Free and Open-source Software

Unit I

Introduction Introduction: Open Source, Free Software, Free Software vs. Open Source software, Public Domain Software, FOSS does not mean no cost. History: BSD, The Free Software Foundation and the GNU Project.

Methodologies Open Source History, Initiatives, Principle and methodologies. Philosophy : Software Freedom, Open Source Development Model Licenses and Patents: What Is A License, Important FOSS Licenses (Apache,BSD,GPL, LGPL), copyrights and copy lefts, Patents Economics of FOSS : Zero Marginal Cost, Income-generation opportunities, Problems with traditional commercial software, Internationalization

Social Impact Open source vs. closed source, Open source government, Open source ethics. Social and Financial impacts of open source technology, Shared software, Shared source, Open Source in Government.

Introduction

Open Source

1) In general, open source refers to any program whose source code is made available for use or modification as users or other developers see fit. Open source software is usually developed as a public collaboration and made freely available.

2) Open Source is a certification mark owned by the Open Source Initiative (OSI). Developers of software that is intended to be freely shared and possibly improved and redistributed by others can use the Open Source trademark if their distribution terms conform to the OSI's Open Source Definition. To summarize, the Definition model of distribution terms require that:

- The software being distributed must be redistributed to anyone else without any restriction.

- The source code must be made available (so that the receiving party will be able to improve or modify it).
- The license can require improved versions of the software to carry a different name or version from the original software.

Need of Open Sources

Few reasons why you need an Open Source Strategy are:

1. **Reduce dependency on closed source vendors.** Stop being dragged through constant product upgrades that you are forced to do to stay on a supported version of the product.

2. **Your annual budget does not keep up with increases in software maintenance costs and increased costs of employee health care.** Your budget remains flat, you bought five new tools last year with new annual costs in the range of 18-20% of the original purchase price for "gold support", and your employees' health care costs shot up 25% again.

3. **More access to tools.** You can get your hands a variety of development and testing tools, project and portfolio management tools, network monitoring, security, content management, etc. without having to ask the boss man for a few hundred thousand green backs.

4. **Try before you buy.** Are you getting ready to invest in SOA, BPM, or ECM? Why not do a prototype without spending huge sums of money? First of all, it allows you to get familiar with the tools so you can be educated when you go through the vendor evaluation process. Second of all, you might find that the tool can do the job and you don't need to lock yourself in to another vendor.

5. **Great support and a 24/7 online community that responds quickly.** Despite the myths that you can't get support for open source software, the leading communities provide support far superior to most closed source vendors. Most communities have a great knowledgebase or wiki for self service support. You can also post a question and one of the hundreds of community members throughout the world will most likely respond in minutes. Make sure you chose software with strong community backing.

6. **Access to source code and the ability to customize if you desire.** You can see the code, change the code, and even submit your enhancements and/or fixes back to the community to be peer reviewed and possibly added to the next build. No longer do you need to wait for a vendor roadmap that doesn't have the feature you need until their Excalibur release in the fall of 2009.

7. **Great negotiating power when dealing with closed source vendors.** Tired of vendors pushing you around because you don't have options? I wonder if companies like Microsoft would be more willing to be flexible with their pricing if you have 20 desktops running Ubuntu as an alternative desktop pilot initiative.

8. **Feature set is not bloated and is driven by collaboration amongst the community.** Tired of products that consume huge amounts of memory and CPU power for the 2000 eye candy features that you will never use? With open source software, most features are driven by community demand. Closed vendors have to create one more feature then their competitors to get the edge in the marketplace.

9. **Bug fixes are implemented faster then closed source vendors.** Actually, many bugs are fixed by the community before they are even reported by the users.

Open Source Software:

Open source software is software that provides access to the source code, meaning that users are free to see how the product is made. Additionally, users have the right to modify the product (change the code) to their liking, experiment with different versions, and give away or resell the new product with the guarantee that they must also provide their source code, and so on. Modifying the product and redistribution are the two main components of open source.

- software.Source code is available
- Make a changes or Modification
- It is released in 1998

Advantages: 1) Transparency : We get fully visibility into the code base.

2) Low Cost

3) Reliability : code is developed on online forums and guided by experts.

4) Better Quality

5) More Flexibility

Example :

1. Mozilla's Firefox web browser.

2. PHP scripting language.
3. Python programming language.
4. Apache HTTP web server.
5. database system.

Reasons to Use Open Source Software

- It promotes creative development
- Those who can't afford proprietary software can download open source programs for free
- Money saved can be used to purchase other needed materials
- Can easily modify your software to suit patron's needs and your needs
- Little to no upgrade costs
- No more grueling over software that doesn't meet your standards -- create it yourself based off of a close preexisting piece of software
- The price (free) makes it easier to change your mind when the software doesn't live up to its expectations
- Little to no viruses!

Why do people prefer using open source software?

1) **Control** : People can examine the code to make sure it's not doing anything they don't want it to do, and they can change parts of it they don't like. Users who aren't programmers also benefit from open source software, because they can use this software for any purpose they wish–not merely the way someone else thinks they should.
2) **Training** : It helps them become better programmers. Because open source code is publicly accessible, students can easily study it as they learn to make better software. Students can also share their work with others, inviting comment , as they develop their skills.
3) **Security** : Some people prefer open source software because they consider it more secure and stable than proprietary software. Because anyone can view and modify open source software, someone might spot and correct errors or omissions that a program's original authors might have missed.

Free Software

- Free software is a term coined and promoted by the GNU project and Free Software Foundation.

- It is used for open source software which allows users to freely look at the source of the software, modify it, distribute it, and use it without any restriction.
- Software available without any payment is not necessarily free software.
- Freeware is software that you don't have to pay for.
- Free software is software you are free to modify and use for your own purposes.
- Freeware does not necessarily have to be free software as the source code can still be protected.
- The Free Software Foundation provides these "four essential freedoms" that software must respect in order to be considered free:

The freedom to run the program, for any purpose (**freedom 0**).

The freedom to study how the program works, and change it so it does your computing as you wish (**freedom 1**). Access to the source code is a precondition for this.

The freedom to redistribute copies so you can help your neighbor (**freedom 2**).

The freedom to distribute copies of your modified versions to others (**freedom 3**). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

Four Freedoms taken from the GNU Operating System

Example :

1. Linux Operating System
2. Send Mail Transport
3. MySQL
4. Mozilla Firefox web Brower.

Free Software vs. Open Source software

Sr. No.	Free Software	Open Source Software
1.	Free Software foundation had introduced one term in 1986 called as free software.	Open source term released in 1998.

2.	Free software is called as open source software and it is a free.	Open source software is not necessary to be free software.
3.	Freely available but copyrighted.	Freely available source also available(can be edited to make your own copyright)
4.	Free software gives just freedom to users.	The purpose of open source software is to maintain the security.
5.	Free software as being software that ensures the end users has freedom in using, studying, sharing and modifying the software.	Open source software is software, which is provided a source code freely for study and everyone to inspect.

For more example follow this link :

http://adciv.org/Examples_of_free_and_open-source_software

Public Domain Software

- Public-domain software is software that has been placed in the public domain.
- There is absolutely no ownership such as copyright, trademark, or patent.
- Public domain software is any software that has no legal, copyright or editing restrictions associated with it. It is free and open-source software that can be publicly modified, distributed or sold without any restrictions.
- Public domain is a legal term that describes a work or product that is not protected by copyright.
- The copyright protection an item in the public domain may have :1) expired, 2) been released by the author, or 3) never existed in the first place.
- Public domain items are publicly available and can be freely accessed and redistributed.
- Many different items can be labeled as "public domain."
- For example, books, speeches, poems, artwork, songs, and videos can all be made freely available to the public. In the computing world, "public domain" is often used to refer to software programs that are offered to the public without copyright restrictions.
- Public domain software is similar to open source software, in which the source code of a program is made publicly available. Public domain software is also similar to freeware, which refers to software offered at no charge.
- EXAMPLE SOFTWARE

1. SQLite

2. I2P
3. CERN's httpd
4. BLAST

- Public domain software is any software that has no legal, copyright or editing restrictions associated with it.
- It is free and open source software that can be publically modifie distributed or sold without any restriction.
- There are no licensing requirements with public domain software.

FOSS does not mean no cost

- This is a common misunderstanding about FOSS, in no small part because nearly all FOSS programs are available free of charge.
- For example, when the text editor Emacs was first released Richard Stallman charged from time to time to get copies.
- Developers have the choice to charge under most FOSS licenses, although they rarely choose to.
- The only requirement to be a truly FOSS project is that the publisher provides the source code with the program, and to allow the user to edit that code.
- On top of the initial cost of purchasing software, there are other ongoing costs associated with all software. This can come in the form of support agreements, the cost of customization, training costs support personnel and other sources. This is true of both traditional commercial software and FOSS programs.
- There is a large and active debate about which type of software is more expensive over the long run for large corporations, for individual users there is little to no question that FOSS is cheaper by far.

Extra Notes: https://en.wikibooks.org/wiki/Open_Source

History:

BSD – Berkeley Software Distribution

- Berkeley Software Distribution (BSD) was a Unix operating system derivative developed and distributed by the Computer Systems Research Group (CSRG) of the University of California, Berkeley, from 1977 to 1995.
- BSD is a branch of the family of UNIX like operating systems. BSD was initially called Berkeley Unix because it shared the same source code with AT&T Research Unix.
- In the 1980s, BSD was widely adopted by workstation vendors in the form of proprietary Unix variants such as DEC ULTRIX and Sun Microsystems SunOS, due to its permissive licensing, and its familiarity to many technology company founders and engineers.
- When these proprietary BSD derivatives were largely superseded by the UNIX System V Release 4 and OSF/1 systems in the 1990s.
- Example: FreeBSD, OpenBSD, NetBSD, DragonFly BSD, Darwin, and TrueOS.
- **FreeBSD** aims for high performance and ease of use by end users, and is a favourite of web content providers. It runs on a number of platforms and has significantly more users than the other projects.
- **NetBSD** aims for maximum portability: “of course it runs NetBSD”. It runs on machines from palmtops to large servers, and has even been used on NASA space missions. It is a particularly good choice for running on old non-Intel® hardware.
- **OpenBSD** aims for security and code purity: it uses a combination of the open source concept and rigorous code reviews to create a system which is demonstrably correct, making it the choice of security-conscious organizations such as banks, stock exchanges and US Government departments. Like NetBSD, it runs on a number of platforms.
- **DragonFlyBSD** aims for high performance and scalability under everything from a single-node UP system to a massively clustered system. DragonFlyBSD has several long-range technical goals, but focus lies on providing a SMP-capable infrastructure that is easy to understand, maintain and develop for.
- BSD stands for “Berkeley Software Distribution”. It is the name of distributions of source code from the University of California, Berkeley, which were originally extensions to AT&T's Research UNIX® operating system. Several open source operating system projects are based on a release of this source code known as 4.4BSD-Lite. In addition, they comprise a number of packages from other Open Source projects, including notably the GNU project. The overall operating system comprises:
- The BSD kernel, which handles process scheduling, memory management, symmetric multi-processing (SMP), device drivers, etc.
- The C library, the base API for the system.
- The BSD C library is based on code from Berkeley, not the GNU project.
- Utilities such as shells, file utilities, compilers and linkers.
- Some of the utilities are derived from the GNU project, others are not.
- The X Window system, which handles graphical display.

- The X Window system used in most versions of BSD is maintained by the X.Org project. FreeBSD allows the user to choose from a variety of desktop environments, such as Gnome, KDE, or Xfce; and lightweight window managers like Openbox, Fluxbox, or Awesome.
- Many other programs and utilities.

The Free Software Foundation and the GNU Project.

- In 1983, Richard Stallman become discouraged with the increasing commercialization of the computer development work that had been done at MIT and with growing limitations demanded on software users.
- After that he started to create software that gave control to users. His aim was to create an entire operating system totally free of the restrictions being demanded by proprietary licensing.
- 1985 Free Software Foundation (FSF) founded by Richard Stallman. Along with other programmers creates the tools needed to make a UNIX compatible OS
- 1985 Professor Andy Tannenbaum creates a UNIX like operating system based on System V Unix for the IBM PC & PC/AT computers. It is called Minix.
- 1989 Richard Stallman releases GPL and GNU software but lacks a free kernel.
- 1991 Building on the concepts in Minix, Linus Torvalds (Finnish college student) develops Linux along with help from other users on the web.
- GNU software guarantees these freedom-rights legally (via its license), and is therefore free software; the use of the word "free" always being taken to refer to freedom.
- When the GNU project first started they had an Emacs text editor with Lisp for writing editor commands, a source level debugger, a yacc-compatible parser generator, and a linker.
- GNU goal was to build a free complete UNIX-like system.
- The Free Software Foundation (FSF) is a non-profit organization founded by Richard Stallman on 4 October 1985 to support the free software movement, which promotes the universal freedom to study, distribute, create, and modify computer software,with the organization's preference for software being distributed under copyleft ("share alike") terms,such as with its own GNU General Public License.
- FSF's funds were mostly used to employ software developers to write free software for the GNU Project.
- The FSF maintains a list of "high priority projects" to which the Foundation claims that "there is a vital need to draw the free software community's attention".
- The FSF considers these projects "important because computer users are continually being seduced into using non-free software, because there is no adequate free replacement.

❖ GNU project – “GNU is not Unix”
  – Goal of completely new Unix-like system, started in 1984 by Stallman
    • Reaction against commercial control of other Unix versions (ATT, BSD)
  – Largely developed applications and API’s
    • Text editor, C/C++ compiler, etc.
    • Widely used by Unix software developers but short of goal

- ❖ Free Software Foundation (FSF)
  - Outgrowth of GNU project in 1985
    - Developed and promoted GPL

- A few of the most common acronyms are:

Acronym	Meaning
GNU	GNU's Not Unix (a project to create an FLOSS operating system)
GPL	GNU General Public License (the most common FLOSS license)
OS, OSes	Operating System, Operating Systems
FLOSS	Open Source Software/Free Software

Methodologies

Open Source History:

- Computer scientists working in higher education were the first to develop and freely share software in the 1950s.
- But as computing systems became more complex and capable, the costs of software development increased so that by the 1960s, computer hardware companies were charging for software that was bundled with their products.

- During the 1970s Unix, an operating system that could run on multiple computer platforms, became popular in academic circles. Ultimately, this led to large-scale adoption of Unix by commercial startups, most notably Solaris, HP-UX and AIX. Among the many variants of Unix, Mac OS X is the most widely used.

- By the 1980s, the importance of software became clear and many technology leaders began speaking out against the ever-increasing costs associated with it.

- The GNU Project (link is external) was launched in 1984 to create a complete computer operating system that was free from constraints on use of its source code, and also in part to protest the costs and limitations of commercial software.

- In 1991, the Linux operating system emerged under the GNU Public License. Millions use GNU/Linux today, though many refer to it as simply, Linux.
- The term "open source software" was coined in 1998 when Netscape finally released the source code for its cutting-edge web browser, in hopes of improving it by allowing more people to find and fix bugs. The attention surrounding the Netscape release created the opportunity to educate and advocate for the superiority of an open development process.
- Over 60 percent of higher education institutions use open source solutions, both on the server and on the desktop. Why? Because they want flexibility and control. Cost savings is another big reason, which can reach six figures over the typical three-year university contract term.