How to choose a license for your software

Neil Chue Hong with contributions from Shoaib Sufi

1st February 2016, Glasgow

Neil Chue Hong (@npch), Software Sustainability Institute
ORCID: 0000-0002-8876-7606 | N.ChueHong@software.ac.uk

Supported by

Project funding from

Slides licensed under CC-BY where indicated:
Software

Helping the community to develop software that meets the needs of reliable, reproducible, and reusable research

Training

Delivering essential software skills to researchers via CDTs, institutions & doctoral schools

Outreach

Exploiting our platform to enable engagement, delivery & uptake

Collecting evidence on the community’s software use & sharing with stakeholders

Bringing together the right people to understand and address topical issues

Policy

Community
I Am Not A Lawyer

(this is not legal advice, only opinion)

For legal insight: http://ifosslawbook.org/
and for everything else: http://oss-watch.ac.uk/
What is IP?

- Intellectual Property are the legal rights (IPRs) from intellectual activity in the industrial, scientific, literary and artistic fields
- A Patent protects ‘new’ ideas and has an ‘inventive step’ that is not obvious to someone who works in the subject area
- Copyright is the protection of a tangible manifestation of an idea; e.g. a book or source/object code
- A License is an agreement or permission that grants a right to use - often in the form of a contract

More info: www.qlegal.qmul.ac.uk
Software Sustainability Institute
Why is IP important?

- Governs and protects exploitation rights
- Fallacies
  - ‘I found it on the internet so I can use it’
    - Publishing in a public place does not automatically grant rights to use
  - It’s fine to not have a license
    - No license worse than a restrictive license
  - You automatically own the IP to your creations
    - Often your employer may own the IP you generate as part of your job
- In the case of IP relating to research data and software, it is often something to be negotiated between you, your university, and your funders
What is a software license?

• A legal instrument governs the permission you are granting to others to use or distribute the software you hold the copyright on:
  ▪ Are you allowed to use it?
  ▪ Are you allowed to copy it?
  ▪ Are you allowed to resell it?
  ▪ Are you allowed to change it?
  ▪ Are you allowed to distribute it?
  ▪ Who is liable if something goes wrong?
Types of licenses

- Closed ("Proprietary")
- Restricted ("Academic" / "Non-commercial")
- Open source license
- Public domain / CC0
- Informal license
- No license

Considerations

- Larger works comprising code with different licenses ("license compatibility")
- Software made available under more than one license ("dual licensing")
- Copyright holders can re-license as they own the IP ("relicensing")
Why open software?

• Quicker start
• Better flexibility
• Improved robustness
• Increases collaborators
• Greater research impact
• Easier to work with industry
• No added cost
  ▪ Caveat: over what you should already be doing
Free as in freedom (Libre)

- Freedom for anyone to run the program as they wish, for any purpose *(non-discriminatory)*
- Freedom to study how the program works, and change it so it does your computing as you wish *(source code available, modifications allowed)*
- Freedom to redistribute copies so you can help others *(redistribution allowed)*
- Freedom to distribute copies of your modified versions to others to give them a chance to benefit from your changes *(derivatives allowed)*
Open Source Software is Free...

Photo “free beer tap” by jakob fenger (CC-BY)

Photo “Speech” by Quinn Dombrowski (CC-BY)

Software Sustainability Institute
Free as in Puppy...

- Long term costs
- Needs love and attention
- May lose charm after growing up
- Occasional clean-ups required
- Many left abandoned by their owners

Inspired by Scott McNealy
Photos of Great Pyrenees from Jen Schopf
Free as in Puppy...

- Long term costs
- Needs love and attention
- May lose charm after growing up
- Occasional clean-ups required
- Many left abandoned by their owners

Inspired by Scott McNealy
Photos of Great Pyrenees from Jen Schopf
Free as in kittens...

- Can become snarling, clawing, aloof...
- ... but they will provide joy if you treat them well

Photo “Kittens!” by James Wragg (CC-BY)

Software Sustainability Institute
Open source, open science

• Non-discriminatory
  ▪ Research should not be restricted or siloed

• Access to source code
  ▪ Research should be transparent, robust, and accessible

• Redistribution of software
  ▪ Providing access to the widest possible community

• Removing barriers to reuse
  ▪ Research should encourage building on the work of others, and giving them credit
Types of open source license

- Permissive (“Use with attribution”)
  - Simple (e.g. Modified “3-clause” BSD, MIT)
  - Grant Patent Rights (e.g. Apache, Eclipse)
- Copyleft (“Share modifications under same license”)
  - Strongly copyleft (e.g. GPL)
  - Weakly copyleft (e.g. LGPL, Mozilla, EUPL)
- All OS licenses allow private and commercial use; modification; distribution; limit liability; retain copyright
- More information
  - [http://choosealicense.com/licenses/](http://choosealicense.com/licenses/)
  - [https://tldrlegal.com/](https://tldrlegal.com/) (attorney verified interpretation)
Choosing an OSS license doesn’t need to be scary

Which of the following best describes your situation?

- **I want it simple and permissive.**
  - The **MIT License** is a permissive license that is short and to the point. It lets people do anything they want with your code as long as they provide attribution back to you and don’t hold you liable.
  - jQuery and Rails use the MIT License.

- **I’m concerned about patents.**
  - The **Apache License** is a permissive license similar to the MIT License, but also provides an express grant of patent rights from contributors to users.
  - Apache, SVN, and NuGet use the Apache License.

- **I care about sharing improvements.**
  - The **GPL (V2 or V3)** is a copyleft license that requires anyone who distributes your code or a derivative work to make the source available under the same terms. V3 is similar to V2, but further restricts use in hardware that forbids software alterations.
  - Linux, Git, and WordPress use the GPL.
License Compatibility

• Valid combining (e.g. libraries)

## License Compatibility

### Adapter's License Chart

<table>
<thead>
<tr>
<th>Status of Original Work</th>
<th>PD</th>
<th>BY</th>
<th>BY-NC</th>
<th>BY-NC-ND</th>
<th>BY-NC-SA</th>
<th>BY-ND</th>
<th>BY-SA</th>
<th>PD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PD</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>BY</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>BY-NC</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>BY-NC-ND</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>BY-NC-SA</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>BY-ND</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>BY-SA</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

For more details, visit [https://wiki.creativecommons.org/index.php/Frequently_Asked_Questions](https://wiki.creativecommons.org/index.php/Frequently_Asked_Questions)

Software Sustainability Institute
Choosing a License - considerations

• Do mindfully (know the implications)
  • 1 day of ‘pain’ might save weeks of ‘agony’ later
• Careful what you are depending upon
  • Check compatibility
• Impact
  • What sort of impact are you seeking?
  • Publish or perish a strong driver
• Contractual agreements
  • e.g EU collaborative agreements
 Licenses vs Governance

• Open source software philosophy is more than the license
  ▪ Some open licensed software is not produced openly
  ▪ Some closed licensed software benefits from open source project principles and processes
  ▪ Most open source contributors are paid (in some way) to do so

• Best practice identified from OSS projects useful for governance and project/product management of all types of software
  ▪ [http://producingoss.com](http://producingoss.com)
    • open forums, websites, transparent decision making, releasing software well, open about status

• Open source licensed software != Open development
Case study: CASTEP

• Originally developed at Cambridge by Mike Payne / Theory of Condensed Matter Group
  ▪ Cambridge Serial Total Energy Package
  ▪ Early 90’s, late 90’s rewrite, lots of partners
• A materials modelling code based on a first-principles quantum mechanical description of electrons and nuclei
• Interesting licensing
  ▪ Free to UK academics
  ▪ Discounted to non-UK academics
  ▪ Commercial offering via plugin to BIOVIA(formerly Accelrys)’s Materials Studio
Transitive – proprietary exploitation

- Dynamic Binary Translation
- Most famously known for supporting Apple’s transition from PowerPC Architecture to Intel back in 2005 – Apple® Rosetta® system (until OSX Lion in 2013)
- Spun out of University of Manchester with Manchester maintaining a share, the academic having a share and venture capital funding
- Eventually bought by IBM and then products stopped now staff working on unrelated things
- Closed source way of exploiting
- CTO Alasdair Rawsthorne back at the University of Manchester
Spark - Building community

• Spark started as academic project at Berkeley in 2009
  ▪ Code released under BSD in 2010
• Started Apache Incubator process in June 2013
  ▪ Gained contributions from 120 developers in 25 organisations, including Intel, Yahoo!, Cloudera, Alibaba
  ▪ Relicensed under Apache 2.0 license
• Became full Apache project in February 2014
  ▪ Most active Apache project in 2014
  ▪ Source code on GitHub
  ▪ Used globally by Alibaba, Amazon, IBM
• Company formed to commercialise in Sep 2013
  ▪ Based around hosting and consulting

DOI: 10.6084/m9.figshare.1434044
Git/GitHub/GitLab - Creating widely used platforms

- Git originally written and released by Linus Torvalds under GPL license
  - Git trademarks enforced for consistency
  - Easy for third parties to create services on top
- GitHub contribute to Git and open source some of their own tools
  - Business model exploits paid hosting / support for enterprise installations
- Gitlab no longer go further and release their commercial EE edition under MIT license (said that this did not work)
R – Open statistical computing

- Basics: Website, mailing list, code repository, issue resolution
- Remove barriers to participation, increase efficiency
- 1993: First public release; 2 devs
- 1995: Code open sourced; 3 devs
- 1996: r-testers list set up
- 1997: lists split: r-announce, r-help, r-devel; public CVS; 11 devs
- 2000: CRAN split and mirror
- 2001: BioConductor
- 2003: Namespaces
- 2005: l8n, L8n
- 2007: R-Forge
- Today: BioConductor (33 core devs), R-Forge (532 projects, 1562 devs), CRAN (1400+ packages)
Case studies in academia

• OSS Watch have produced case studies of several open source software projects in academia
  ▪ Koha: a case study in project ownership
  ▪ Wookie: a case study in sustainability
  ▪ ATutor LMS: a case study
  ▪ TexGen: a case study
  ▪ MediaWiki: a case study in sustainability
  ▪ WebPA: the road to sustainability
  ▪ Apache Cocoon: a case study in sustainability
  ▪ Moodle: a case study in sustainability
  ▪ Exim: a case study in sustainability

• [http://oss-watch.ac.uk/resources/casestudies](http://oss-watch.ac.uk/resources/casestudies)
Pros and Cons of Open Source

**Pros**
- Easier to evaluate
- No lock-in
- Easier to attract contributors and staff
- Frictionless code sharing
- Free advertising
- Able to take with you
- Generally better modularisation
- Reduced duplication
- Can work together on common platforms

**Cons**
- Harder to corner cut when shipping to deadline
- Harder to gain income from selling the software
  - But most sell services
- Cannot restrict to non-commercial use only
  - No neat definition of non-commercial.
- Once licensed, cannot revoke license
  - Though can relicense
- Can increase competition

DOI: 10.6084/m9.figshare.1434044
GPL vs Apache

- Software freedom vs developer freedom
- Copyleft licenses have increased license management costs
- Industry is generally more supportive of permissive licenses
- Open source software no longer written as an alternative to closed, but as the platform to drive service provision
- Software no longer the competitive differentiator, but the ability to operate it at scale

http://timreview.ca/article/650
What about making some £’s?

- Roy Azoulay – University of Oxford Innovation Centre
  - Supporting software startups
- Systematic 5 year review of exploitation
  - Patent protection of software IP is not profitable
  - Licensing your code is
- Other advice
  - If you want to earn by licensing your code later - make it available under copyleft license
    - GPL vs MIT/BSD/Apache
    - External contributions – ask for a contributor license agreement (CLA) http://oss-watch.ac.uk/resources/cla
Limitation of liability
(note opinion not legal advice)

• Open source licenses often drafted from a US legal perspective – some terms/condition don’t translate to other jurisdictions.

• E.g. In UK and EU Can’t contract out of ‘loss’
  • Unfair Contract Terms Act 1977
  • Watford Electronics Ltd vs Sanderson CFL Ltd
    • http://www.out-law.com/page-8689

• Clauses/terms that don’t apply likely to be ineffective in law

• Small risk

• If you thought defensively you would not do anything.
Summary

• Understand who owns the IP
• Identify if your software incorporates existing licensed code
• Decide what is the impact you are seeking
• Choose an appropriate license
• Make sure there is a governance model which supports contributors under your chosen license
Other resources

• Helping you choose a license:
  - http://choosealicense.com/licenses/
  - https://tldrlegal.com/
  - http://ufal.github.io/lindat-license-selector/

• Open source advice - OSSWatch – http://oss-watch.ac.uk

• All things Copyleft – GNU – http://www.gnu.org

• Legal defenders of Copyleft - https://www.softwarefreedom.org/

• Legal resources around IP, Licensing and other aspects -
  http://www.qlegal.qmul.ac.uk/resources/index.html

• Current debate around open source and sustainability -http://www.software.ac.uk/blog/2015-08-27-price-open-source-software-joint-response

• Open development and open source - http://producingoss.com/

• Country specific insights around open source - http://ifosslawbook.org/uk/

• University and Industry IP sharing templates - https://www.gov.uk/guidance/lambert-toolkit