

COMP7047 – Open Source Projects

01 - Introduction

Your Lecturer

- Larkin Cunningham
- Larkin.Cunningham@cit.ie

- PhD in Digital Humanities (submitted, awaiting viva), UCC
- MSc Creative Writing, Edinburgh University
- BSc (hons) Computer Services Management, CIT

Syllabus (1)

- Brand new module, slightly experimental (in other words, we'll see how things go and be agile, responding to how you get on)
- Module is a hybrid of traditional lectures/labs and a research project module – therefore delivery of module will be a mix of lectures/labs and supervision
- But, first few weeks will cover:
 - History of Open Source
 - Free Software vs Open Source
 - Licencing
 - Structures of Open Source communities
 - Corporations and Open Source
 - How to contribute to an Open Source project
 - And more...

Syllabus (2)

- Later lecture/lab time (perhaps from week 5 or 6) will be given over to supervision and project support

Assessment (1)

- Note from module descriptor (<https://courses.cit.ie/index.cfm/page/module/moduleId/13171>):

Short Answer Questions	The students would be assessed on the theoretical elements presented in the module.	1,2	20%	Week 3!
------------------------	---	-----	-----	----------------

- Assessing these learning outcomes:

LO1 Discuss the underlying principles and the benefits of contributing to open source projects.

LO2 Understand the typical structure of an open source project by identifying: the community owner, maintainers, contributors and members; the documentation; and licencing.

Assessment (2)

Project	In this report the students would be expected to evaluate various open source projects and will select an open source project to contribute to, bearing in mind their own technical skill set and interests. In addition, they will have identified how to connect and communicate with their selected open source project.	3,4	20%	Week 5
---------	---	-----	------------	---------------

- **Assessing:**

LO3 Evaluate various open source projects and select an open source project to contribute to.

LO4 Identify how to contribute and connect to the open source project and workflow procedures to enable contribution.

Assessment (3)

Reflective Journal	The student will reflect on their experience of learning about open source communities and their journey as novice contributor.	1,2,3,4,5,6,7	20%	Every Week
--------------------	---	---------------	-----	------------

- Each week, starting with week 2, write a reflective journal entry of 200 words approx.
- A rubric and some guides will be posted to Canvas offering guidance on how to write good reflective journal entries

Assessment (4)

Project	The student will employ tools, identify the contribution workflow and using this workflow will contribute to an open source community by fixing bugs, development of source or document or scripts etc.	1,2,3,4,5,6,7	20%	Sem End
Presentation	The students will demo their project and will present their contribution to the peers and lecturer.	1,2,3,4,5,6,7	20%	Sem End

Free Software Foundation

- 1985 – Free Software Foundation (FSF) established by **Richard Stallman**

“The Free Software Foundation (FSF) is a nonprofit with a worldwide mission to promote computer user freedom. We defend the rights of all software users.” (<https://www.fsf.org/>)

- Required viewing - https://www.youtube.com/watch?v=Gnw_K5DPkbc
(Richard Stallman Interview on the History and Ethics of Free Software)
 - After viewing, ask yourself: Is he a fundamentalist?



Free Software Foundation

- Also see <https://www.fsf.org/blogs/rms/20140407-geneva-tedx-talk-free-software-free-society> and the embedded TED talk video
- Note the following text on the page:
 - *“We've embedded this talk using the HTML5 video element, but if you watch any TEDx talks on YouTube, please remember to use a free program like [ytdl](#) instead of your browser, because watching YouTube in your browser requires [proprietary JavaScript](#).”*
 - An example of fundamentalism?
- Now we will watch and discuss the video in class...

Free Software Definition

- “Free software” means software that respects users' freedom and community. Roughly, it means that **the users have the freedom to run, copy, distribute, study, change and improve the software**. Thus, “free software” is a matter of liberty, not price. To understand the concept, you should think of “free” as in “free speech,” not as in “free beer”. We sometimes call it “libre software,” borrowing the French or Spanish word for “free” as in freedom, to show we do not mean the software is gratis.

<https://www.gnu.org/philosophy/free-sw.html>

The 4 Essential Freedoms

- The freedom to run the program as you wish, 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 others (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.

Free Software is the same as Open Source, Right?

- ““Open source” is something different: it has a very different philosophy based on different values. Its practical definition is different too, but nearly all open source programs are in fact free. We explain the difference in [Why “Open Source” misses the point of Free Software.](https://www.gnu.org/philosophy/free-sw.html)” (<https://www.gnu.org/philosophy/free-sw.html>)
- Stallman: “When open source proponents talk about anything deeper than that, it is usually the idea of making a “gift” of source code to humanity. Presenting this as a special good deed, beyond what is morally required, presumes that distributing proprietary software without source code is morally legitimate.”
:
• ““Free software” and “open source” are different ideas but, in most people's way of looking at software, they compete for the same conceptual slot.”
(<https://www.gnu.org/philosophy/open-source-misses-the-point.html>)
- FOSS – Free and Open Source Software
- FLOSS – Free/Libre and Open Source Software
- What UNESCO has to say about FOSS: <https://en.unesco.org/foss>

Examples of FOSS



- Describes itself as “free and open source”, developed by users who “believe in the principles of Free Software”
(<https://www.libreoffice.org/about-us/who-are-we/>)



- See https://en.wikipedia.org/wiki/List_of_Apache_Software_Foundation_projects for a list of projects of the Apache Software Foundation (most famously, the Apache httpd web server, released in 1995/96)

A Case Study of Open Source Software Development: The Apache Server (2000)

- Mockus, A., Fielding, R.T., Herbsleb, J., 2000. A case study of open source software development: the Apache server, in: Proceedings of the 22nd International Conference on Software Engineering - ICSE '00. Presented at the the 22nd international conference, ACM Press, Limerick, Ireland, pp. 263–272.

<https://doi.org/10.1145/337180.337209>

A Case Study of Open Source Software Development: The Apache Server (2000)


ABSTRACT

According to its proponents, open source style software development has the capacity to compete successfully, and perhaps in many cases displace, traditional commercial development methods. In order to begin investigating such claims, we examine the development process of a major open source application, the Apache web server. By using email archives of source code change history and problem reports we quantify aspects of developer participation, core team size, code ownership, productivity, defect density, and problem resolution interval for this OSS project. This analysis reveals a unique process, which performs well on important measures. We conclude that hybrid forms of development that borrow the most effective techniques from both the OSS and commercial worlds may lead to high performance software processes.

Example of a Small Open Source Project

- Let's go back to the open source project (ytdl) referred to on the previous slide:
 - <https://github.com/fent/node-ytdl#readme>
- Clicking on LICENSE reveals the following:

Branch: master ▾ node-ytdl / LICENSE Find file Copy path

 fent/node-ytdl is licensed under the **MIT License**

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions	Limitations	Conditions
<ul style="list-style-type: none">✓ Commercial use✓ Modification✓ Distribution✓ Private use	<ul style="list-style-type: none">✗ Liability✗ Warranty	<ul style="list-style-type: none">ⓘ License and copyright notice

This is not legal advice. [Learn more about repository licenses.](#)

- See text of the licence on the next page... (note how short it is compared to other licence agreements you may have agreed to in the past)

MIT Licence

MIT License

We will examine the common licences used by Open Source Projects later

Copyright (C) 2012 by fent

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Exercise

- What more can we tell about the project from GitHub?
- Go to <https://github.com/fent/node-ytdl#readme>
- Look at all the various links... what do they tell us about the project?
(Let's do this informally for now in any way you want you – we will get more formal over the next few weeks)
 - Spend 10 to 15 minutes examining the repository and taking notes
- Note: if you don't have a GitHub account, create one and log in

GNU

- GNU (Gnu's Not Unix)
- A Unix-like operating system that is free/libre (see <https://www.gnu.org/#More-GNU>).
 - 1984 – Development of GNU starts
 - 1989 – FSF publishes Version 1 of GNU General Public License (GPL)
 - 1991 – GPL Version 2 (we'll return to these licences later)
 - 1991 – Linus Torvalds launches development of the Linux kernel under the GPL v.2 licence
 - GNU + Linux Kernel = GNU/Linux
 - Worth reading this article about Thorvalds and further clicking into the *Linux Code of Conduct* (<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=8a104f8b5867c682d994ffa7a74093c54469c11f>) – you can also see the source code of the Linux kernel at that site