Free-Range Code

Adam Sampson
University of Abertay Dundee

Overview

- I've been involved in several academic open source projects – some successful, some not
- Why might open source be of interest to you?
- How can you make it work in an academic context?
- I am not a lawyer; none of this is legal advice

Abertay University

Open source

Abertay University

Licenses

- A license describes what you're allowed to do with a piece of software you receive
- Legally, without a license, you're allowed to do almost nothing – the copyright owner must provide one to make the software useful
- Most software licenses are long and complex...
  - ... so most users – and developers – don't bother reading them

The Open Source Definition

- Anyone may redistribute the software, without paying a fee
- The software may be distributed in source code or compiled form
- Modifications and derived works are permitted under the same license as the original
- No restrictions on use, or discrimination against particular groups of people or fields of work

What does that mean?

- If you receive a piece of Open Source software, you can do all these things:
  - Use it
  - Make modifications and customisations
  - Give or sell copies of it to other people
  - Sell support services for it
  - Incorporate it into other projects
- ... provided you follow the terms of the license

Abertay University

Why open source?

- From the user's perspective:
  - no licensing fees
  - no vendor lock-in
  - you can freely share it with others
  - you can customise it for your purposes
- From the developer's perspective:
  - avoid duplicating work – good business sense
  - widely-used software is widely-tested:
    more eyes means fewer bugs

Abertay University

The scientist's view

- The validity of our results relies upon the correctness of our software
  - Data collection, simulation, analysis, visualisation
  - People aren't skeptical enough!
- The use of software is an increasingly important part of the experimental method
- So we should aim to publish the software we build *as part of* our method
  - … including spreadsheets, MATLAB/R scripts...

Abertay University

The scientist's view

- We should do our best to ensure our software works correctly
- Decent software engineering is hard work...
  - … so build on a solid foundation
- Make platforms and tools available for *reuse* in later work, rather than developing from scratch
  - Plenty of examples of this already
  - But development must be *sustainable*

Abertay University

Sustainability

- A healthy, ongoing, long-lived project
- Cooperating team of developers and occasional contributors
- Actively maintained:
  - problems are identified and fixed
  - new features added as required
  - the project adapts to changing circumstances
- Users and developers can easily obtain and make use of the software

Abertay University

Looking at licenses

Abertay University

The BSD license

- A *permissive* license
  - The simplest license that's widely used
  - Most licenses have more conditions
- Originally developed at Berkeley around 1990
  - Modified (and simplified) several times

BSD license (2-clause version)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

... warranty disclaimer ...

Abertay University

BSD license: disclaimer

THIS SOFTWARE IS PROVIDED BY <copyright holder> ''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 <copyright holder> 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.

Other options

- Other licenses offer different features
- *Copyleft* licenses say that if you distribute a modified version of the software, you must also distribute the corresponding source code
  - Encourages people to contribute work back
  - e.g. GNU General Public License
- *Patent grant* licenses protect users against software patent infringement
  - e.g. Apache license

Abertay University

Licenses for non-software

- These licenses assume you're talking about software (has source code, etc.)
- Similar licenses exist that can be used for documentation, graphics, scientific data...
- The *Creative Commons* licenses are the best-known example: several options for specifying how your work can be reused

How to do it

Abertay University

1 Consider the context

- How will you use the software in the future?
- Will others use it? (Is it worth releasing?)
- Are there existing projects that you can work with or build upon?
- Is there anything that'd prevent (or delay) you from releasing the software?
  - Pending publications
  - Patent problems

Abertay University

2 Get permission

- Identify the copyright owner
- You're probably *not* the copyright owner – especially if you're a student or working on a funded project
- If you aren't absolutely certain, ask your local intellectual property expert
  - Talk to other people who've done this
  - It's definitely better to ask permission than to seek forgiveness here

Abertay University

3 Use version control

- Version control software keeps track of the history of changes to your project, and makes it possible for developers to collaborate safely
- Good software engineering practice
- Especially important for open source: it enables external collaboration, and makes releasing software easier
- Many good, free VC systems: Subversion, Git...

Abertay University

4 Choose a license

- There are hundreds of existing licenses
- Pick one that is *widely-used*
  - BSD, Apache, GPL, LGPL...
- Make sure you understand the license, and that it meets your project's needs
- Do not *under any circumstances* attempt to write your own license
  - You will get it wrong
  - Many, many examples of this

Abertay University

5 Find a home

- Somewhere to host the project web site, version control history, files for users to download...
- Regular, reliable, automated backups
- Public, private, or some mix of the two?

Abertay University

Hosting options

- WIthin the university
  - Will it still be there after you leave?
  - Can external contributors work with it?
- A commercial service (SourceForge, GitHub)
  - Are the licensing and privacy conditions OK?
  - If so, usually free; can pay when they aren't
- Host it yourself
  - Is the service reliable?
  - Who will pay for it and look after it?

Abertay University

6 Keep it tidy

- Make it easy for others to use and reuse
- Write sufficient documentation
  - Both for end-users and other developers
  - Examples and test cases are really useful
- Follow convention as far as possible
  - e.g. how you package your software
  - Look at what similar, successful projects do
- Clear copyright statements and license terms
- Make releases!

Abertay University

7 Provide infrastructure

- Project web site
  - Stable location
  - Description, downloads, instructions, contacts
  - Name and logo
- Support discussion: mailing list, forum...
- Provide help: FAQ, wiki...
- Keep track of problems: issue tracking
- A hosting service will give you most of this without much effort

Abertay University

8 Advertise

- Mention the project in papers, posters, presentations, web pages...
- Are there other researchers who would be interested in using (and contributing to) it?
- How about members of the public?
  - Unconferences, user groups, science cafes...
  - List on open source software sites – can get surprising reuse
- Impact

Abertay University

9 Support the community

- Be responsive
- Help with problems (feed into documentation)
- Discuss plans (news blog, etc.)
- Encourage collaboration
  - Make contributors feel welcome
  - Provide helpful feedback on contributions
  - (but learn to say “no” when appropriate)

Abertay University

10 Let go

- Be willing to give up some control for the good of the project
- Encourage branching
  - … where someone else maintains a modified version of your code for a different purpose
- If you don't have time to maintain it yourself, let someone else take over

Conclusions

Abertay University

Conclusions

- Advantages of open source for science:
  - Avoid duplicated work
  - Improve software quality
  - Make collaboration easier
  - Assist with public engagement
  - Build confidence in validity of results
- General advice: follow what the successful projects do!

Abertay University

Resources

- **OSS Watch**
  UK academic open source advisory service
  http://www.oss-watch.ac.uk/
- **Software Carpentry**
  Software engineering for scientists
  http://software-carpentry.org/
- Thank you for listening – any questions?

Abertay University

License details

- This presentation is:

- The image on the title page is:

Abertay University