



***Advanced Housing Technologies, LLC***

**AHT-POSITION-0050  
Open Source Policy Paper**

**CONFIDENTIAL**

# Table of Contents

<b>WHAT IS OPEN SOURCE SOFTWARE? .....</b>	<b>4</b>
<b>ADVANTAGES OF OPEN SOURCE SOFTWARE.....</b>	<b>5</b>
Open Source Has Advantages for Software Users .....	5
Free is Not Free .....	5
Valuing Independence .....	6
Customization and Freedom .....	8
<b>OPEN SOURCE FOR AHT’S ENTERPRISE COMPUTING NEEDS .....</b>	<b>9</b>
<b>HOW OPEN SOURCE SOFTWARE DEVELOPMENT WORKS.....</b>	<b>10</b>
Licenses (The Foundation of Open Source) .....	10
Close as Appropriate — the Mixed Model.....	12
Getting Help from a Community.....	13
The Need for Committers – Creating Opportunities for Success .....	14
<b>The Employee Perspective .....</b>	<b>16</b>
Benefits of Being a Committer .....	16
How to Become a Committer .....	16
<b>THE CASE FOR SUSTAINABILITY .....</b>	<b>18</b>
<b>ABOUT COMPETITION.....</b>	<b>19</b>
<b>THE EFFECT OF OPEN SOURCE ON AHT’S BOTTOM LINE.....</b>	<b>19</b>
Financial Sources as a Distributor.....	19
<b>Fiscal Advantages of Open Source Development for AHT as a System Integrator .....</b>	<b>20</b>
Solutions Demand Curve.....	20
Enhanced Business Growth Potential.....	22
Generating Software Profits.....	23
<b>PERCEIVED DISADVANTAGES OF OPEN SOURCE MODELS.....</b>	<b>24</b>
The Legal Challenges.....	24

**THE JURY IS OUT..... 25**

**CONCLUSIONS..... 27**

## What is Open Source Software?

The main features that characterize free (open source) software is the freedom that users have to:

- Use the software as they wish, for whatever they wish, on as many computers as they wish, in any technically appropriate situation.
- Have the software at their disposal to fit it to their needs. Of course, this includes improving it, fixing its bugs, augmenting its functionality, and studying its operation.
- Redistribute the software to other users, who could themselves use it according to their own needs. This redistribution can be done for free, or at a charge, not fixed beforehand.

It is important now to make clear that we are talking about freedom, and not obligation. That is, users of an open source program can modify it, if they feel it is appropriate. But in any case, they are not forced to do so. In the same way, they can redistribute it, but in general, they are not forced to do so.

To satisfy those previous conditions, there is a fourth characteristic which is basic, and is necessarily derived from the others:

- Users of a piece of software must have access to its source code.

The source code of a program, usually written in a high level programming language, is absolutely necessary to be able to understand its functionality, to modify it and to improve it. If programmers have access to the source code of a program, they can study it, get knowledge of all its details, and work with it as the original author would.

Paradoxically, if this freedom is to be guaranteed for a given piece of software, with current legislation, it is necessary to “protect” it with a license which impose certain restrictions on the way that it can be used and distributed. This fact causes some controversy in certain circles, because it is considered that these licenses make the software distributed under them “less free”. Another view, more pragmatic, is that software will be made more free by guaranteeing the perpetuation of these freedoms for all its users. Because of that, people holding this view maintain that it is necessary to limit the ways of use and distribution. Depending on the ideas and goals of the authors of a piece of code, they can decide to protect it with several different licenses.

## Advantages of Open Source Software

Computer Economics (CE) conducted a survey of visitors to its website in May of 2005 regarding the perceived advantages in the use of open source software.<sup>1</sup> Although not a scientific sample, the results were interesting.

As nearly everyone knows, open source software is a low cost alternative to proprietary software. For example, the open source Linux operating system is commonly seen as a low cost alternative to Microsoft's Server 2003 operating system, or Sun's version of Unix. The popularity of open source is seen in the fact that today the largest market share for web servers is held by the open source Apache system.

One might think, therefore, that the key advantage of open source software is its low cost of ownership. But visitors to the Computer Economics website didn't think so.

### *Open Source Has Advantages for Software Users*

The results of the survey are shown in the chart below.

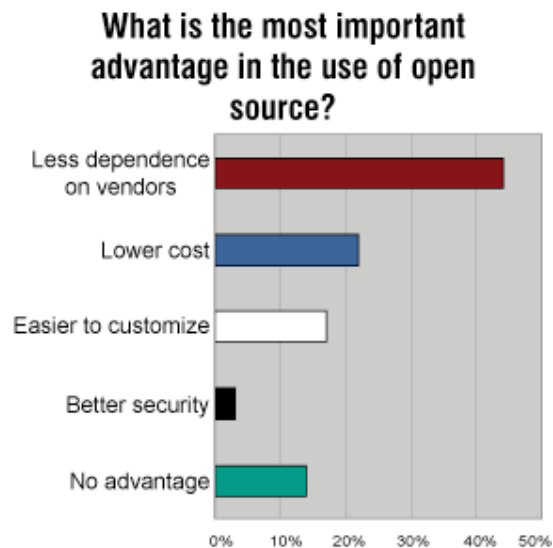


Figure 1. Computer Economics Survey Chart

### **Free is Not Free**

So, what is the top advantage of open source? The leading vote-getter was “reduced dependence on software vendors” at 44%, followed by “lower total cost of ownership” at 22%. Although these were the top two vote-getters, it is significant that respondents valued reduced dependence on software vendors by a two-to-one margin over lower cost.

<sup>1</sup> <http://www.computereconomics.com/article.cfm?id=1043>

The second place ranking for “lower cost” indicates that IT decision makers recognize that open source software is not really free. With most types of software, administration and support costs overshadow initial software license cost and annual maintenance fees — the costs that are minimized by open source. Therefore, software buyers do not see the low or zero initial cost of open source as its most important advantage.

Whether open source software is less costly to administer than proprietary software depends largely on a ready pool of resources trained on the system, the availability of administration tools that allow system administrators to manage a greater number of systems, and the number of version upgrades and patches that are issued by the developer. In this regard, open source software may have little if any advantage over proprietary software, although the situation varies from application to application. Therefore, low cost, although important, is not the key advantage of open source.

## ***Valuing Independence***

The CE survey cited above indicated that IT decision-makers value “reduced dependence on software vendors” as the most important advantage of open source. This indicates that software buyers must feel some level of dependence on proprietary software vendors, from which they desire freedom. Such dependence includes reliance on the vendor for maintenance and support and the necessity for the buyer to accept version upgrades that the buyer may not need or want.

For example, when Microsoft announces a new version of its Windows Server operating system, it invariably phases out support for older versions of the system. Users that are satisfied with older versions of Windows will be eventually forced to upgrade if they want to continue receiving vendor support. In contrast, there is no forced upgrade cycle with open source. Older versions of open source products continue to be supported through the open source community and third party support providers as long as there is demand in the marketplace for such support.

The issue of independence is very important.<sup>2</sup> When no one holds exclusive rights on a given code (sometimes mentioned as “life or death rights”), several traditional problems of the proprietary software model are overcome:

- *There is no one with the power to restrict, in a unilateral way, how the software is used, even retroactively.* Such a power manifests, for instance, when a proprietary software vendor decides not to upgrade a software product for some old platform. In this case, customers can only stick to the old version of the software, or switch to another product. If open source software is used, customers can also fund some development for the desired platform, or look for other vendors to provide the upgrades (of the very same product).

---

<sup>2</sup> “Free Software / Open Source: Information Society Opportunities for Europe?”, Jesus M. Gonzalez-Barahona – April 2000, [http://eu.conecta.it/paper/Advantages\\_open\\_source\\_soft.html](http://eu.conecta.it/paper/Advantages_open_source_soft.html)

- *There is no single entity on which the future of the software depends.* This is a very common concern with proprietary software. Let us say that a company uses a software product, and relies on the software manufacturer for upgrades and continued development. If the software manufacturer closes doors, or decides to discontinue development of the product, no one has the right to take the program and continue development on it, effectively killing its usability in the market. This has happened many times, and this problem is amplified by the recent mergers in the software market, that usually lead to “cannibalization” of some software product to allow just one or two to get to the market. Open source software effectively protects against this, because if the group or company that originated the code decides to stop development, it is always possible to fund another software group to continue the maintenance and improvement, without legal nor practical limitations.
- *No “black boxes” are possible.* This point is so important that open source is now considered by many experts as one of the necessary conditions for dependable applications. There are several reasons for this importance. One of them is related to the dependability of the services provided by a given piece of software. By having the source code available, it is possible to perform a thorough inspection and verify the correctness of the algorithm and the implementation scheme used. This is also possible in part even with closed source or nearly free licenses. The difference lies in the fact that users are allowed to modify everything they find appropriate to suit their needs. A glaring example is the Linux kernel and its “international patches”, a set of enhancements with legal problems in some countries. These patches include support for encrypted communications, and can be legally used in large parts of the world. The patches have being developed by concerned parties in countries where such a development is allowed, and therefore users in those countries can use those enhancements. With binary-only products no inspection is possible, with closed source or nearly free licenses inspection is possible, but modifications are forbidden, so the inherent advantage of having source code available is rendered ineffective.
- *There is always the possibility of “forking”, or creating an alternative code base if the current one is in some way perceived as wrongly managed.* Forking introduces several levels of competition within the model. For instance, before forking, several programmers can work harder to keep everybody happy integrating as many well-engineered features as possible, to prevent a fork by people whose needs are not addressed. After a fork, both branches tend to compete for the user base with very similar products: only good quality and quick improvement can maintain them in the market.
- *No per-copy fees can be asked for modified versions, and anyone can use the current code base to start new projects.* Working knowledge can be gathered at a minimal cost. This is what made Internet software systems such an important factor in the new economy: students, and people trying new technologies were able to integrate and adopt them immediately, without the hurdles of commercial

or non-disclosure license agreements. In addition, the right to freely modify them allowed for the incredible expansion in the number of communication protocols and systems, each perfectly tailored to the needs of their users. This is also a reason for the overwhelming success of the Linux kernel, widely employed by students thanks to its near-zero cost, and subsequently used by the same students in the startups originated by them, when they turn into entrepreneurs after leaving University.

- *There are fewer conflicting priorities due to marketing pressures.* This is a simple consequence of the fact that there is no single commercial entity pushing for precise delivery dates or features that must be supported. Usually open source software is delivered “when it is ready”, and when the development team feels that its quality is good enough. This means that software usually does not need as many “service packs”, updates and such, reducing the maintenance cost.

The CE survey indicated that vendors of proprietary software are missing the mark when they argue that open source software has a higher total cost of ownership, is less secure, or higher risk in terms of ongoing support. These factors, although important, are not the key concern of software buyers. The key appeal of open source software is that it avoids vendor lock-in and gives buyers the freedom to choose what to do and when to do it.

For software buyers, the best strategy is to consider mature and established open source products as well as proprietary software products that adhere to open standards. In this way, buyers can choose the best software product for the job, knowing that the value of their investment will be preserved without locking the organization in to a single vendor solution.

## ***Customization and Freedom***

Nearly 18% of the CE respondents reported that customization was important to them. Although not immediately clear to many (who never look under the hood of their computers), the availability of the source code and the right to modify it is very important. It enables the unlimited tuning and improvement of a software product. It also makes it possible to port the code to new hardware, to adapt it to changing conditions, and to reach a detailed understanding of how the system works. This is why many experts are reaching the conclusion that to really extend the lifetime of an application, it must be available in source form. In fact, no binary-only application more than 10 years old now survives in unmodified form, while several open source software systems from the 1980s are still in widespread use (although in many cases conveniently adapted to new environments). Source code availability also makes it much easier to isolate bugs, and (for a programmer) to fix them.

## Open Source for AHT's Enterprise Computing Needs

Put bluntly, it makes no sense for AHT to feed the Microsoft or Apple empires hundreds of thousands dollars for business software that is free, and is just as capable for essential computing needs (email, word processing, page layout, spreadsheets, and graphics). It also does not make sense to pay premium prices for more hardware than is necessary. In fact, computers running various versions of Microsoft operating systems need beefed up hardware to mask the inefficiencies of Microsoft operating systems. Much less expensive Intel computers run just as efficiently with better, more stable Linux operating systems for most applications (and the peripherals they use).

Reflecting the savings possible by adopting open source software, France's Gendarmerie Nationale, the national police force, says that it has saved millions of dollars by migrating to various open source solutions.<sup>3</sup> As part of its pilot phase, the organization has already migrated to Ubuntu Linux on some 5,000 workstations. Plans are in place to switch a total of 15,000 workstations to Linux before the year (2009) is up, and all of its computers by 2015<sup>4</sup>.

With some 90,000 workstations, the Gendermerie Nationale is hardly a small business. Indeed, a cursory examination of the transition reveals that the savings come from the following areas:

- Productivity software: From Microsoft Office to OpenOffice.org
- E-mail client: Using Thunderbird now instead of a proprietary client like Microsoft Outlook
- Operating system: From Windows to Linux

It is important to note that Linux, in itself, is not always the magic bullet for every organization's software licensing woes. Most computer vendors do not offer discounts for selling computers not already bundled with Microsoft's Windows operating system. However, consider that companies like System76 sells desktop computers loaded with the free Ubuntu operating system, and the astounding suite of free applications that go with it starting at \$249 (not counting monitor or keyboard). The same company sells similarly capable laptops starting at only \$739.<sup>5</sup> Just for fun, you might want to cruise on over to the websites of such companies as Eight Virtues<sup>6</sup>, Dell<sup>7</sup>, and Logic Supply<sup>8</sup> to get a first-

---

<sup>3</sup> "Do a Partial Switch to Open Source Solutions" IT Business Edge – Posted by Paul Mah, 3/12/2009 - <http://www.itbusinessedge.com/cm/blogs/mah/do-a-partial-switch-to-open-source-solutions/?cs=31059>

<sup>4</sup> "French Gendarmerie Ditch Windows for Ubuntu" IT Business Edge – Posted by Lora Bentley - <http://www.itbusinessedge.com/cm/blogs/bentley/french-gendarmerie-ditch-windows-for-ubuntu/?cs=14988>

<sup>5</sup> <http://www.system76.com/?gclid=CJjnoMjd4JkCFRINDQodpgGqVA>

<sup>6</sup> <http://sales.eightvirtues.com/models.html?gclid=CNDLsNbe4JkCFSQMDQodEAjKWg>

<sup>7</sup> <http://desktoplinux.com/news/NS7159701171.html>,

<sup>8</sup> [http://www.dell.com/content/topics/segtopic.aspx/linux\\_3x?c=us&cs=19&l=en&s=dhs](http://www.dell.com/content/topics/segtopic.aspx/linux_3x?c=us&cs=19&l=en&s=dhs)

[http://www.logicsupply.com/system\\_solutions?referrer=googleAd&kw=linux&nw=content&cr=813024865&pl=desktoplinuxathome.com&gclid=COGU4OXf4JkCFR7yDAodrBCAXQ](http://www.logicsupply.com/system_solutions?referrer=googleAd&kw=linux&nw=content&cr=813024865&pl=desktoplinuxathome.com&gclid=COGU4OXf4JkCFR7yDAodrBCAXQ)

hand look at the open source revolution. Then compare the cost/benefits of Intel-based computers running the various other operating systems in the market.

Of course, there are many circumstances where a particular combination of software and hardware is necessary, and the solution is not one easily offered by open source vendors. For example, some engineering software only runs on the Microsoft OS, while some graphics software is only available on the Mac OS (or just runs better in either OS). So, AHT will always need to employ a mixture of operating systems and hardware to meet specific needs.

Given the ability of some Linux distributions, such as Ubuntu, to coexist on the same computer running either Windows or Mac OS, the cost advantages and benefits of open source operating systems are apparent.<sup>9</sup>

## How Open Source Software Development Works

### *Licenses (The Foundation of Open Source)*

With the current legal framework, the license under which a program is distributed defines exactly the rights which its users have over it. For instance, in most proprietary programs the license withdraws the rights of copying, modification, lending, renting, use in several machines, etc. In fact, licenses usually specify that the proprietor of the program is the company which publishes it.

In the world of open source software, the license under which a program is distributed is also of paramount importance. Usually, the conditions specified in licenses of open source software are the result of a compromise between several goals, which are in some sense contradictory. Among them, the following can be cited:

- Guarantee some basic freedoms (redistribution, modification, use) to the users.
- Ensure some conditions imposed by the authors (citation of the author in derived works, for instance).
- Guarantee that derived works are also open source software.

Authors can choose to protect their software with different licenses according to the degree with which they want to fulfill these goals. In fact, authors can (if they desire) distribute their software with different licenses through different channels (and prices)<sup>10</sup>. Therefore, the author of a program usually chooses very carefully the license under which

---

<sup>9</sup> Ubuntu, which can be installed on almost any Intel computer, takes such little memory that a complete bootable system can be stored on a 1GB thumb drive. Ubuntu offers three ways to install onto computers: 1) wiping the contents of the hard drive completely and having only Ubuntu as the OS, 2) as a “virtual machine” that runs as an application under Windows or Mac OS, and 3) in a “Dual Boot” mode that allows the user to choose which operating system to run when the computer is turned on.

<sup>10</sup> For instance, there are authors that decide to distribute their software under a license which guarantees the derived works will be open source software too. However, in some cases, they sell that same software to companies interested in keeping derived works proprietary under a different license which allows it.

it will be distributed. And users, especially those who redistribute or modify the software, have to carefully study its license.

Fortunately, although each author could use a different license for her programs, the fact is that almost all open source software uses one of the common licenses (GPL, LGPL, Artistic, BSD-like, MPL, etc.), sometimes with slight variations. To simplify things even more, some organizations have appeared recently which define which characteristics a software license should have to qualify as an open source software license. Amongst them, the two most widely known are the Debian Project, which defines the Debian Free Software Guidelines, and the Open Source Initiative (OSI), which defines “open source” licenses<sup>11</sup>, and is based on the DFSG. The GNU Project also provides its own definition of free software.<sup>12</sup>

It is easy to see from the DFSG that price or availability of source code in itself is not enough to characterize a product as “open source software”. The significant point lies in the rights given to the community, to freely modify and redistribute the code or modifications of them, with only the restriction that these rights must be given to all and must be non-revocable.

Some of the more common open source software licenses are as follows:

- BSD (Berkeley Software Distribution). The BSD license covers, among other software, the BSD (Berkeley Software Distribution) releases. It is a good example of a “permissive” license, which imposes almost no conditions on what a user can do with the software, including charging clients for binary distributions, with no obligation to include source code. In summary, redistributors can do almost anything with the software, including using it for proprietary products. The authors only want their work to be recognized. In some sense, this restriction ensures a certain amount of “free marketing” (in the sense that it does not cost money). It is important to notice that this kind of license does not include any restriction oriented towards guaranteeing that derived works remain open source.
- GPL (GNU General Public License). This is the license under which the software of the GNU project is distributed. However, today we can find a great deal of software unrelated to the GNU project, but nevertheless distributed under GPL (a notable example is the Linux kernel). The GPL was carefully designed to promote the production of more free software, and because of that it explicitly forbids some actions on the software, which could lead to the integration of GPLed software in proprietary programs. The GPL is based on the international legislation on copyright<sup>13</sup>, which ensures its enforceability. The main characteristics of the GPL are the following: it allows binary redistribution, but

---

<sup>11</sup> The Open Source Initiative. <http://www.opensource.org/osd.html>

<sup>12</sup> Free Software Foundation. <http://www.gnu.org/philosophy/free-sw.html>

<sup>13</sup> This is the case of all software licenses, including open source licenses. However, the GPL makes an interesting use of this legislation, since it is here used to promote the distribution of software which guarantees far more freedom to the user than usual copyrighted works do. Therefore, some times the software covered by GPL is said to be “copylefted”, an interesting word game.

only if source code availability is also guaranteed; it allows source redistribution (and enforces it in case of binary distribution); it allows modification without restrictions (if the derived work is also covered by GPL); and complete integration with other software is only possible if that other software is also covered by GPL. This is not the case with LGPL (GNU Lesser General Public License), also used in the GNU project, which allows for integration with almost any kind of software, including proprietary software.

- MPL (Mozilla Public License). This is the license made by Netscape to distribute the code of Mozilla, the new version of its network navigator. It is in many respects similar to the GPL, but perhaps more “enterprise oriented”.
- Other well-known licenses are the Qt license (written by Troll-Tech, the authors of the Qt library), the Artistic license (one of the licenses under which Perl is distributed), and the X Consortium license.

### ***Close as Appropriate — the Mixed Model***

The Free and Open Source Software (FOSS) model allows for any user to view and modify a product's source code. Organizations and individuals that adhere to this model, such as Canonical Ltd. and the Mozilla Foundation, believe that the benefit that they gain from improvements to their software provided by the community of software developers is more important than protecting their competitive advantage. Common advantages cited by proponents for having such a structure are expressed in terms of trust, acceptance, teamwork and quality.<sup>14</sup>

Most FOSS is licensed under what is often termed a “copyleft” license, a term which emphasizes the license's reversal of the principles of copyright. As stated above, a traditional license is used to limit freedoms, which the free software movement considers essential. By contrast a copyleft license protects the “four software freedoms” by explicitly granting them and then explicitly prohibiting anyone to strip them away when redistributing the package or reusing the code in it to make derivative works. Some licenses grant the four software freedoms but allow redistributors to remove them if they wish.

FOSS can, and has been commercialized, both by purely FOSS companies such as Red Hat and more traditional software companies such as IBM and Novell. The archetypal example is the Linux operating system.

AHT will make most of its software available as FOSS. This will be especially true of those software products and processes that computing enthusiasts, competitors, and the public interacts with (for example, our House Automation software or protocols to connect sensors to computers). Other software, hardware, and firmware that is more IP-

---

<sup>14</sup> The GNU Manifesto - GNU Project - Free Software Foundation (FSF)

related, will be developed as systems proprietary to AHT. That is where our patent efforts will be centered. Why attempt to patent systems that can be easily replicated by others?

## ***Getting Help from a Community***

Open source projects are built and maintained by a network of volunteer programmers and engineers. Prime examples of open source products are the Apache HTTP Server, the internet address system Internet Protocol, and the internet browser Mozilla Firefox.

So, why do so many people contribute to open source initiatives? The following excerpt from a weblog<sup>15</sup> helps to answer the question:

I noted down the motivational factors for developing open source applications. Let's go through the list.

- *Passion.* Remember that most of you are working in this field because you like it. Sometimes developers get to work in a company and in a project that they actually don't like. Open source is their chance to do what they like. So, even though this takes extra time and effort, the moral satisfaction pays it off. Keep the fire burning with Open Source!
- *Educational.* Working in an open source is the best way to keep you up-to-date with the latest technologies. Due to economical reasons, the projects developed in closed profit-based organizations, are not up-to-date with the latest technologies. Keep learning with Open Source!
- *Portfolio.* What you developed as open source could be a very good showcase to obtain a better job. If you don't have yet industry practical experience, this can compensate. Show off your Open Source!
- *Status.* There is a special pride of being a member in a successful open source community. Be proud with your Open Source!
- *Need.* This is actually one of the main reasons for developing open source applications. Either there isn't a software solution for your problem or it is too expensive, you can always make your own or contribute to other people effort to do it. Use Open Source!
- *Influence.* This is somehow related to need. Because a company is needing some features in an open source application, they encourage their employees to take an active role in the development of it. In this way they can easily influence the direction in which the open source project is heading. Influence your Open Source!
- *Altruism/Knowledge sharing.* Some people simply do it because it is the right and moral thing to do. They used and enjoyed open source and now it's time to give something back. Share by Open Source!
- *Quality.* I know that it may sound like a paradox, but I really think that a successful open source project has a higher quality than a successful closed, private one. Usually an open source project has a more varied and wide pool of users. It means that it also has a wider pool of critics. Moreover, the commercial products will have to constantly improve because of the competition. Improve Open Source!

---

<sup>15</sup> <http://beradrian.wordpress.com/2008/11/27/motivation-in-open-source>

- *Economical.* Open source is not entirely free. And here we have to talk about cost reduction and profit.
  - *Cost reduction.* A company can develop a project, but due to the lack of resources, they decide to make it open source. If the project is not on their portfolio, but it was merely developed as a library for other projects or to support the internal infrastructure, this is a very good choice. Their business won't be affected, but their winnings could be huge. In the first place, they win a huge amount of users, which are actually free ad-hoc testers. As the project is growing they can also gain developers or other specialists, thus tremendously decreasing the maintenance costs.
  - *Profit.* A new business model has lately emerged. The organization is offering the product for free, but they are charging for related services, such as support, development on request etc. More and more companies are moving from a product oriented business model to a service oriented business model. Win with Open Source!

### ***The Need for Committers – Creating Opportunities for Success***<sup>16</sup>

An employee's position in the open source project is another key part of his or her value to a firm. The organizational setup varies between open source projects, but in some form, people always play user, contributor, and committer roles. Users use the software, contributors contribute in some form, and committers decide what contributions to accept into the project.

Figure 2 illustrates how a developer might progress through the ranks of AHT's community open source project: A committer typically promotes a user to a contributor role implicitly by accepting the user's contribution into the software. A contributor is typically promoted to a committer position explicitly, through a prior vote of the existing group of committers and a subsequent public announcement of the contributor's ascension to committer status.

---

<sup>16</sup> This section is liberally taken from "The Economic Motivation of Open Source Software: Stakeholder Perspectives", Dirk Riehle, 2007 - <http://www.riehle.org/computer-science/research/2007/computer-2007-article.html>

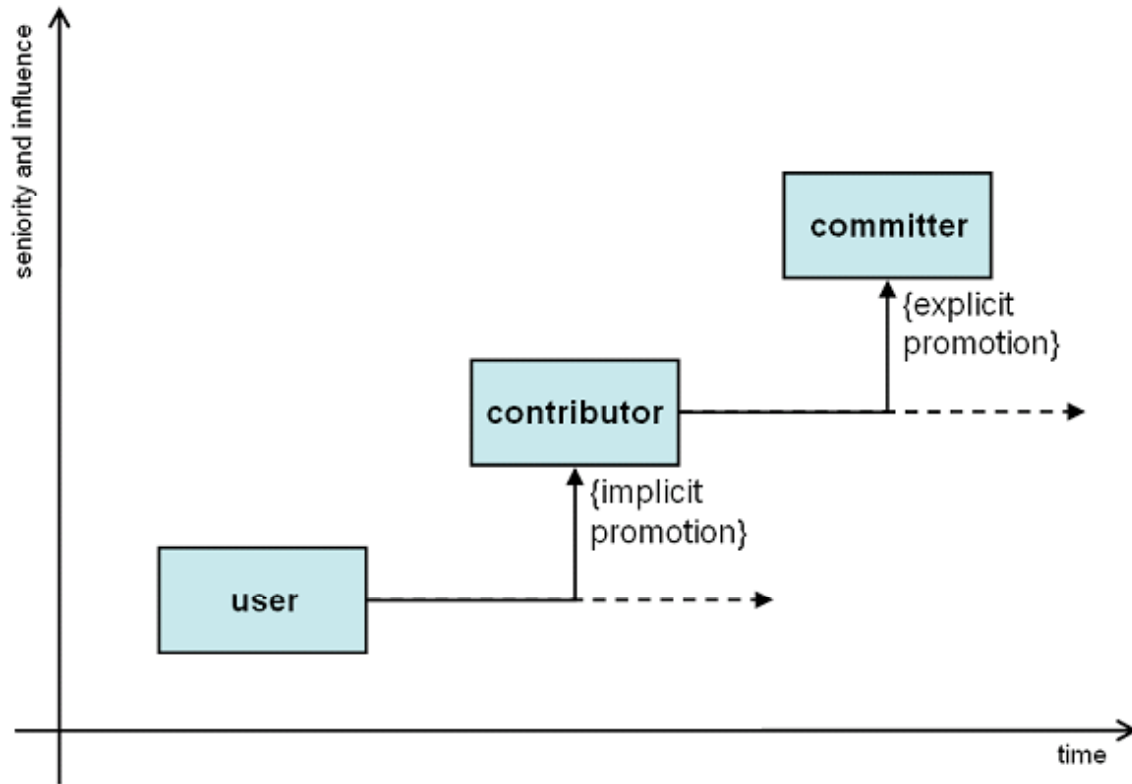


Figure 2: Positions and promotions. In open source projects, users are implicitly promoted to contributor status, while contributors are explicitly promoted to committer status.

Committers determine where the open source project is headed, strategically and on a day-to-day basis. They can typically resolve technical problems faster than non-committers, and have high visibility to the user community. Most projects are set up so there's only a small inner circle of committers, a larger set of contributors, and an even larger user community.

For AHT, the value of employing a committer is manifold. Through the committer, AHT

- gets problems with the open source software fixed faster and better,
- can better align company strategy with the open source project and vice versa,
- appears as a more attractive employer than competitors who don't employ a committer, and
- has higher visibility with the user community and can reach out more effectively.

A major goal of any open source service company is to convert freeloading users into paying customers. A committer's visibility with the user community is an important marketing advantage that an employer can use to support this goal.

Thus, committers have a strong negotiation position with their employers. Employing a committer is important for a first-level support and implementation services company, and it's critical for a second-level support service company.

## ***The Employee Perspective***

Open source software and service businesses make life more complicated for employees. Employees build up less firm-specific knowledge simply because there's less of it. People from the outside can replace them more easily. At the same time, an employee's day-to-day work improves non-firm-specific knowledge of an open source project that can be taken to another employer. So a developer who is fired can find a job faster than before.

## **Benefits of Being a Committer**

An employee who is a committer is likely to earn higher compensation. Hann and colleagues have empirically verified this for committers to Apache Software Foundation projects<sup>17</sup>.

At any time, the committer-employee can credibly threaten to leave the company, taking significant power and reputation away from the current employer. Employers often pay premium salaries just to employ prominent committers.

But how do you become a committer? Community open source projects tend to be meritocracies, judging developers by their social and technical contributions. In contrast, a company owning some commercial open source gives committer status to its employees (and takes it away) as it sees fit.

Consequently, it makes little sense for the economically rational software developer to invest time in commercial open source. The value these developers create is tied to the product and the owning company. Unless the product is in wide use or the developer wants to work only for this one company, it makes more sense to invest time in a community open source project.

## **How to Become a Committer**

Developers who start projects immediately become committers. However, they now face the task of creating a successful project out of nothing. This is a highly entrepreneurial activity: Developers must promote their project while doing the actual programming work, understanding that the outcome is uncertain.

It's more common to join an existing open source project. Assuming a fair and transparent promotion process, the two main criteria that will get a developer promoted from contributor to committer are:

---

<sup>17</sup> I-H. Hann et al., "Economic Returns to Open Source Participation: A Panel Data Analysis," unpublished working paper, Univ. of Southern California.

- the developer's social and technical abilities, and
- demonstrated commitment to the open source project

This is what most project Web sites state and what developer surveys have revealed.<sup>18 19</sup>  
<sup>20</sup> However, these surveys rely on what developers say they do, and actual behavior could vary from what people believe motivates them.

Because being a committer can have clear financial benefits, keeping the group of committers small is in the economic interest of a committer to a successful project. Not doing so would dilute the committer's value to current and future employers. At least this would be an economically rational person's train of thought.

Counteracting the existing committers' economic interests is the need and desire to build a working community. Also, the participants in a new project are likely to appreciate every helping hand while a mature project might not need any additional committers.

Thus, the following forces possibly influence a developer's promotion:

- the economic self-interest of the group of existing committers,
- the committers' philosophical convictions on running the project, and
- the project's need for more committers

In many ways, investing in an open source project is like joining a startup. The earlier a developer joins, the higher the risk of the project not working out but also the more likely the ascension to committer status. The later a developer joins, the lower the risk, but also the lower the chances of becoming a committer any time soon.

The window of opportunity is small for those aspiring to achieve committer status in an important open source project. With the ongoing commercialization of open source, many current projects expect a committer to work full-time on the open source project. Otherwise, committer status wouldn't be granted. This, for example, is what the Eclipse project Web site states about its core projects<sup>21</sup>. However, a company is likely to let an employee work full-time on an open source project only if that person is already a committer; otherwise how many of the benefits of its contributions the company would reap is uncertain.

A developer who chooses the right project can gain and maintain a position that will increase salary-negotiation power and job prospects. The developer will enjoy those benefits as long as the project is of significance to potential employers.

---

<sup>18</sup> E. Raymond, *The Cathedral and the Bazaar*, O'Reilly, 2001.

<sup>19</sup> E. Haruvy, F. Wu, and S. Chakravarty, "Incentives for Developers' Contributions and Product Performance Metric in Open Source Development: An Empirical Exploration," Univ. of Texas working paper.

<sup>20</sup> K.R. Lakhani and R.G. Wolf, "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects," J. Feller et al., eds., *Perspectives on Free and Open Source Software*, MIT Press, 2005, pp. 3-22.

<sup>21</sup> [www.eclipse.org](http://www.eclipse.org)

Open source reinforces the trend toward employees becoming "free agents." Committers who rationally follow their economic interests are likely to be more loyal to the open source project than to their current employer because that's where their market value lies.

## The Case for Sustainability

AHT is a company that is designed from its inception, to be sustainable, practicing sustainable methodologies to support that goal. In a section of its Business Plan entitled "At the Core", AHT made the following statement:

"AHT management, says 'AHT is a software company masquerading as a housing company'. That statement goes to the heart of the case for AHT possessing a sustainable competitive advantage. Intellectual property (IP) is at the core of AHT, which can be properly called a 'knowledge age' enterprise.

In the knowledge age, a company's value is largely determined by its ability to convert individual and organization knowledge into net worth in time to seize a new market opportunity. As product cycles shorten, and competitors reduce time to market, a competitive corporation must continuously validate and improve its processes to develop and commercialize new ideas. Intellectual property management is the heart of this transformation process. It is a process that addresses the explosive growth of tangible assets and their impact on the company's strategic market position and stakeholder value.

AHT will pursue business practices and technological innovation focused on developing IP based corporate capabilities that are fundamental to improved competitiveness and economic performance."

Open source allows AHT to build innovative leadership into our business model from the beginning — the "sustainable competitive advantage" that will drive future growth. Value is the common denominator between an open source business and its long-term sustainability. The trends in the market clearly show that closed proprietary systems cannot be viably sustained by companies, mostly for three reasons: 1) the cost of supporting a large technical staff, 2) the ever-increasing complexity and expected features of modern software, and 3) the need to stay technically competitive and relevant.

Open Source provides a cost effective alternative to proprietary software. It offers more flexibility for AHT to modify its software to reach our business goals. And, we can have multiple vendors supporting the software solution. For example, Oracle provides cheaper support for RedHat Linux than RedHat itself.<sup>22</sup>

A keen focus on IP sustainability provides AHT with a strategic tool to cut costs, create new revenue options and maintain our innovative image. It provides a way AHT to constantly create new products, manage our processes better and reach new target markets for our products.

---

<sup>22</sup> "Oracle's Red Hat Rip-Off" Linux-Watch, 10/25/2006 - <http://www.linux-watch.com/news/NS7753065928.html>  
CONFIDENTIAL [clewis@advancedhousingtech.com](mailto:clewis@advancedhousingtech.com) 6/4/09

## About Competition

AHT will generally allow, and even depend on, third-parties to modify and improve our open source software. In this way, a feedback-loop is generated:

- We give our software away for free, and charge a service fee for support
- The purchaser finds that some aspect of the software needs to be modified, and makes the appropriate modifications freely
- The purchaser then freely gives the software to another potential purchaser, who agrees in turn to purchase support from AHT

This model has proved somewhat successful, as witnessed in the Linux community. There are numerous GNU/Linux distributions available, but a great many of them are simply modified versions of some previous version. For example, Fedora Linux, Mandriva Linux, and PCLinuxOS are all derivatives of an earlier product, Red Hat Linux. In fact, Red Hat Enterprise Linux is itself a derivative of Fedora Linux. This is an example of one vendor creating a product, allowing a third-party to modify the software, and then creating a tertiary product based on the modified version. All of the products listed above are currently produced by rather successful software service companies.

## The Effect of Open Source on AHT's Bottom Line

Open source software companies create applications that cost less—many are distributed free of charge to start—than mainstream alternatives. Even when customers start paying the software developer a so-called “support license,” costs are amortized on a monthly basis rather than through a large, upfront license. That's an attractive payment model when cash and credit are tight. And IT spending indeed is tight. According to IDC, U.S. tech spending this year is expected to decline 0.9%.

During the dot-com meltdown in the early years of this century, the Linux operating system—still the heart and soul of the open source movement—established its legitimacy, setting the stage for the success of such startups as Red Hat. Even established companies like IBM Corp. and Sun Microsystems decided to make open source a major part of their software strategies.<sup>23</sup>

### ***Financial Sources as a Distributor***

The FOSS revenue model is based mainly on support services. Canonical Ltd. is one such company that gives its software away freely, but charges for support services. The source code of the software is usually given away, and pre-compiled binary software frequently accompanies it for convenience. As a result, the source code can be freely modified. However, AHT will impose some license-based restrictions on re-distributing the

---

<sup>23</sup> “Open Source Vendors Push Advantage” B-to-B Magazine, Rich Karpinski, 3/9/2009, <http://www.tbtonline.com/apps/pbcs.dll/article?AID=/20090309/FREE/303099965/1150/ISSUENEWS#seenit>

software. Generally, software can be modified and re-distributed for free, as long as credit is given to the original manufacturer of the software.

Open source software has a large number of alternative funding streams, which are actually better connected to the real costs of creating and maintaining software. After all, the cost of making a copy of a software program is essentially zero, so per-use fees are unreasonable. At one time, open-source software development was almost entirely volunteer-driven, and although this is true for many small projects, many alternative funding streams have been identified and employed for FOSS:

- Give away the program and charge for installation and support (used by many Linux distributions).
- “Commoditize complements”: make a product cheaper or free so that people are more likely to purchase a related product or service you do sell. This is similar to The Gillette Company giving away razor handles so they could make money on razor blades.
- Cost avoidance/cost sharing: many developers need a product, so it makes sense to share development costs (this is the genesis of the X Window System and the Apache web server).

### ***Fiscal Advantages of Open Source Development for AHT as a System Integrator<sup>24</sup>***

Large system integrators, or solution providers like AHT, gain the most from open source software because they increase profits through direct cost savings and the ability to reach more customers through improved pricing flexibility. Every dollar a system integrator saves on license costs paid to a software firm is a dollar gained that the customer might spend on services.

### **Solutions Demand Curve**

Customers in AHT markets want us to deliver "solutions" — not just houses.

PARAGRAPH REMOVED

That allows the customer to talk to one company, rather than many. The left side of the Figure 3 below (a) illustrates this stack together with the customer demand curve.

---

<sup>24</sup> This section is liberally taken from “The Economic Motivation of Open Source Software: Stakeholder Perspectives”, Dirk Riehle, 2007 - <http://www.riehle.org/computer-science/research/2007/computer-2007-article.html>

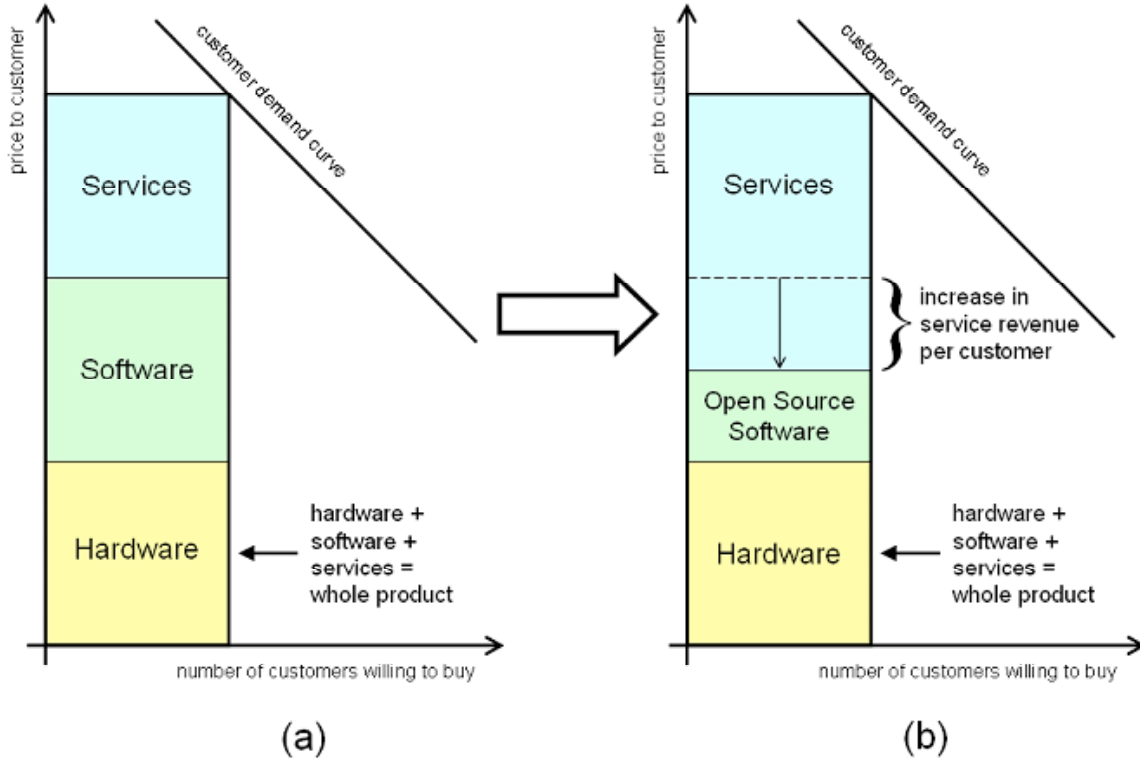


Figure 3. AHT solutions demand curve. (a) System integrators sell a stack of hardware, software, and services. (b) Integrators can charge customers similar prices even if they use open source software.

The demand curve shows how many customers would be willing to buy AHT’s solution at a given price. On the Y axis is the customer’s cost to purchase, and on the X axis is the number of customers who are willing to pay for that solution at the given price. The form of the demand curve varies depending on what is being sold. However, in general, the demand curve is downward sloping: The lower the price, the more customers are willing to buy.

AHT’s profit, as a system integrator depends on which of the stack’s components we own and which we must buy. Usually, a system integrator’s stronghold is services, which puts together the hardware and software pieces to meet the customer’s need. However, if the system integrator owns only the services component, it will have to pay other companies for the software and the hardware and thereby share revenue, leaving less profit for itself.

It is therefore in AHT’s interest to acquire hardware and software as cheaply as possible. Open source software and hardware is typically much cheaper than closed source software, hence its use increases profits for AHT.

Figure 3b illustrates how with stable supply and demand, more money is made in the services part of the value stack if software cost goes down.

Software cost savings are not easily passed on to customers for two reasons: First, customers tend to care about the whole product rather than individual components; second, large system-integration projects are complex and new competition doesn't spring up easily. Thus, system integrators, like AHT can maintain their prices.

While this is one good reason for system integrators to support open source software, there's another equally compelling reason for them to support and contribute to open source software.

## Enhanced Business Growth Potential

The simple value stack that Figure 4 illustrates suggests that AHT can charge customers only one price. In reality, AHT can charge varying prices for a total solution to a prospective customer's problem.

The price charged per customer that Figure 4 shows can be split into our service cost, plus the markup or margin needed to make a profit. If we own just the services part of the stack, the cost for providing that service defines the lower price limit for the work. In a reasonably competitive market, we will accept deals above this limit if we have the resources.

This limit, together with the demand curve, determines the maximum number of customers AHT can sell to and take on, as Figure 4a illustrates.

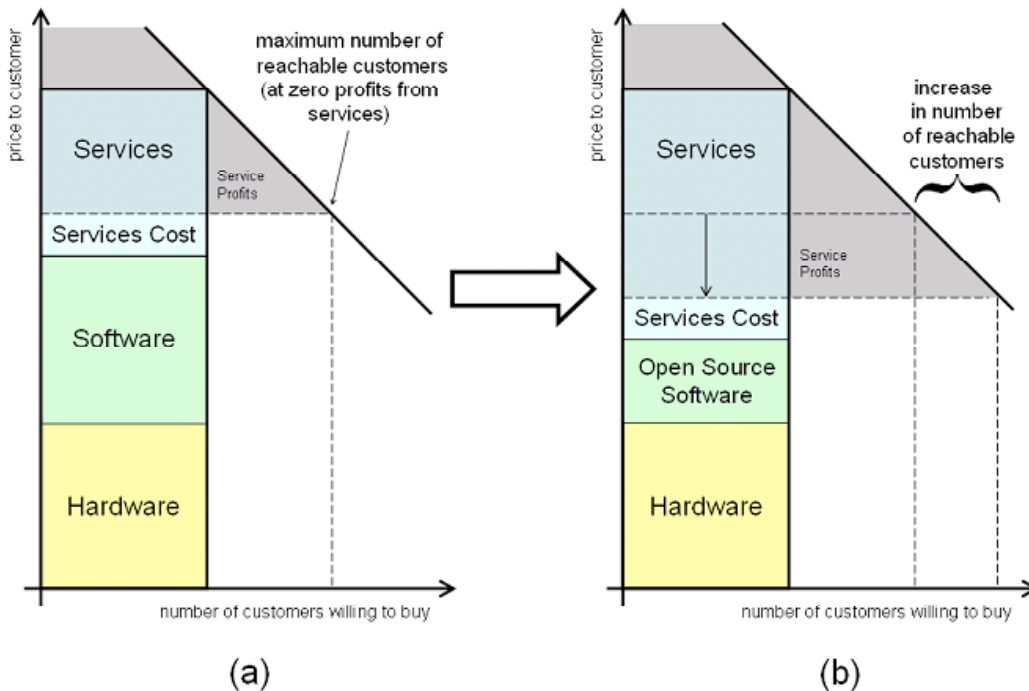


Figure 4. Sales margins and number of customers. (a) The lower price limit determines the customers AHT takes on. (b) Switching from closed source software to open source software can result in more customers and higher profits.

Switching from more expensive closed source software to less expensive open source software increases the profits of a sale through the money saved on the software. It also reduces the lower price limit for possible deals and puts a new set of more price-sensitive customers within reach. Not only does open source software improve profits on the original individual sales, it also increases the total number of potential customers.

Figure 4b shows how a switch from closed source to open source software results in more potential customers. And more potential—and presumably satisfied—customers mean higher sales and profits. The total profit is represented as the area of the gray triangle under the demand curve, showing the increase in profits when moving down the curve. Since in reality AHT will own many of a total solution's components, including software and hardware, more customers mean more profits through these components as well.

## Generating Software Profits

AHT will increase its profits if it reduces the software cost. By reducing software cost, we can move down the demand curve and sell to more customers.

Closed source software is the main obstacle to doing this: It cuts into profits on an individual sale and reduces overall pricing flexibility.

Before the advent of open source software, entering an established and well-defended market was a risky proposition. With increasingly well-understood open source processes, setting up an open source project competing with an established closed source market leader's product is much less risky and carries a significantly higher chance of success than before.

The fact is that community open source software has a high chance of taking new markets early on. Only strong intellectual property protection or other competitive advantages might lead a closed source company to win and keep a new market. Leaders in established markets might be able to defend their positions for a long time. They tend to dig in with complex products, established processes, customer data lock-in, and many other positional advantages. Still, open source might well prove to be disruptive enough to conquer even these markets.<sup>25</sup>

---

<sup>25</sup> C. Christensen, *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*, Harper Business, 2000.

## Perceived Disadvantages of Open Source Models

The open source development model has its critics, who validly point out disadvantages. However, most are only disadvantages to classical (proprietary) development models. Let us review some of them:

- *There is no guarantee that development will happen.* In other words: it is not possible to know if a project will ever reach a usable stage, and even if it reaches it, it may die later if there is not enough interest. Of course, this is also a problem with proprietary software, but it is more evident in the case of open source.
- *There may be significant problems connected to intellectual property.* This point is especially important, now that some countries are accepting software and algorithm patents. It is very difficult to know if some particular method to solve a software problem is patented, and so the community can be considered guilty of intellectual property infringement. Some open source packages are already addressing this issue with switches or patches that enable or disable patented code fragments according to the country where the code is used. In other cases, developers consider source code not as an executable device, but a mere description of how a device (the computer) executes, and therefore uphold the idea that source code is not by itself (in absence of an executable program) covered by patent law even in countries where software patents are accepted. However, this idea has still to be tested in courts, and many people do not believe in its viability.
- *It is sometimes difficult to know that a project exists, and its current status.* There is not much advertising for open source software, especially for those projects not directly backed by a company willing to invest resources in marketing campaigns. However, several “aggregation points” for open source software do exist, although in many cases they are usable only by experts, and not by the general public. They are also in many cases very specific to some software category, such as scientific software or database systems.

### **The Legal Challenges**

Software patents, especially when they are granted for trivial algorithms that can easily be reinvented by many developers, represent a serious threat to individual open source developers and small organizations, that cannot afford the costs of patent litigation. Ironically, the situation is even more crucial for open source software than for black box proprietary software, since the code is directly accessible by the patent holders.

In many cases, companies and individuals are trying to get exclusive right on certain technologies through patents, and recently more and more patents on fundamental algorithms and procedures have been granted, especially in the United States. This is a potentially dangerous practice, not only for open source software in particular, but for the

software industry and software practitioners in general. That is because the relatively long time span of currently issued patents and the breadth of some of them are especially disturbing. Also, there is insufficient control on the existence of previous work, and many patents are issued on obvious and ill-defined concepts. These patents can be used as broad-fire weapons against competitors, especially against smaller ones, unable to afford the costly legal expenses needed to demonstrate that a patent is invalid.

Open source software may be especially vulnerable to patent-based attacks, because only a few open source-based companies have the financial power to protect themselves against patent lawsuits. Also, if a patent is issued on a very broad technology or technique it may be impossible to circumvent the patent and create a patent-free alternative.

A not surprising, and ironic (given its IP fights with Apple in the 1990's) take on open source software development is offered by Microsoft. In its 2008 Annual Report, Microsoft stated that FOSS business models challenge its license-based software model and that the firms who use these business models do not bear the cost for their software development. The company also stated in the report<sup>26</sup>:

“Some of these [open source software] firms may build upon Microsoft ideas that we provide to them free or at low royalties in connection with our interoperability initiatives. To the extent open source software gains increasing market acceptance, our sales, revenue and operating margins may decline.

Open source software vendors are devoting considerable efforts to developing software that mimics the features and functionality of our products, in some cases on the basis of technical specifications for Microsoft technologies that we make available. In response to competition, we are developing versions of our products with basic functionality that are sold at lower prices than the standard versions.”

## The Jury is Out

Microsoft's objections to open source actually reveals the flaws in their, and other's, legal arguments (and a bit of hypocrisy). What Microsoft fails to mention is its own reliance on mimicking the features and functionality of others' products (Apple, WordPerfect, Sony PlayStation, etc.). This is how the world works. It's how the industry progresses instead of being forced to reinvent the wheel over and over. Microsoft has benefited from this as much as any other company, if not much more so.

It is extremely important to “see” through the various interpretations of the advantages and disadvantages of open source, and if possible try to analyze with quantitative methods if open source can be helpful in a given situation, or for a given user or company. The unique philosophy of open source plus the recent explosive growth in availability and usage of open source software have focused interest on how open source licensing works and its implications. As open source has come into the spotlight, a

---

<sup>26</sup> <http://www.sec.gov/Archives/edgar/data/789019/000119312508162768/d10k.htm>

number of points have stirred special interest. The licenses on which the Open Source approach is based have never been conclusively tested in court.<sup>27</sup>

In his assessment of the legal implications of open source software, DM Kennedy writes:

“We are likely to see further modifications in the basic Open Source licenses and further rethinking of the philosophies at the foundation of these licenses. In addition, it is fair to expect that most of the licenses will see some discussion and potential revision in order to shore up some of the legal issues that have arisen and may later be raised by the Uniform Computer Information Transaction Act (UCITA) or other new laws. There will be a continuing battle to keep the number of Open Source licenses limited and specific, and a growing pressure from each commercial developer releasing software into Open Source to create its own licenses. The role of the community in the development of this process will remain important, but the role of courts will likely play a larger role in the development of Open Source.

Perhaps the most important of the notions that arise out of Open Source, the notion of copyleft, using licenses to protect freedom and a public interest approach to intellectual property, will continue to play an important part in the development of intellectual property law. There will be a place at the table not just for the protection of intellectual property rights but for the protection of the rights of the community to use intellectual property developed for the good of the community. The development of the Open Source movement, the Open Source licenses, and the notions of copyleft and software freedom will play a very important role in the development of the Internet and our approach to intellectual property law in the future.”<sup>28</sup>

The various open source organizations<sup>29</sup> have resources to legally fight off the increasingly less than ardent challenges of proprietary organizations. Proponents of closed software (and hardware) systems are seeing the writing on the wall. They simply are not going to yank all the computers being sold with Linux, and companies like Dell and Sun have the resources to aggressively fight any attempt.

An intriguing, inexpensive, and effective tactic to fight concerns about patent liability employed by open source software companies and consortiums is to ask the U.S. Patent & Trademark Office to re-examine patents that may pose a threat to them. Fears about patent litigation have plagued both open source companies and their customers, and stem from patent trolls as well as competitors. Going to court over a patent typically costs \$1M to \$4M, while filing this type of re-examination request costs \$50,000 to \$100,000,

---

27 Rosenberg, Open Source: The Unauthorized White Papers 92 (M&T Books 2000)

28 “A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft and Copyfuture”, Dennis M. Kennedy - <http://www.denniskennedy.com/opensourcedmk.pdf>

29 Open Source Initiative, The Eclipse Foundation, The Free Software Foundation, The Linux Foundation, The Mozilla Foundation, OASIS, ODF Alliance, Open Invention Network, Python Software Foundation, etc.

depending on the complexity of the claim. That's still a tidy sum for a small company, but one that is easily divided among several concerned parties.<sup>30</sup>

## Conclusions

Whether running a ubiquitous desktop computer, part of a deeply embedded system, or elemental to the inner workings of a complex software system, open source is essential to how computing is done in the world. Like it or not, open source will play an increasingly important role in the evolution of all aspects of business in the future.

Fortune 500 companies are now making sure the software they need for their business continues to be available in open source. They've gone through multiple license nightmares with Microsoft, where suddenly their outlay became much larger and got locked into multiyear-long deals. Many don't want to be on the pusher-addict model of software anymore and are going to take a more active voice in directing that software.

Legal challenges to open source hardware and software initiatives have become weaker over time, not stronger. As enterprise-level businesses, and even governments switch to open source operating systems like Linux, the likelihood of it becoming the norm is vastly increased.

PARAGRAPH REMOVED

The financial case for open source development can hardly be more compelling: as a new technology company specializing in systems integration AHT is poised to ride the forward edge of the emerging wave of sustainable companies. Money we save on development costs, licensing, and engineering is directly plowed back towards growth.

PARAGRAPH REMOVED

Open source software has already started to modify the rules in the information technology industry, which will produce enormous changes in the years to come. Given these facts, it is clear that those countries and companies which adopt open source technologies, in the short term, will have a huge competitive advantage, and that society in general will benefit from this. As a multinational corporation AHT is in a good position to take early advantage of open source, and can also help the open source movement to get stronger. This collaboration can only be good for both parties.

---

<sup>30</sup> “Open Source Firms Urged to go on Legal Offensive”, James Niccolai, IT World, March 26, 2009.  
<http://www.itworld.com/open-source/65185/open-source-firms-urged-go-legal-offensive>