

White Paper

**Opposites attract: how can the flexibility of open source and the assurance of proprietary software coexist?**

March 2010

# Contents

INTRODUCTION ..... 3

TWO OPINIONS, TWO TRUTHS ..... 3

CHEESE, MOUSETRAPS AND CAPITALISTS..... 4

A CAT STILL IN THE BAG ..... 6

SALES AND SERVICE ..... 7

SECURITY FIRST ..... 8

RIGHTS LEFT AND RIGHT ..... 10

A REASONED COMPROMISE ..... 10

ABOUT BITRIX ..... 13

CONTACTS ..... 13

## INTRODUCTION

The time of full-throttle competition between free open-source software (FOSS) and proprietary software (PS) is now in the past. The interaction between these paradigms has changed course into a search for coherent models which unify, rather than separating, the main principles of the two. Enterprises which use heterogeneous information systems are hardly exotic; they are rather the norm—so it seems that these two systems, which exist side-by-side in reality, may not be the natural enemies that they appear to be at first glance.

Over the last ten years, FOSS has traveled along a long and difficult path from ‘homemade’ products to breakthroughs at all levels up to the enterprise level. Well beyond the realm of computer geeks and niche users now, the FOSS of today presents real alternatives to proprietary software both at the technical level and at the psychological level, as customers find that the open source model brings user communities, a level of welcome self-determination, and several other advantages with it.

The number of installations of FOSS products has always shown impressive growth, but still ranks well below PS. According to the research group IDC, the FOSS market in 2013 will reach \$8.1 billion<sup>1</sup> from a total market of hundreds of billions. The rise in FOSS use is not surprising, considering the economic impact for the end-user. According to Gartner, 85% of organizations used FOSS for some purposes, with about 15% of the rest claiming that they will make some move in that direction within 12 months.<sup>2</sup>

When deciding on a platform for any corporate service, end-users are faced with a dilemma. What are the real strengths and weaknesses of FOSS and PS? Which concept is better for the task at hand? Is there an alternative somewhere in the middle? This document will investigate that and several other issues as well as introduce the reader to the hybrid licensing concept which Bitrix has adopted.

## TWO OPINIONS, TWO TRUTHS

According to Wikipedia: FOSS is “software that is liberally licensed to grant the right of users to use, study, change, and improve its design through the availability of its source code.”<sup>3</sup> There is a dozen of types of ‘free’ licenses, which determine the specific use, installation, execution, distribution, modification and other actions undertaken with the program.

---

<sup>1</sup> <http://www.gnu.org/philosophy/free-sw.html>

<sup>2</sup> Gartner, "User Survey Analysis: Open-Source Software, Worldwide, 2008", November 2008, <http://www.gartner.com/it/page.jsp?id=801412>

<sup>3</sup> [http://en.wikipedia.org/wiki/Free\\_and\\_open\\_source\\_software](http://en.wikipedia.org/wiki/Free_and_open_source_software)

The origins of FOSS are rather curious. At first, FOSS was distributed without any documented commitment for free usage and modification. Developers felt that personal agreement was enough. However, the situation that occurred with Richard Stallman, an authoritative American software freedom activist, proved that assumption to be false. Stallman developed a text editor on the basis of source code developed by a fellow developer, the rights to which were soon sold to a commercial company. This company in turn required that Stallman stop distributing his editor, making its claim on the basis of the purchased source codes. Stallman had to re-write the program and develop the General Public License (GPL) to protect future work.

Generally, FOSS has the following characteristics:

- Open source code;
- Very little or no limitation on the use or modification of code;
- No payment for the product license
- Product developed and supported by independent developers

Use of PS is always determined by a license agreement, and use is usually quite limited. The license agreement generally states what rights the end users have, which results in the very secure protection of the rights of the developer. In summary, FOSS licenses assure that the product will always be publicly available; PS licenses ensure that the owners will have material interest in keeping the product relevant.

Proprietary software is generally distinct for the following:

- Licenses are sold for a price determined by the rights owner;
- Support and development are performed by one company;
- Licenses limit or can forbid certain types of use, distribution, and modification.

## **CHEESE, MOUSETRAPS AND CAPITALISTS**

‘Free’, when talking about free open source software does not mean without cost, despite common assumptions. The official site of GNU, a free software collaboration project, clearly states that

“Free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer.”<sup>4</sup>

Indeed, FOSS is provided without charge. A user can download the source code from the official site and use it, with a few exceptions, at his or her discretion. This seems to be an obvious advantage of FOSS. Indeed, this feature is the foundation of 268 government initiatives suspending use of PS, as researched by the Center for Strategic and

---

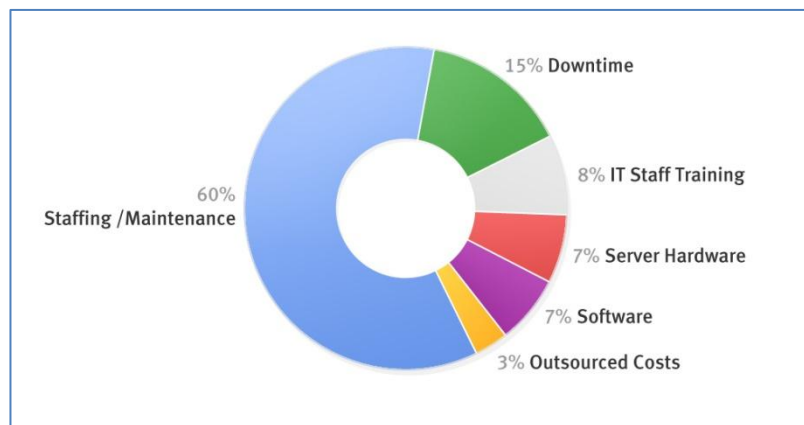
<sup>4</sup> <http://www.gnu.org/philosophy/free-sw.html>

International Studies<sup>5</sup>. Likewise, a comment by the speaker of the Israeli Finance Ministry concerning the refusal of use of Microsoft office products bears witness to the importance of this factor: “The move with Microsoft was a purely economic decision”.<sup>6</sup> The same spirit hangs over recent decisions by the US Navy<sup>7</sup> and Defense Department<sup>8</sup>, and the government of Ukraine’s undertaking of development of its own Linux-based operating system.

However, the difference between a ‘program’ and a ‘solution’ is rather large. This difference is rooted in the inherent complexity of software programs and their usage, and is revealed in the effort and money required during installation, configuration, maintenance and training of personnel. To illustrate, a ‘program’ is the main ingredient to a meal, but without proper preparation, it will remain unfit to eat.

“Just because something is free doesn't mean that it has no cost,”<sup>9</sup> says Laurie Wurster of Gartner, clearly marking the difference. The real value of ‘solutions’ describes the TCO (Total Cost of Ownership) indicator, which has a number of contributing factors, among which license costs is only the most easily measured.

According to an IDC<sup>10</sup> study based on analysis of the costs of about 300 corporate customers in over 3 years, TCO of software products consists of the following elements:



<sup>5</sup> CSIS, “Government Open Source Policies”, August 2007, [http://csis.org/files/media/csis/pubs/070820\\_open\\_source\\_policies.pdf](http://csis.org/files/media/csis/pubs/070820_open_source_policies.pdf)

<sup>6</sup> InformationWeek, “Israel Suspends Acquisitions Of Microsoft Software”, October 2003, <http://www.informationweek.com/news/software/enterpriseapps/showArticle.jhtml?articleID=17100349>

<sup>7</sup> “Department of the Navy Open Source Software Guidance”, June 2007 [http://oss-institute.org/Navy/DONCIO\\_OSS\\_User\\_Guidance.pdf](http://oss-institute.org/Navy/DONCIO_OSS_User_Guidance.pdf)

<sup>8</sup> “Clarifying Guidance Regarding Open Source Software (OSS)”, October 2009. <http://cio-nii.defense.gov/sites/oss/2009OSS.pdf>

<sup>9</sup> CNet, “Gartner: 85 percent of companies using open source”, November 2008, [http://news.cnet.com/8301-1001\\_3-10098624-92.html](http://news.cnet.com/8301-1001_3-10098624-92.html)

<sup>10</sup> IDC, «Demonstrating Business Value: Selling to Your C-Level Executives”, April 2007, <http://download.microsoft.com/download/1/9/2/192e73a4-7abb-4bad-b469-34632d54a8a6/IDC%20Whitepaper%20Demonstrating%20Business%20Value.pdf>

This research shows that the savings attainable from eliminating license fees is only 7%. Thus, in the case of designing a website, where a company needs to pay for services including site development, maintenance and updates, hosting, and content management, the license cost is likely to an even smaller percentage of the total cost. Scalability is often an extremely important issue and with it come additional long-term development costs. The larger the project, the smaller the share of the license fee in total cost.

PS apologists emphasize that additional expenses associated with FOSS 'compensate' for the initial economy. This claim is based on the additional expenses involving service, training, and losses associated with system instability and downtime. According to research by Forrester, 57% of small and medium businesses expressed concern with the complexities of installing FOSS<sup>11</sup>. Apologists of FOSS, naturally, counter these arguments by noting the SP is less than perfect and add that the SP community has material interest in showing statistics which favor SP. Overall, it seems that the PS crowd has the stronger argument, and the figures and the market, side with them.

In our view, there is not a categorical answer to the question about what type of product is better to use. Deciding which course to take must be based on knowledge that comes from experience.

## **A CAT STILL IN THE BAG**

The different concepts and philosophies of distribution of software make an impression on the software that is produced in each respective category.

FOSS development looks somewhat like theoretical science, while PS is much more like applied science. The goal of FOSS developers is rarely connected directly to the market requirements; it is often an addition or add-on to a product that holds theoretical interest, but is not in demand among the vast majority of users. Among the most obvious consequences of this rudimentary fact of FOSS development is a shortcoming in usability exhibited by the vast majority of products across the industry, as compared with PS.

In contract to FOSS, a successful PS product will have a long, guided path of development through several editions. This process is directed by the needs of the majority of clients of the product, not by the desires of developers. The division of development duties allows specialized tasks to be performed more easily and efficiently, and in tune with market tendencies and the needs of the target audience. From the end-user's point of view, that provides a large degree of long-term confidence and transparency concerning the future of the product.

---

<sup>11</sup> ComputerWorld UK, "Forrester: Open source security fears persist", June 2009, <http://www.computerworlduk.com/toolbox/open-source/open-source-business/news/index.cfm?newsId=15147>

In general, commercial development focuses more attention on practical problems, which are not of great interest to independent developers. Thus, it is natural that more attention is given to user interfaces, issues of usability, and well-developed real-life scenarios in the PS system than in FOSS.

On the other hand, the theoretical approach to development has advantages: FOSS solutions are often address fundamental issues more directly than commercial solutions, and the very latest standards and technologies are often more developed by the FOSS community. Unlike the trailblazers in the FOSS community, suppliers of PS are risk-averse and prefer to see a clear market need before implementing new elements to their product range. Summarily, it works out that FOSS covers a wider range of tasks, but requires significantly more adaptation for each individual installation.

It should be noted that access to the source code means that, other than the license agreement, there are really no barriers for the use or modification of the functionality of FOSS products. However, this is a rather complicated and time-consuming process, and so the number of beneficiaries from this feature is limited, even if the diversity in developed features in the whole of the community is great. For most end-users, however, it is considerably easier to find a finished product which does not require serious modifications or additional development.

Finally, let it be said that as a rule, FOSS is a construction set, a set of parts with much assembly required, as well as patience, to obtain the desired outcome. Basic functionality is provided in FOSS platform products, but significant experience and knowledge to incorporate third-party modules is required. The extent to which these third-party modules are compatible and convenient in use is another issue. A PS product put out onto the market is completed, tested and built around the overall concept of the system in which it functions.

## **SALES AND SERVICE**

It's probable that the Achilles' heel of FOSS is sales and service. Without a unified development team and a dedicated service department, FOSS users are relegated to searching for answers in forums, chat rooms, and other similar media. While in some sense, the information to be found through such methods is an example of the marvelous fruit of the worldwide web, it is not a practical or reliable method for small and medium-sized companies who simply need the answer or a new patch. Support can be found on a paying basis, but this defeats the main advantage of FOSS.

Same situation is met concerning documentation of FOSS, which often is incomplete and spread out over a wide range of sources. Developers which are not forced to write documentation often leave it off of their to-do lists. PS comes with complete documentation, and developers, for the great part, are obligated to keep this documentation updated and easily accessible to the end-user.

In using FOSS, a company becomes the de facto bearer of a wide range of additional responsibilities and risks connected with the solving, and delays in solving, unavoidable

small and not-so-small problems which arise in the implementation of virtually any software product. The large and loyal community of developers – an indisputable plus of successful FOSS applications – is still not a service level agreement and cannot guarantee a timely response to the urgent problem of a customer.

Indeed, the main differences between FOSS and PS appear in the long-term use of the respective products. If in the first case, the risk comes down on the shoulders of the end-user, then in the second, they are shared to a degree by the vendor.

Although the typical end-user agreement describes the purchase of the proprietary product ‘as is’, the vendor nonetheless supplies a guarantee of the quality of the technical support and the regularity of updates to the product and its documentation. In contrast to an independent developer, a commercial developer has something to lose, since reputation and word-of-mouth (or forums) play an important role in business, no matter the exact wording of the contract.

Centralized support protects the investment made in an IT project. In practice, when FOSS is used, the situation often arises where only one person in a company has the knowledge and abilities to keep the system functioning – creating a variety of risks for management and an imbalance of power inside the company. Purchase of PS solution, which to a great extent is an off-the-shelf product, eliminates nearly all of this risk, as the end-user can always receive service from the vendor or qualified engineer. Forrester finds that 68% of small companies express concern about the risks of service and support associated with FOSS<sup>12</sup>.

## SECURITY FIRST

Is there any point in buying an ‘advanced’ product, head and shoulders above its competitors in functionality, if it does not provide an acceptable level of security? This is a rhetorical question, but an app with poor security is like a pocket with a hole – whatever you put in falls out. For some applications, this is critical. For example, a website management system must protect itself and the data of its users. There is no point in having all the functions in the world if the website is vulnerable to attack. A single incident can ruin a reputation forever.

FOSS developers often declare that their products exhibit unsurpassed levels of safety, explaining that FOSS is attacked less and has fewer known vulnerable points. For malicious software to appear in a given environment, three criteria must be met: popularity, documentation, and vulnerability. Although FOSS is not nearly as popular as PS in most application categories, there are many examples of vulnerabilities being exploited, not least because security, much like documentation, is simply not high on the priority list of most FOSS developers. The decentralized development model, by

---

<sup>12</sup> ComputerWorld UK, “Forrester: Open source security fears persist”, June 2009, <http://www.computerworlduk.com/toolbox/open-source/open-source-business/news/index.cfm?newsId=15147>



definition, does not allow implementation of quality control techniques that are used in commercial software development.

The truth is that open source code (albeit carefully inspected by the community) does not guarantee security, nor does the quick release of new editions somehow guard against vulnerabilities. In fact, practice indicates the opposite. According to a study by IBM X-Force, among the top ten most dangerous applications in 2009 were 4 free CMS products (Drupal, Joomla!, TYPO3 and Wordpress). Furthermore, only 33% of the vulnerabilities discovered in these systems were fixed.<sup>13</sup>

For these reasons and others, the worldwide IT market is still cautious in its use of FOSS. Forrester reports that security issues with FOSS concern 58% of IT professionals.<sup>14</sup> Given that there is no guarantee of future development of the products, including patches to close security breaches, let alone the timeliness of the fixes which are produced, it is clear that this wary 58% recognize the risk that they bring upon themselves.

Conversely, IT security is one of the very highest priorities of the commercial developer. Security is a major issue to nearly all customers, and thus it is important to vendors. PS solutions which are released must protect the customer, or else they will not protect the future business of the vendor; the connection could not be more direct.

Presently, practically all large software development companies have quality assurance techniques, which guarantee security in all steps of the development process. This means that the source code of the product is meticulously checked from the beginning to the end. Clearly, this systematic approach is many times more effective than the happenstance modifications pooled together in the FOSS model.

The increased level of security provided by PS is confirmed in reports by various analytical companies and agencies. Thus Gartner, traditionally loyal to FOSS, recently admitted that Microsoft solutions are no less vulnerable than their FOSS counterparts. Firefox took first place among the most vulnerable browsers, with almost twice the number of breaches as Internet Explorer and three times that of Safari<sup>15</sup>.

Of course, not all PS products fit the descriptions or contain the characteristics mentioned here. Which is why independent certification is used to confirm the security of various products.

---

<sup>13</sup> IBM X-Force, "2009 Trend and Risk Report", February 2010, <http://www-935.ibm.com/services/us/iss/xforce/>

<sup>14</sup> Forrester, "The State Of SMB Software: 2009", June 2009, [http://www.forrester.com/rb/Research/state\\_of\\_smb\\_software\\_2009/q/id/54556/t/2](http://www.forrester.com/rb/Research/state_of_smb_software_2009/q/id/54556/t/2)

<sup>15</sup> Cenzic, Web Application Security Trends Report Q3-Q4, 2009, November 2009, [http://www.cenzic.com/downloads/Cenzic\\_AppsecTrends\\_Q3-Q4-2009.pdf](http://www.cenzic.com/downloads/Cenzic_AppsecTrends_Q3-Q4-2009.pdf)

## RIGHTS LEFT AND RIGHT

Although there is a certain friendliness in the FOSS community, it can hardly be said that it is without danger. Communities can and do have their intrigues, some of which can lead to witch-hunt proportions.

More concretely, there are several dozen different types of licensing arrangements which are grouped under the 'open source' label, including four types of GNU, MIT, BSD, Creative Commons, CDDL, AROS, and Mozilla Public License. "Free", as stated before, doesn't mean without restrictions, and while private users are unlikely to run up against the barriers instated by such agreements, organizations should read these agreements no less carefully than end-user agreements from commercial vendors.

An experience by network equipment maker Cisco can serve as an example. In 2008, a claim was made against the company's use of a program which was distributed under a GPL license. As a result, the Free Software Foundation brought a suit against the company, which settled the matter with a contribution to the Foundation.<sup>16</sup>

When implementing FOSS into an information system, "Companies must have a policy for procuring [open source software], deciding which applications will be supported by [open source software], and identifying the intellectual property risk or supportability risk associated with using [it]. Once a policy is in place, then there must be a governance process to enforce it,"<sup>17</sup> according to a Gartner research director.

For the business user, there is more than meets the eye concerning in the FOSS model.

## A REASONED COMPROMISE

"Even if you've been swallowed whole, you've still got two ways out," a certain folk saying goes. Between FOSS and PS there is a reasonable compromise, which takes into consideration the aforementioned advantages and weak points. Bitrix offers its clients a hybridized license agreement, which offers the flexibility and openness of FOSS along with the guarantees, confidence, and attention to detail of PS.

This model entails a reasonable license fee, which includes tech support and updates for the product. The end-user receives an out-of-the-box product *with its source code* and a ready API. Thus, the acquired product allows for further development, including the plugging in of additional modules to accomplish user specific tasks. Access to the source

---

<sup>16</sup> Wikipedia, "Free Software Foundation v. Cisco Systems", [http://en.wikipedia.org/wiki/FSF\\_vs.\\_Cisco](http://en.wikipedia.org/wiki/FSF_vs._Cisco)

<sup>17</sup> Daniweb, "Gartner Report Exaggerates Open Source IP Concerns", November 2008, <http://www.daniweb.com/news/story219909.html>

code also makes full auditing of the product possible with the goal of certifying and prompt changes application.

The terms of the license agreement allow for modification of the code, creation of bespoke modules, and their commercial or non-commercial distribution through, or without, the Bitrix Marketplace. A Bitrix reseller, distributor and/or partner receives the right to add services, modules, support, or other customization to the Bitrix products which are sold on to that partner’s customers. On the one hand, the stability and guarantee to the end-user for the quality of the product is provided by Bitrix. On the other hand, the partner has the opportunity to adapt the product so as to acquire significantly higher margins in the final transaction.

Specially created for small and medium business, Bitrix products stand out for their affordability, ease of installation, and quality of support. The simplicity of installation and use significantly reduces overall expenses associated with software acquisition, since installation of many products often requires the assistance of expensive IT consultants.

Thanks to the hybrid licenses and product features Bitrix is distinguished for lower TCO and shorter implementation time. The table below carries an illustrative example of the first-year TCO figures and project implementation times which are typical for commercial websites. Basic functionality of a project includes content management, web analytics, administration of advertising campaigns, multimedia capabilities, e-learning, search engine optimization, e-commerce and integrated protection against web attacks. A website is ‘packed’ in its original design and configured according to the requirements of the client. The work is assessed as being completed by a hired consultant. The table shows costs and timeframes for implementing the website using an average FOSS product, PS product and Bitrix Site Manager.

	FOSS	PS	Bitrix Site Manager
License Cost	Free	\$5000	US\$ 2,399 <small>Enterprise Edition</small>
Training	US\$ 5000 3 days  <small>Cost includes the creation of a customized training program for the solution.</small>	US\$ 3000 1 day  <small>Training by an authorized distributor or VAR.</small>	0 1 day  <small>Training can be accomplished online through courses, video material and other documentation.</small>
Installation	US\$ 10,000  10 days  <small>Includes integration of additional modules for basic functionality.</small>	US\$ 10,000  10 days	US\$ 10,000  7 days  <small>Installation is quicker because of readymade templates and standard procedures.</small>

Support	US\$ 10,000 Contract with supporting company.	0	0 Technical support comes with the price of the license
Customization of the product	US\$ 5,000 10 days	US\$ 5,000 10 days	US\$ 5,000 7 days
Legal services	US\$ 3,000 5 days 8 hours consulting with the developer, 8 hours consulting time with an attorney.	0	0
Total project expenses and timeframe	US\$ 33,000 28 days	US\$ 23,000 21 days	US\$ 17,399 15 days

These calculations show that there are many issues hidden below the seemingly calm surface of FOSS. In spite of the cost-free license, any economy is eaten away by the higher costs of deployment and support. More importantly, the client becomes, essentially, the hostage of the installation company, which has exclusive knowledge of the resulting system. With Bitrix, the client receives a product that can be serviced by any number of potential service providers.

It is notable that the hybrid system not only provides significant economy in total cost, but also in installation time.

Obviously, actual figures from product installations vary widely and companies have differing requirements based on innumerable factors. However, it is clear that the product hybrid licensing is a well-founded choice, which incorporates the advantages and evens out the disadvantages of both FOSS and PS. This approach is suitable for small and medium-sized businesses, for which effective use of resources is paramount. With Bitrix products, the end-user gets a wide range of functionality, flexibility, and deep development of usability and security with lower total cost and risk.

## ABOUT BITRIX

Bitrix is a privately-owned company developing an advanced business communications platform to bridge SMBs with their customers (Internet), partners (Extranet) and employees (Intranet). Founded in 1998 and headquartered in Alexandria, VA, Bitrix now incorporates 70+ staff, 30,000+ customers and 4,000+ partners worldwide. The customer list includes Hyundai, Volkswagen, Panasonic, Gazprom, Xerox, PricewaterhouseCoopers, DPD, VTB, Samsung and Cosmopolitan. Localized into 13 languages, the company's products are distinguished for their pioneering technology, unique security features, extreme performance capacity and unmatched ease-of-use.

## CONTACTS

### US HEADQUARTERS

901 N. Pitt str  
Suite 325  
Alexandria VA 22314  
USA

Tel./Fax: +1 703 740 8301

### RESEARCH & DEVELOPMENT

261 Moskovskiy Prospekt  
Kaliningrad  
236001  
Russian Federation

Tel./Fax: +7 4012 51 05 64

SKYPE: consult.bitrixsoft

TWITTER: <http://twitter.com/bitrixsoft>

E-mail: [info@bitrixsoft.com](mailto:info@bitrixsoft.com)

[www.bitrixsoft.com](http://www.bitrixsoft.com)

© 2010 Bitrix, Inc. All rights reserved.

Bitrix is a registered trademark of Bitrix, Inc. in the United States and other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.