Open Source Software: Overview, Legal and Business issues

Timo Tuunanen

TietoEnator
Building the Information Society

TietoEnator © 2003

Open Source?

TietoEnator
Building the Information Society

TietoEnator © 2003

OS history and definitions

- Is based on software culture in 60’s – 70’s
- Some important OS milestones
  - 1984 Richard Stallman founded Free Software Foundation
  - 1991 Linus Torvalds started Linux -project
  - 1998 Netscape Navigator goes Open Source
- Open Source is defined in Open Source Definition (OSD) http://www.opensource.org/docs/definition_plain.php
- Free Software is another term used. Free Software means basically the same but their interpretation is slightly tighter.

TietoEnator
Building the Information Society

TietoEnator © 2003

Open Source definition in brief

- Licensees are free to use Open Source software for any purpose whatsoever
- Licensees are free to make copies of Open Source software and to distribute them without payment of royalties to a licensor
- Licensees are free to create derivative works of Open Source software and to distribute them without payment of royalties to a licensor
- Licensees are free to access and use the source code of Open Source software
- Licensees are free to combine Open Source and other software
- Open Source is about the licenses (and development model)

TietoEnator
Building the Information Society

TietoEnator © 2003

Open Source projects

- There is a huge number of OS projects
  - http://sourceforge.net/ includes over 75 000 active projects
  - http://freshmeat.net/ includes over 38 000 active projects
- Majority of the contribution and developers are concentrating on most popular software
- Projects are done by the OS community
  - Project leaders
  - Core developers
  - Active developers
  - Random developers
  - Bug fixers
  - Bug reporters
  - Readers
  - Passive users (for example in case of Apache 99%)

TietoEnator
Building the Information Society

Open Source characteristics

- Number of developers in projects
  - Almost half of the projects have only one developer
  - 15 % 2-3 developers
  - 20 % 4-10 developers
  - 10 % 11-20 developers
  - 5 % more than 20 developers
- Programmers develop programs for other developers
- Over 80% of the software less than 50 000 LOC
- 60 % of the projects less than one year old
- More than 80 % of the software licensed under GPL or LGPL
- Different programming languages often used within one software

TietoEnator
Building the Information Society

Open Source characteristics cont.

- Documentation level varies
  - README (15%)
  - Man pages (45%)
  - API description or user documentation (40%)
- Most common projects (Linux, Apache, Mozilla..) however ARE NOT typical OS projects

TietoEnator
Building the Information Society

Type of OS software available

- 2/3 of the OSS is horizontal
  - Horizontal = programs that are used for creating other programs or system programs (In Sourceforge under: Internet, system, software development, database or security -sections)
- Only 1/3 of the programs are vertical applications
  - Vertical = programs that end user just uses

- Programs that “home user” can use, are likely to be found as OS
- Business software is less common

TietoEnator
Building the Information Society

OS quality and warranty

- Quality varies, but there is no evidence that OS method would produce better or worse quality software than commercial methods
  - Most common OSS however is generally high quality
- OS applications come without warranty.
  - THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. (Part of BSD -license)

TietoEnator
Building the Information Society

Open Source Motivations

- Private developers
  - Internal motivations
    - ”Coding is fun”
    - Learning
    - Altruism
    - Community membership
  - External motivations
    - Reward in the future
    - Personal needs
    - Recognition
- Foundations
  - Political, philosophical and commercial motivations
- Companies
  - Commercial motivations including time-to-market, image building, ”let the others do the work”, ...

TietoEnator
Building the Information Society

Intellectual property rights (IPR)

- IPR’s
  - Copyright (expression)
  - Patents (idea)
  - Trademarks
- In case of software a difference between expression and idea is often unclear

TietoEnator
Building the Information Society

Copyright

- All software is automatically covered by copyright
  - as long as work is copyrightable
- Copyright holder has the following rights to his work (covered by the international copyright laws)
  - Exclusive right to make copies
  - Exclusive right to prepare derivative works
  - Exclusive right to distribute copies of the original or derivative works
  - In case of literature, music, movies etc. you have an exclusive right to display the work publicly
- Other people/companies don’t have these rights and they are not allowed to perform these actions without copyright holders permission

TietoEnator
Building the Information Society

Copyright cont.

- “As to copyright, a single sentence is generally accepted to be too short to qualify for copyright protection. Still, at the same time, a haiku is likely to be protected even though it's easy to write single sentences that are longer than haikus.” Arnoud Engelfriet, debian-legal@lists.debian.org, 7.10.2005
- In case of software one can’t say the LOC that qualifies for copyright protection

TietoEnator
Building the Information Society

Patents

- Far more complicated to obtain than copyright
- Patent holder rights
  - Right to exclude others from making products embodying your patented invention
  - Right to exclude others from using products embodying your patented invention
  - Right to exclude others from selling or offering for sale products embodying your patented invention
  - Right to exclude others from importing products embodying your patented invention

TietoEnator
Building the Information Society

Trademarks

- Purpose is to differate from other products
- Can be owned, sold and licensed
- OS licenses don’t license trademarks
  - If you want to use Linux trademark in your product the license must be obtained from Linux Mark Institute

TietoEnator
Building the Information Society

License

- License is simply a permit to do something that is not legal otherwise (Driving license / Software license)
- Software license describes copyright and patent holders promise to use their intellectual property
- Open Source licenses guarantee certain rights to the user (see Open Source Definition)
- Open Source Initiative approves OS licenses. Approved licenses can be found from: http://www.opensource.org/licenses/
- By contrast commercial software licenses give users a limited right to use the program

TietoEnator
Building the Information Society

Open Source licenses

- License types
  - Academic licenses (for example BSD and MIT)
  - Reciprocal (for example LGPL ja MPL) licenses
    - Extreme cases are viral licenses (for example GPL)
  - Standard licenses
  - Content licenses (for example AFL)

TietoEnator
Building the Information Society

License comparison

Criteria	*Free Distribution*	*Free Use*	*Open Source*	*Reciprocal*	*Viral*	*Network usage*
Shareware	*X*					
Freeware	*X*	*X*				
BSD	*X*	*X*	*X*			
LGPL	*X*	*X*	*X*	*X*		
GPL	*X*	*X*	*X*	*X*	*X*	
OSL	*X*	*X*	*X*	*X*	*X*	*X*

Table by Mikko Välimäki

TietoEnator
Building the Information Society

Academic licenses

- Origins of these licenses are at universities
- Most common academic licenses: BSD, MIT, Apache
- Idea is to give the software to the users and let them use it any way they want
- Permission to re-license the software.
- Derivative works can be closed source (commercial) software
- Typical requirements are
  - Copyright information can't be removed
  - Names of the organizations that produced the software can't be used when promoting the software
  - In binary distributions, the copyright holders must be mentioned in documentation

TietoEnator
Building the Information Society

Reciprocal licenses

- Derivative works must be distributed under the same license as the original work
- In GNU world, a term Copyleft is often used: http://www.gnu.org/copyleft/copyleft.html
- Viral licenses require that all software that use the original work must be licensed under the same license
- LGPL and MPL are reciprocal but not viral licenses
- GPL is viral license

TietoEnator
Building the Information Society

GPL and LGPL

- GPL is the most common Open Source license
- Copyleft is the most important idea behind GPL license
- Iterpretations about GPL can be found from GNU web pages: GPL FAQ http://www.gnu.org/licenses/gpl-faq.html
  - According to GNU, programs that link GPL libraries must be under GPL license
  - Software compiled using GPL compiler can be licensed under any license
  - Same distribution (for example CD) can contain both GPL and otherwise licensed programs
- LGPL, Lesser General Public License (formerly Library General Public License)
  - Permits linking. A software that links against LGPL libraries can be licensed under any license

TietoEnator
Building the Information Society

License compatibility

- Licenses must be compatible in order to create works that contain software that is licensed under different licenses
- List of GPL compatible and incompatible licenses can be found from FSF web pages: http://www.fsf.org/licensing/licenses/index_html#SoftwareLicenses
- Known GPL incompatible licenses
  - Mozilla Public License (MPL)
  - Xfree 86 1.1 -license
  - Original BSD license
  - Apache license

TietoEnator
Building the Information Society

Mozilla Public License (MPL)

- MPL (and NPL) license is somewhere between BSD and GPL
- Reasons behind MPL creation can be read from Mozilla web pages: http://www.mozilla.org/MPL/FAQ.html
- MPL is high-quality, professional legal accomplishment in a commercial setting
- Acts as a basis for many other OS licenses
- License itself is long and difficult to understand (at least for software developer) but it's reciprocity provisions can be presented shortly: "If you create and distribute a Modification to one of the files containing Original Code or previous Modifications, those files must be released as Modifications under the same MPL license"

TietoEnator
Building the Information Society

Open Source business

- How do I make money out of Open Source?
  - Services
  - Books and seminars
  - Combining OS and commercial software
  - Support and consulting
  - Custom software development
  - Dual licensing
  - Development tools
  - Consumer devices
  - ...

TietoEnator
Building the Information Society

Potential and pitfalls

- Potential
  - OS provides a very rich base of existing software
  - OS is getting more and more popular
  - Maintenance may be done by someone else
  - Freedom from other companies
  - Quick time-to-market
- Pitfalls
  - Legal issues must be taken care of
  - Quality of each package must be examined
  - Old development and product processes must be enhanced
  - Trade secrets may become public

TietoEnator
Building the Information Society

Different approaches for Open Source usage

- Use OS as basis and combine it with proprietary software from someone else (Montavista, Trolltech)
  - Easiest and quickest way to start
- Use OS as basis and build your own software on top of it. Distribute derivative works only if someone asks for them
  - Requires more work, but full control achieved
- Be part of OS community: Take responsibility of some OS packages and keep only the critical components closed
  - Full control
  - Others benefit from your work
  - Easiest maintenance and control over OS packages
- In most cases some sort of combination is the best solution
  - However selection of approach is probably the most important decision if OS is used in product development
- Usually forking is not recommended (fork = using existing OS component and developing own version of it)

TietoEnator
Building the Information Society

Company development process using Open Source

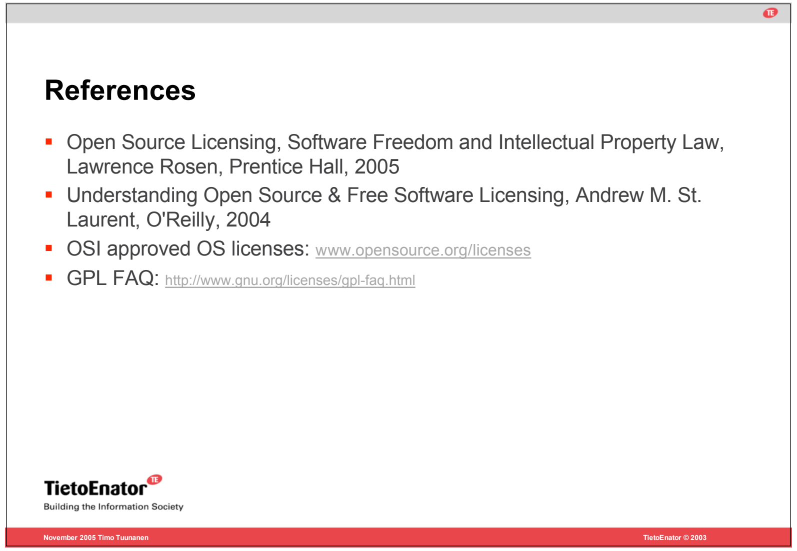
- Using Open Source should not differ from using any 3rd party component
  - Functionality of the component must be evaluated
  - Quality of each component must be ensured
  - It must be made clear that the component doesn't infringe patents or trademarks
  - In Open Source world the license check must be also conducted
- As an addition, designated engineers must follow the development process of key components
- Usually only stable packages should be selected

TietoEnator
Building the Information Society

References

- The Open Source Definition http://www.opensource.org/docs/definition.php
- Free Software Definition http://www.fsf.org/licensing/essays/free-sw.html
- Debian Free Software Guide: http://www.debian.org/social_contract#guidelines
- Categories of Free and Non-Free Software http://www.fsf.org/licensing/essays/categories.html
- Characteristics of Open Source Projects, A. Capiluppi, P. Lago, M. Morizio
- Working for free? - Motivations of Partisipating in Open Source Projects, A. Hars, S. Ou

TietoEnator
Building the Information Society



References

- Open Source Licensing, Software Freedom and Intellectual Property Law, Lawrence Rosen, Prentice Hall, 2005
- Understanding Open Source & Free Software Licensing, Andrew M. St. Laurent, O'Reilly, 2004
- OSI approved OS licenses: www.opensource.org/licenses
- GPL FAQ: http://www.gnu.org/licenses/gpl-faq.html

TietoEnator
Building the Information Society

TietoEnator © 2003