DQ1: LAMP

[Collapse](https://elearning.uol.ohecampus.com/webapps/discussionboard/do/message?action=list_messages&forum_id=_952804_1&nav=discussion_board&conf_id=_441678_1&course_id=_1470762_1&message_id=_15931440_1)

Top of Form

Summarise the advantages and disadvantages of using open source technologies such as LAMP (Linux-Apache-MySQL-PHP).

Bottom of Form

1. Introduction
   1. What is open source software
      1. OSI, 2014)
         1. Open source software is software that can be freely used, changed, and shared (in modified or unmodified form) by anyone. Open source software is made by many people, and distributed under licenses that comply with the Open Source Definition.
      2. Opensource.com, 2014
         1. The term "open source" refers to something that can be modified because its design is publicly accessible.
         2. Open source software is software whose source code is available for modification or enhancement by anyone.
         3. "Source code" is the part of software that most computer users don't ever see; it's the code computer programmers can manipulate to change how a piece of software—a "program" or "application"—works. Programmers who have access to a computer program's source code can improve that program by adding features to it or fixing parts that don't always work correctly.
         4. Open source software is different. Its authors make its source code available to others who would like to view that code, copy it, learn from it, alter it, or share it. LibreOffice and the GNU Image Manipulation Program are examples of open source software. As they do with proprietary software, users must accept the terms of a license when they use open source software—but the legal terms of open source licenses differ dramatically from those of proprietary licenses. Open source software licenses promote collaboration and sharing because they allow other people to make modifications to source code and incorporate those changes into their own projects. Some open source licenses ensure that anyone who alters and then shares a program with others must also share that program's source code without charging a licensing fee for it. In other words, computer programmers can access, view, and modify open source software whenever they like—as long as they let others do the same when they share their work. In fact, they could be violating the terms of some open source licenses if they don't do this.
      3. Jaisingh, J, See-To, E, & Tam, K 2008
         1. The primary difference between OSS and cSS is that the source code of an OSS is accessible to everyone whereas the source code is proprietary in the case of a cSS.
   2. What are examples of open source software?
      1. (University of Liverpool/Laureate Online Education, 2014)
         1. Linux, Apache, MySql, PHP
      2. Filezilla
2. Advantages
   1. Jaisingh, J, See-To, E, & Tam, K 2008
      1. One consequence of keeping the source code open is that the OSS can benefit from modifications and improvements made by programmers from all around the world. The cSS, on the other hand, can only be improved or modified by programmers hired by the firm developing the cSS (henceforth referred to as the firm when there is no risk of confusion). It is well known that OSS such as Linux and apache have been developed and enhanced by contributions from thousands of volunteer programmers. another consequence of keeping the source code open is that the firm cannot charge a price for purely selling the OSS—the open nature of the source code will drive the price down to zero. Most of the OSS such as Linux, apache, SendEmail, etc., can be obtained free of charge.
   2. Updegrove, A 2009
      1. Access to code: When a user installs proprietary software, they become entirely dependent on the vendor for it's quality, improvement and performance, because the customer has neither the technical means (access to source code) nor the right (legal permission) to alter the code. If the customer needs new or different features, the vendor may or may not be willing to customize the program (either at all or at a price the customer is willing to pay), and if the vendor discontinues support for the product, or goes out of business, the customer is stranded.“ In contrast, a customer with a FOSS alternative has the ability as well as the legal right to change the code any time that it wants to. It can also hire anyone it wants to help it change or maintain the code, and if the project that created the code goes dormant, it may be disappointed, but it will not be stranded.
      2. Freedom from lack in: While open standards increasingly give customers protection from “lock in" dependency on a single vendor, and the certainty of signiﬁcant switch g costs if they wish to change vendors), changing from one product to another can still be difficult and expensive in many situations. In the case of systems based on Linux, the increasingly popular FOSS operating system (05), there are currently over we independent “distributions,” all based on the same core software (the |Jnux kernel). While application software will not always automatically run across all versions, all of the major distributions certify their products to the |Jnux standards Base (LSB), a set of standards currently maintained by the Linux Foundation, in order to permit application software to run more interoperably across compliant distributions.
      3. Security: While it may seem counterintuitive that code Vlslble to anyone anywhere would be safer to use, popular FOSS programs are acknowledged to be more secure, largely for the same reasons just stated: because anyone can see the code, anyone can tmck down the source of a vulnerability, let project managers know of the cause of concern, and/or propose a ﬁx herself. As a result, security issues can typically be identiﬁed, ﬁxed, and propagated to all users faster than ﬁaws in proprietary code.‘ As a result, FOSS is increasingly being used by defense, ﬁnancial and other types of users where security concerns are greatest.’
   3. K Ven, J Verelst, H Mannaert - Software, IEEE, 2008
      1. Proponents argue that making source code available lets everyone peer review the code, resulting in higherquality software. It’s also suggested that it gives users more choice and control because it lets them read and modify the source code. Although many OSS advocates have proclaimed these advantages, several authors have questioned or cast doubt on them.
   4. Garger, 2010
      1. From a developer’s point of view, open source technologies provide a platform on which to build custom solutions. Rather than develop, say, an entire proprietary operating system, developers can build and improve upon the already existing open source Linux operating system platform. In this way, more time is spent on the specific task to be solved rather than getting the system up and running to the point where it is stable and reliable.
      2. The biggest advantage of open source for users is that most projects are free to download and use. Without the pressure to show a profit from their work as in a large corporation, open source developers can work more toward end users’ individual needs and less on appealing to the masses. The result can be fewer bugs and more productivity due to the semi-custom solution.
3. Disadvantages
   1. Hammond, S 2009
      1. Disadvantages: the issue of ongoing maintenance and support.
      2. Support and IP issues: Lee pointed out another issue with OSS: the open nature of the code which can cause confusion on intellectual property issues. "With the GPL (GNU General Public License, see sidebar "What's a GNU?"), be careful if you develop some of your IT on top of the public code. But dynamic linking means that you link your code in runtime rather than compile time, and there are services for OSS audits, to ensure that you're in compliance. You don't want to give up your competitive edge."
      3. Another issue that concerned Lee is that "the OSS community seems to be interested in fashionable tech—they develop a 'hot area' for a few months, then it's out of fashion and they work on something else, and what can you do? It's like you're left with an obsolete technology— you can't control your destiny."
   2. K Ven, J Verelst, H Mannaert - Software, IEEE, 2008
      1. In the first scenario, the source code’s availability is neither an advantage nor a disadvantage for the organization.
         1. Half of the organizations in our sample indicated that they didn’t consider source code availability to be an advantage and that they never used the source code. This is consistent with other studies in this field.2,4–6 Joseph Feller and Brian Fitzgerald labeled this the “Berkeley Conundrum,” which questions the importance of the source code’s availability if no one actually uses it.13 At least two factors can account for this observation:
         2. Our study focused on highly mature infrastructure software such as Linux and Apache. Organizations have little need to modify the source code of such applications.2,5,6 Other types of OSS might give different results.
         3. Few—even experienced—programmers can modify the source code of mature software such as Linux and Apache.2
         4. In this scenario, OSS serves as a black box, and its advantages and disadvantages are comparable to proprietary packaged software.
      2. In the second scenario, the organization considers the source code availability to be an advantage, but doesn’t use it to study or customize the program.
         1. Some organizations in our case study expressed a greater trust in OSS because of the source code’s availability. They felt that the program was less likely to contain hidden features and that bugs in the software would be quickly fixed. In addition, the source code arguably gives organizations more control over their IT infrastructure. An organization can access (portions of) the source code of proprietary applications, either through vendor programs (such as those of Microsoft and Oracle) or through escrow agreements. However, vendors generally limit the organization’s rights. OSS gives organizations full access to the source code and generally places no restrictions on their right to modify or redistribute it. So, any interested party can obtain the source code and further develop, maintain, distribute, and support the software.
         2. Although organizations in this scenario might not actually use the source code, its availability gives them the option of doing so later. In this respect, the use of OSS implies a learning process, in which the organization gains experience and skills in OSS. This process can, however, be a considerable investment. Depending on the modification’s nature and the application’s modularity, developers might have to study a considerable portion of the source code, even for a limited modification.
      3. In the third scenario, OSS serves as a white box. Organizations can use the source code to study the software’s inner workings or to adapt the software to their own needs. This is primarily interesting for organizations developing OSS-based applications. In this case, the source code availability can be an important factor in the adoption decision. Three organizations in our sample indicated that although they didn’t modify the OSS, the source code’s availability let them better understand the OSS components’ inner workings. This helped them locate errors in the software developed on top of OSS. Organizations might also customize the software. For example, two organizations in our sample customized their web mail applications.
   3. Garger, 2010
      1. Unfortunately, this means that other developers have access to the developer’s work and can easily create a competing product. One of the reasons why Microsoft has such a huge market share of the operating system industry is because its software operates on a closed source paradigm; no one but those people privy to the source code can customize Windows. In addition, developers must be prepared to have their work publicly scrutinized. Almost everything in open source is viewable by other developers. If a developer makes a major mistake, it can become public information and tarnish the developer’s reputation.
      2. However, one con to using open source technology is that the focus is often on backend processing of information and not on user interfaces. Microsoft Windows has arguably one of the easiest interfaces with which to work. Often, open source software such as Linux requires the user to have specialized knowledge that cannot be configured with just clicks of a mouse. In addition, open source projects often do not have good documentation to walk the user through the learning and using of the technologies.
      3. In the case of Linux, for example, many hardware manufacturers seem to make their drivers and software work on Windows platforms first and consider other operating systems as an afterthought. The result can be poor support and more of a reason to switch to a better supported, yet closed source solution such as Windows.
   4. Ask.com
      1. May not offer commercial support
      2. Documentation might be less robust or more cryptic
      3. No guarantee that a project will continue as long as you need it
      4. More choices can lead to fragmentation of community support
   5. Lead Commerce, 2012
      1. Staffing Requirements
         1. Obviously most small businesses who end up choosing an open source ecommerce solution will have to maintain and potentially enhance their system. This will require staffing in order to make sure that the system is optimized and running properly. This type of staff can be difficult to source and expensive to a companies bottomline.
      2. Security Vulnerabilities
         1. Most open source software can not only be downloaded and used by people with good intentions but also by malicious hackers as well. Its important for business owners to realize that open source solutions provide the complete "blue prints" to how the system works and operates; which is like giving a thief keys to your home in order to break in and steal your valuables.
      3. Hosting
         1. Any open source technology will require hosting which depending on your needs can increase the total cost of ownership. Any company that chooses to host themselves should make sure that they go with a reputable hosting provider and look out for overage charges on space and bandwidth packages.
      4. Upgrading Issues
         1. When your open source ecommerce software vendor releases updates to their platform, these usually contain changes to security vulnerabilities which should be implemented in a timely manner. These "upgrades" can have issues working with 3rd party modules or "apps" that you might have installed previously causing more time for your development staff or having to bring in expensive consultants.
      5. Support
         1. In most cases open source solutions offer little to no support unless you are willing to pay for it. This is normally how the company makes their money. Meaning, the company charges substantial rates per incident or requires you to be on some kind of monthly or yearly support contract. In either case, this is an important factor to consider when choosing to go with an open source ecommerce product.
4. Conclusion
   1. Garger, 2010
      1. Open source has a number of advantage and disadvantages from both the point of view of developers and the point of view of users. These pros and cons must be weighed carefully before switching from a better-supported close source solution to a riskier open source technology. In the long run, the end user is the ultimate judge of which is a better solution. Without support from the user, developers and open source projects cannot continue except as a hobby or personal challenge for the developers.

References:

Gacek, C. & Arief, B. (2004),'The Many Meanings of Open Source', *IEEE Software*, 21, 1, pp. 34-40, Business Source Complete, EBSCOhost, (Accessed: 15 June 2014)

Garger, J. (2010) ‘What are the Pros and Cons of Open Source?’, *Bright Hub* [Online]. Available from: <http://www.brighthub.com/computing/linux/articles/62136.aspx> (Accessed: 14 June 2014)

Hammond, S 2009, 'Open source 2009. (cover story)', Computerworld Hong Kong, 26, 6, pp. 12-14, Computers & Applied Sciences Complete, EBSCOhost, viewed 14 June 2014.

Jaisingh, J, See-To, E, & Tam, K 2008, 'The Impact of Open Source Software on the Strategic Choices of Firms Developing Proprietary Software', Journal Of Management Information Systems, 25, 3, pp. 241-275, Business Source Complete, EBSCOhost, viewed 14 June 2014.

Lead Commerce (2012) *The Pros & Cons of Open Source eCommerce Software* [Online]. Available from: <http://www.leadcommerce.com/blog/the-pros-cons-of-open-source-ecommerce-software.html> (Accessed: 14 June 2014)

Opensource.com (2014) *What Is Open Source*? [Online]. Available from: <http://opensource.com/resources/what-open-source> (Accessed: 14 June 2014)

Open Source Initiative (2014) *The Open Source Initiative* [Online]. Available from: <http://opensource.org> (Accessed: 14 June 2014)

University of Liverpool/Laureate Online Education (2014) *DQ2 from Programming the Internet Module Seminar 4* [Online]. Available from: University of Liverpool/Laureate Online Education Blackboard (Accessed: 12 June 2014).

Updegrove, A 2009, 'A Concise Introduction to Free and Open Source Software', Standards Today, 8, 5, pp. 8-46, Business Source Complete, EBSCOhost, viewed 14 June 2014.

Get from refworks:

Should you adopt open source software?

K Ven, J Verelst, H Mannaert - Software, IEEE, 2008 - ieeexplore.ieee.org

... Table 1 Claims and counterclaims about open source software ... is neither an advantage nor a

disadvantage for the organization. ... In this scenario, OSS serves as a black box, and its advantages

and disadvantages are comparable to proprietary packaged software.14 In the ...

Cited by 78 Related articles All 8 versions Cite Save More