HTTP server and differentiate by additional functions and integration with other products. This saves software development cost and facilitates interoperability. IBM spent a year documenting who owned the pieces of the software. IBM obtained contribution agreements with both the software developers and, when needed, their employers. The resulting Apache license permitted commercial use of the software and any derivative works.

In this business model, participants share development cost. Some do not contribute but use the software, but the cost savings still exceeds any value to the competition. According to early surveys between 80 and 90% of the software developers were employed and assigned to the development of open-source software. That practice began to decline in 2008 as firms restricted funding only to open-source software projects had immediate value for them.

**Buying a market – IBM leads again**

IBM had a multi-language software environment used by programmers called VisualAge. It was one of the leading products in this area. IBM Canada set up a non-profit organization called the Eclipse Foundation in 2004 and donated the code for further community development.

Eclipse was architected to support “plug-in” software from a number of different firms. In a 2004 briefing Aldo Eisma, IBM Corporation, observed Eclipse “Partners want Open Source less dependent on IBM [and] freedom of action for partners [to] complement IBM products [and to] implement their own products.” (He also observed for 30% of open source developers “Open Source is their job!”) He said Eclipse would generate revenue for IBM from complementary products.” Because of interoperability, customers of all of the vendors became potential customers for complementary IBM products. Thus the market for these complementary expands without a marketing effort.

**Providing software support**

In 1993 Bob Young incorporated the ACC Corporation, a catalogue business that sold Linux operating system. In 1995 two companies merged and became Red Hat Software. Originally Red Hat sold the open-source Linux software in bookstores packaged with a user manual. Users could purchase telephone support to assist in installation and use of the software. The availability of downloaded software ended package sales, but support was expanded with different service levels—how many hours Red Hat had to respond to a support request. Both the desktop and server versions of Linux were supported.

Red Hat has acquired 20 small software firms. This extends the complementary products and revenue, and differentiates Red Hat from other sources of the open-source Linux software.
Wikipedia succinctly describes the Red Hat business model: “Red Hat partly operates on a professional open-source business model based on open code development within a community, professional quality assurance and subscription-based customer support.” In 2007 Red Hat reached a marketing agreement with other open-source companies. It resells their software via an Amazon Marketplace-like distribution portal retaining the original branding.

Red Hat is less than 1% the size of Oracle. Oracle’s profit margin is 24%, Red Hat’s is 12%. Yet the Oracle price-earnings ratio is 16 and Red Hat is 62. This suggests investors see more potential in Red Hat than they do in Oracle.5

The support model has become the most frequent business model for software.

Insurance as a product?

In an investors’ call several years ago the Red Hat CFO was questioned on the amount of support that is provided. In the answers she gave, analysis shows the average customer makes one telephone support call—not one incident—each three years. A typical support call costs about $25. This yields an estimated gross operating margin of 98%. Red Hat’s combination of quality control and service level pricing and support delivery provides high profit that can be used for development of complementary software, acquisition of software firms, and funding of the Fedora open source community. From the users’ perspective it is the ability to obtain immediate technical support, not only the support itself, that is valued.

Insurance differs from the typical support model. Insurance is successful by distributing software without errors, a well-managed and well-trained support staff that can respond quickly, and a pricing model that matches the needs of the various types of users. Under those conditions it becomes very profitable.

Software goes commercial

Can open-source software be commercialized? Yes. Some software firms as MySQL AB have two database management system products—the open-source product available free and another with few additional features and performance improvement. The open-source Community Server version was licensed under the GNU General Public License which prevents its use in any derivative work requiring license payment. The commercial Enterprise Server software has, compared to Oracle, a very reasonable cost.

MySQL was purchased by Sun Microsystems, and in turn, by Oracle Corporation. Current prices for the commercial version vary from USD 2,000 to USD 10,000 per year. Although rumoured, Oracle has not yet increased MySQL pricing.

Patents and open source software

There have been claims, typically by non-practicing entities [NPEs, sometimes referred to as ‘patent trolls’] that open-source software violates patents. The Open Invention Network patent pool organized to defend Linux weakened when Oracle and Google left the coalition with their patents. In 2008 Red Hat settled with an NPE that included licensing for users of Red Hat products. Red Hat paid USD 4.2 million. It settled with another NPE in October 2010 for an undisclosed sum.

---

5 Financial statistics are from Forbes as of 9 September 2011.
Red Hat can continue to offer “patent safety for developers, distributors and users of open-source software” for alleged patent infringement only if the settlements are an acceptable business risk. Florian Muller, FOSS Patents, commented on 2008 settlement with FireStar Software via DataTern: “Red Hat feeds the patent trolls.” He implied settlement encourages other NPEs to seek licence fees.

Most commercial software firms offer patent indemnity. Most open source firms do not.

What does this mean to a user of open-source software?

- Licensing – be certain whether the license permits use in a commercial derivative work or not. Check current software to ensure there are no violations of software licenses.
- Support – include an evaluation of software support firms with each use of open-source software. This evaluation should also analyze the long-term sustainability of the firm.
- Acquisition – think through the impact of the acquisition of the software project or firm on the future availability, support, and costs of the software.
- Patents – Evaluate the risk.

Open-source software has additional risk, but does offer unlimited control of the software and lower costs.