# Models of exploitation with open source

Wilson / Rahtz / Metcalfe

OSS Watch

# Who is/are OSS Watch?

OSS Watch provides unbiased advice and guidance about free and open source software for UK further and higher education.

- Funded by the Joint Information Systems Committee
- Based in the Research Technologies Service
- ...which is based in the Computing Services, University of Oxford

# What Does OSS Watch Do?

- Best practice guidelines
- Investigative reports
- Briefing materials
- National conferences
- Focused workshops
- Consultation

# It’s about the licence, stupid

- Open source software is open source because of the licence under which it is distributed
- The Open Source Initiative publishes a definition of ’open source’
- They also maintain a list of licences that meet the strictures of this definition
- There are currently 58 licences on this list

# It’s about the licence, stupid

The OSI’s Open Source Definition

- Freely Redistributable
- Source Code Included
- Derived Works Permitted
- Integrity of Author’s Source Code
- No Discrimination Against Persons or Groups
- No Discrimination Against Fields of Endeavour
- Distribution of Licence (Rights)
- License Must Not Be Specific to a Product
- License Must Not Restrict Other Software
- License Must Be Technology-Neutral (must be usable without ’click-wrap’ mechanism etc)

# It’s about freedom, stupid

The Free Software Foundation’s *Four Freedoms*

- The freedom to run the program, for any purpose
- The freedom to study how the program works, and adapt it to your needs. (Access to the source code is a precondition for this).
- The freedom to redistribute copies so you can help your neighbour
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. (Access to the source code is a precondition for this).

# It’s about community, stupid

- Distributed development model
- Communal support
- High-powered members as well as bedroom-bound geeks
- Security and stability gains

# What's the government's attitude?

- Open Source Software Use within UK Government (v2 October 2004)
- (http://www.govtalk.gov.uk/policydocs/)
- "Publicly funded Research and Development projects which aim to produce software outputs shall specify a proposed software exploitation route at the start of the project. At the completion of the project, the software shall be exploited either commercially or within an academic community or as OSS."
- ...although: "The policy on exploiting Research and Development software will not apply to software developed in the areas of defence, national security or law enforcement. It will also not apply to software developed by Trading Funds."

# What’s the funding bodies’ attitude?

- JISC has a published policy on OSS
- "Copyright of software, documentation, design materials, manuals, user interface and source code must be released under an OSI-approved open source licence, unless the bid explicitly argues why this should not be the case and proposes an alternative licence."
- Other funding bodies are currently discussing their policy reaction to the government’s policy document.

# Open Source Licences - Five Examples

- GNU General Public License v2
- GNU Lesser General Public License v2.1
- Modified BSD (Berkeley Software Distribution) License
- Apache License v2
- Mozilla Public License v1.1

# Common Features

They all

- allow anyone to distribute the software for a fee (or give it away) without royalty to the licensor
- allow modified versions of the software to be distributed by licensees (under varying terms depending on which licence is chosen)
- try to exclude liabilities to the extent possible under local laws

# Common Features

*’Public Licences’*

- Copyright law prevents the copying, distribution and/or modification of copyright works (subject to certain exceptions)
- The licence allows anyone to perform these activities under certain conditions
- A licensee who disclaims knowledge of the terms of the licence is acknowledging that they had no permission. There is no other route by which the software can be lawfully used.

# GNU General Public License v2

*Significant Features*

- All modified versions of GPL-licensed software must also be distributed under the GPL (if they are distributed at all)
- All modified versions must advertise prominently what has been modified, who modified it, and when it was modified
- Source code must be provided with all GPL-licensed software, either directly or via a request to the licensor

# GNU General Public License v2

*Significant Features*

- All licensees of the software gain their licence directly from the original licensor. This preserves the licensor's standing to take action against all licensees
- No redistributing licensee may impose further restrictions on recipients
- Additional restrictions placed on a licensee by a court mean that the licensee cannot distribute the software at all

# GNU General Public License v2

*What Does It Try To Do?*

- Allow free distribution and modification of the software
- Prevent modifiers distributing their modified versions in a non-free way

# GNU Lesser General Public License v2.1

*What's A Software Library?*

- A collection of software functions and/or data for use with other programs
- Elements can be copied into a program when it is compiled or accessed when it is run
- Writing your program to use a GPL-licensed library will tend to necessitate the use of the GPL for your program

# GNU Lesser General Public License v2.1

*Significant Features*

- All modified versions of LGPL-licensed software must also be distributed under the LGPL (if they are distributed at all)
- All modified versions must advertise prominently what has been modified, who modified it, and when it was modified
- Source code must be provided with all LGPL-licensed software, either directly or via a request to the licensor

# GNU Lesser General Public License v2.1

*Significant Features*

- Modified works must be libraries
- Some works which might ordinarily be considered derivative are classed as special cases
- These ’works that use the library’ need not necessarily be distributed under the LGPL

# GNU Lesser General Public License v2.1

*What Does It Try To Do?*

- Allow free distribution and modification of the software
- Prevent modifiers distributing their modified versions in a non-free way
- Allow use of the software library by non-GPL-licensed software more freely than the GPL does

# Modified BSD License

*Significant Features*

- Short
- Unmodified versions of the software must retain the copyright statement, the licence conditions and the disclaimer of warranties
- Prior permission must be obtained from the licensor before their name can be attached to any modified version
- Compatible with the GPL

# Modified BSD License

*What Does It Try To Do?*

Explicitly:

- Preserve the accreditation of the original authors where the code is unmodified
- Protect the original authors from association with modified versions

Implicitly:

- Make the code it covers attractive to all potential modifiers, not just those committed to open source

# Apache License v2

*Significant Features*

- Explicitly allows linking to interfaces in the covered code without becoming a derivative work (compare GPL)
- Grants patent rights where necessary to operate and modify the code
- Provides a mechanism for modifiers to ’contribute’ their work into the ownership of the original author
- Revokes patent rights from anyone who engages in patent litigation against the author(s) (thus GPL incompatible)

# Apache License v2

*What Does It Try To Do?*

- Unify ownership to simplify litigation
- Explicitly grant patent rights where necessary

# Mozilla Public License v1.1

*Significant Features*

- Written by corporate lawyers - similar content in extremely different form
- Grants patent rights where necessary to operate and modify the code
- Mandates the inclusion of a list of necessary patent permissions, if these permissions are not granted by the licensor
- Modifications must be MPL-licensed, but additions may be licensed in any way

# Mozilla Public License v1.1

*What Does It Try To Do?*

- Steer a course between the BSD and GPL licences
- Preserve the openness of the code itself while allowing proprietary expansions that can be distributed along with the code

# Exploitation of open source software

We, an HE institution, own the IPR for some software. Do we

- bury it?
- try to sell licences?
- give the IPR to the developer?
- apply for a patent just in case?

or use an open source exploitation route?

# Burying

- our developers will be fed up
- we get no benefit from investment at all
- we may have to acquire an alternative

# Selling licences

This can be tricky. Do we have

- venture capital?
- sales and marketing skills?
- access to distribution channels?
- long-term development and support staff?

# Giving back IPR

Are you sure the developer won’t be selling it back to you next year?

# Patenting

- Do you have a tame patent lawyer?
- Do you see any immediate sales of rights?
- Can you afford to take action against people who abuse the patent?

# Open source exploitation opportunities

Going open source with your eyes open can avoid the problems above, and offer

- complementary revenue activities (research and consulting)
- training for your students
- achievement of non-financial goals (pedagogy and research)
- good publicity for your organisation

# and the downside?

- it might reduce the value of other assets
- one of the theoretical liability risks may come off

# Scenario one: the giveaway OSS release

You have created something, you do not want to hoard the work, but you do not have the resources to manage development:

- release the code under a liberal OSI licence
- offer the IPR to a neutral guardian
- limit the time your staff can spend on it
- do not be surprised if no-one picks it up. . .
- . . . but do not be jealous if an SME does
- Rewards: improved functionality, kudos
- Example: too numerous to mention

# Scenario two: the dual-licensing option

You are proud of your work, but you think it will survive better if it is widely used and integrated:

- release the code under a strict OSI licence
- create a legal entity to own IPR
- set up a *separate* activity to provide support and training
- licence the code under a commercial licence to companies who want to bundle it with their own
- accept the risk that someone else may start a better training company
- Rewards: improved functionality, kudos, money
- Example: LAMS, MySQL

# Related subjects

- Creative Commons
- MIT Open Courseware