NATIONAL CONFERENCE ON INNOVATIVE TRENDS IN ENGINEERING & TECHNOLOGY – NITET-19
15-16th March 2019
NOVATEUR PUBLICATIONS
International Journal Of Innovations in Engineering Research And Technology [IJIERT]
ISSN: 2394-3696

ISSUES AND CHALLENGES OF OPEN SOURCE TECHNOLOGIES IN FINANCIAL SYSTEMS

Anil Kumar Patil,
Dept. of Computer Engineering. AGPIT, Solapur
E-Mail: anils3.kumar@gmail.com

Abstract

Open source software is worming its way into the local banking industry. Financial institutions are using them to balance accounts, calculate payrolls and trade on the stock market. Because of the stiff reliability and scalability requirements, only the most competent enterprise vendors have survived for long in this rarefied atmosphere. The purpose of this article is to analyze and understand the use of open source in banking.

Introduction

Open source refers to a program in which the source code is available to the general public for use and/or modification from its original design free of charge. Open source code is typically created as a collaborative effort in which programmers improve upon the code and share the changes within the community. Open source sprouted in the technological community as a response to proprietary software owned by corporations.

"Open source" is one of the most misunderstood concepts in the computing industry today. Most people in the industry, and even a number of people outside the industry, have heard of open source software. They may be able to name some of the most prominent open source projects, like Linux, Apache, and Firefox. But what most people don't understand about open source software is why it works. Software released as open source costs money and time to make and it has value.

The open-source model includes the concept of concurrent yet different agendas and differing approaches in production, in contrast with more centralized models of development such as those typically used in commercial software companies. a main principle and practice of open-source software development is peer production by bartering and collaboration, with the end-product, source-material, "blueprints", and documentation available at no cost to the public.

Most economists agree that open source candidates have a good information aspect. In general, this suggests that the original work involves a great deal of time, money, and effort. However, the cost of reproducing the work is very low, so that additional users may be added at zero or near zero cost – this is referred to as the marginal cost of a product. Copyright creates a monopoly so the price charged to consumers can be significantly higher than the marginal cost of production.

Use of Open source softwares

The open-source movement has inspired increased transparency and liberty in biotechnology research, it is argued that a trend toward democratized innovation in physical products is occurring like the free and open-source software movement, and that the difference between crowd sourcing and open source is that open-source production is a cooperative activity initiated and voluntarily undertaken by members of the public.

Computer software

- Blender is an open source 3D graphics editor.
- Ubuntu is a popular Linux-based open-source computer operating system.
- Android, the most popular mobile operating system

Open source software is software whose source code is published and made available to the public, enabling anyone to copy, modify and redistribute the source code without paying royalties or fees. Open source code can evolve through community cooperation. These communities are composed of individual programmers as well as very large companies. Many of these individuals programmers who start an open source project usually end up as large companies with open source programs.

Examples of open-source software products are:

Application software

- Zip – file archiver
- Blender – 3D graphics editor
- Eclipse – development environment comprising an IDE
- GIMP – graphics editor
- Inkscape – Vector graphics editor for .svg
- Mozilla Firefox – web browser
- Chromium – web browser
- Mozilla Thunderbird – e-mail client

- NASA World Wind – virtual globe, geobrowser
- OpenOffice.org (and the LibreOffice fork) – office suite
- PrestaShop – Electronic commerce platform

Operating systems

- Android – operating system derived from Linux
- FreeBSD – operating system derived from Unix
- Linux – family of Unix-like operating systems
- OpenIndiana – a free Unix-like operating system
- ReactOS – operating system built on Windows NT architecture
- Haiku – free and open source operating system compatible with BeOS

Server software

- Apache – HTTP web server
- Drupal – content management system
- MediaWiki – wiki server software, the software that runs Wikipedia
- MongoDB – document-oriented, non-relational database
- Moodle – course management system or virtual learning environment

Media

Open-source journalism, referred to the standard journalistic techniques of news gathering and fact checking, and reflected a similar term that was in use from 1992 in military intelligence circles, open-source intelligence. It is now commonly used to describe forms of innovative publishing of online journalism, rather than the sourcing of news stories by a professional journalist. In the December 25, 2006 issue of TIME magazine this is referred to as user created content and listed alongside more traditional open-source projects such as OpenSolaris and Linux.

The Open Source in Banks

While many banks still ponder the benefits of using open source technology for their coding needs, nascent *BankSimple* has gone full steam ahead. The start-up bank, which is testing its product with select customers, uses open source platforms for the development of nearly all of its Web and mobile applications as well as the backend systems that will power the bank when it finally throws the on switch for the 100,000 people who have signed up for its branchless services.

Just about every large bank, whether directly or through vendors, uses open source applications and platforms today. The benefits are substantial: Banks can potentially save 80% on project costs, which they can complete more quickly, because no one vendor is developing the code.

But working in an open source environment isn't a panacea either. In fact, it brings a critical set of management issues to which banks need to pay attention. Banks have to carefully examine licenses for open source code, for example. And they must pay strict attention to upgrades, patches and security. Finding knowledgeable coders can also be an issue, and because open source is based on communities, banks have to learn how to share part or all of the code they are developing. There is a critical mass and virtuous cycle for those contributing to an open source community, and you spread the cost of development over time to a broader audience than just your company.

Open source software is code that is available to the public to use and change, free of charge under an open license. Communities typically develop the software collaboratively. And the theory goes that because so many eyes are on the code, it is more stable and more secure, as opposed to proprietary code where only the project team gets to see it. Companies using open source must change their way of thinking about code and coding. As opposed to the typical vendor relationship, where the customer can pay the vendor to work on fixes, work in open source communities is collaborative. When there's a problem, you have to rely on the community to provide an answer. Users of open source are also expected to contribute code, or something equivalently useful, back to the community. There is a supplier-customer relationship that people know how to deal with in the commercial world, but in the open source world, no money exchanges hands. There are thousands of open source projects, and about 70 platforms licensed by the Open Source Initiative, a nonprofit group overseeing standards and licenses. However, banks only use about half a dozen of those. The most popular platforms include the Linux operating system.

Banks and financial institutions are looking at the competitive advantages from open source and they are not afraid of getting their fingernails dirty with the code because there is now a very sophisticated and mature development environment.

Bank of America, which has been using open source development since 2005, says the open source market is mature and can save programmers time and money, freeing them from reinventing the wheel on non-critical code. Open source communities are pushing innovations and they are no longer relevant solely for delivering alternatives to commodity products.

The biggest management issues around using open source have migrated over time . Initially they were about accountability. As opposed to the traditional vendor relationship, The current issues are more familiar to those using open source. Licensing, for example, has to be handled extremely carefully. Open source licenses can be distributed in one of two ways. The first is called a General Public License, or GPL, which links any code you develop to the open source and makes it available to the community, as opposed to more restrictive licenses that let you maintain more of your code by overlaying it on the open source platform.

Handling copyright issues

Many FOSS projects face similar copyright issues. Proper understanding of these issues when the project is young can help avoid problems later. This document provides general information, and not legal advice. If your FOSS project has a specific need for legal advice, please contact the Software Freedom Law Center or seek other legal counsel.

All software is subject to copyright law. The moment you save code to a file, copyright law gives you certain rights to control what other people can do with your work. Because almost everybody who contributes code to a software project has rights with respect to their code, understanding the basics of copyright is essential to running a FOSS project.

A software copyright is the exclusive legal right to control the rules for copying, modifying, and distributing a work of software. A person, company, foundation, trust, or other legal entity who has these exclusive legal rights is called a "copyright holder". Legal rules prohibit non-copyright holders from copying, modifying or distributing copyrighted works without permission from the copyright holder. Copyright holders can permit other people to copy or modify their software. That permission can be as simple as a perpetual, unconditional and universal grant of permission to do any of the acts that are exclusive to the copyright holder.

Other licenses are conditional. They allow people to copy or modify software only if certain conditions are met. If you don't meet the conditions, you don't have permission to copy or modify the software. If you make copies or distribute modified versions of the software without satisfying the conditions i.e. without permission, you infringe the copyright, which gives the copyright holder access to certain legal remedies. In particular, the copyright holder can sue you for damages or ask a court to order you not to make or distribute further copies. It is important for projects to understand the conditions in their licenses as well as those in the licenses of code they link to and code they incorporate into their project. Complying with the conditions in the license is essential to avoiding copyright infringement.

Risks in using open source software

The following are certain risks in using the open source. Some of the risks mentioned below are inherent while the other risks might arise due to poor software management practices.

- Absence of meticulous evaluation
- Spurious open source
- Lack of sponsorship

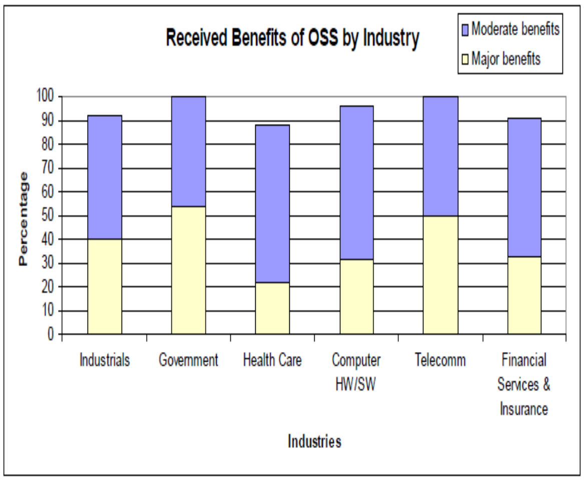
Guidelines for deploying open source software in an enterprise environment

Here are some guidelines that could be considered before deploying any open source software in an enterprise environment

- Security policy
- Evaluation
- Avoid ad-hoc installations
- Download open source software only from trusted sites
- Prefer source code to binaries wherever possible
- Scan for vulnerabilities
- Disable unwanted services
- Have Defense-in-Depth strategy
- Install and forget model is very dangerous
- Training and documentation are important
- Consider open source software in DR and BC plans

Benefits to industry

Industries are getting the greatest benefits from open source software. While asking respondents whether they are getting no, minor, moderate or major benefits does not allow for a fully quantitative comparison, it does show some interesting trends (see Figure 1 – Benefits to Industry). Telecommunications and government organizations were much more likely to cite major benefits from open source . Health care and financial institutions were less likely to report major benefits, although a sizable percentage (at least 20% of the health care companies and more than 30% of the financial institutions) reported major benefits from open source.



Benefits of open source security

- More people can inspect the source code to find and fix a possible vulnerability.
- Proprietary software forces the user to accept the level of security that the software vendor is willing to deliver and to accept the rate that patches and updates are released.
- The end-user of Open Source code has the ability to change and modify source to implement any extra "features" of security they may wish for a specific use, which can extend to the kernel level if they so wish.

Drawbacks of open source security

- All people have access to the source code, including potential attackers. Any unpatched vulnerability can be used by attackers.
- Having a large amount of eyes reviewing code can "lull a user into a false sense of security". Having many users look at source code does not guarantee that security flaws will be found and fixed.

Conclusion

The open source trend would keep turning the chunks of IT infrastructure into commodities by offering alternate solutions to proprietary software. As this trend continues, the enterprises would have equivalent or better open source alternatives available for their enterprise requirements. The end users would continue to look out for security assurance in open source products before considering them for mission-critical enterprise requirements. There are greater chances for most of the vendors to change to a support/service model from their ownership model by offering various support services for open source products. As part of this effort, the popular open source products would receive sponsorship from various vendors to undergo rigorous security evaluation and certification. The enterprises should do an extensive risk and security analysis before choosing open source solutions over their closed source counterparts. The analysis should consider various factors such as the expertise available in house and the support options available for the respective open source product. Well documented and implemented security policies and best practices help an enterprise to mitigate the risks and enjoy the real benefits of open source.

References

1. "Web Power to the People," The Boston Herald, September 29, 2005
2. "The Glory Days of the Rock of Boston," Boston Globe, Op-Ed, July 18, 2009.
3. 'BCN Documentary Looks for Archival Contributors," Boston Herald, August 11, 2009
4. "WBCN and The American Revolution," The Huffington Post, August 11, 2009
5. Apache HTTP Server Project Guidelines and Voting Rules Foundation 2004.
6. Open-source license - Wikipedia, the free encyclopedia.htm
7. OpenLearn - Wikipedia, the free encyclopedia.htm

8. Open Source Banking Application – Prog. Nairaland.htm

9 ^ Raymond, Eric S. The Cathedral and the Bazaar. ed 3.0. 2000.

10 ^ "Science 2.0 is here as CSIR resorts to open-source drug research for TB" Business Standard, 1 March 2009

11 ^ a b c d James J. Flink (1977). The Car Culture. MIT Press. ISBN 0-262-56015-1.

12 ^ Fisher, Franklin M.; James W. McKie, Richard B. Mancke (1983). IBM and the U.S. Data Processing Industry: An Economic History. Praeger. pp. 172–9. ISBN 0-03-063059-2. IBM unbundled (began charging for) software June 23, 1969

13 ^ Dave Pitts' IBM 7090 support – An example of distributed source: Page contains a link to IBM 7090/94 IBSYS source, including COBOL and FORTRAN compilers.

14 ^ a b c Tiemann, Michael (September 19, 2006). "History of the OSI". Open Source Initiative. http://www.opensource.org/history. Retrieved August 23, 2008.

15 ^ Muffatto, Moreno (2006). Open Source: A Multidisciplinary Approach. Imperial College Press. ISBN 1-86094-665-8.