

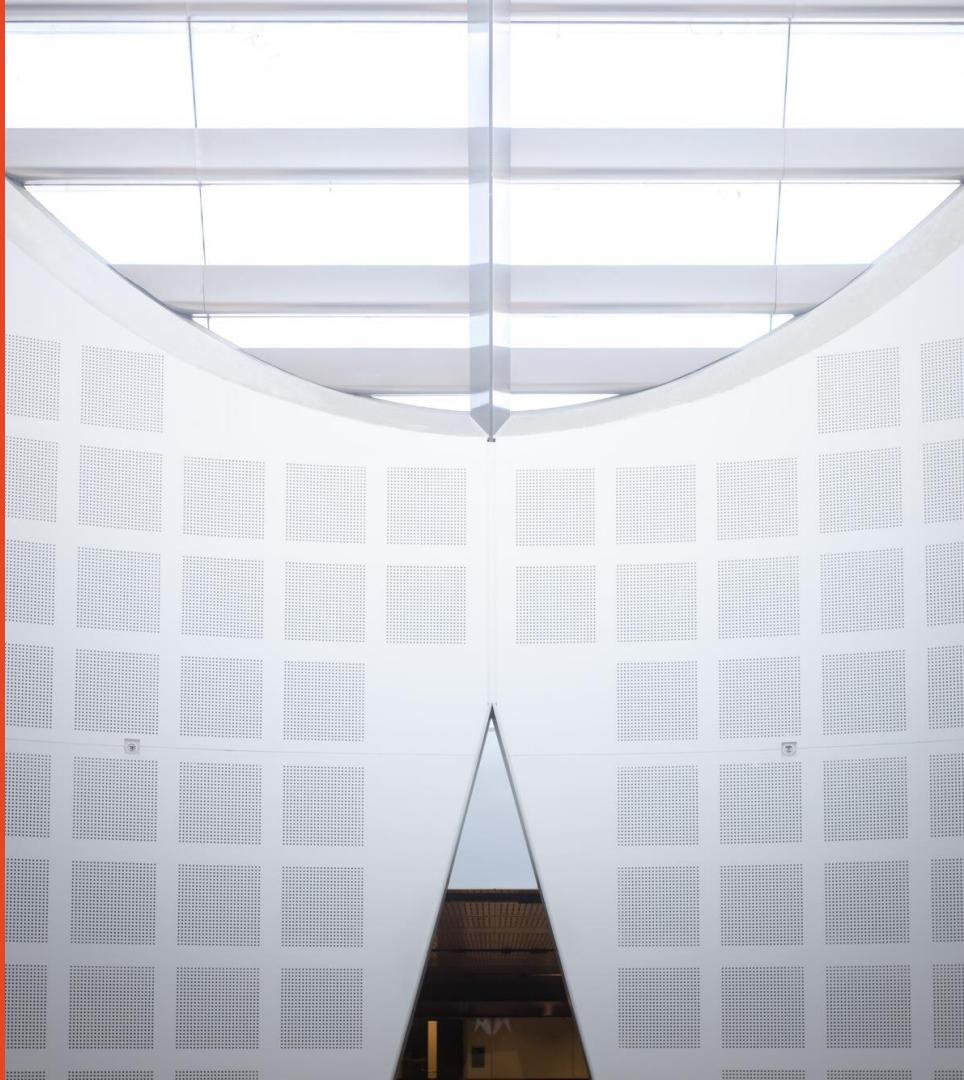
Agile Software Development Practices (SOFT2412/COMP9412)

Ethics, Intellectual Property, and Open-Source Software

Dr. Basem Suleiman

School of Computer Science

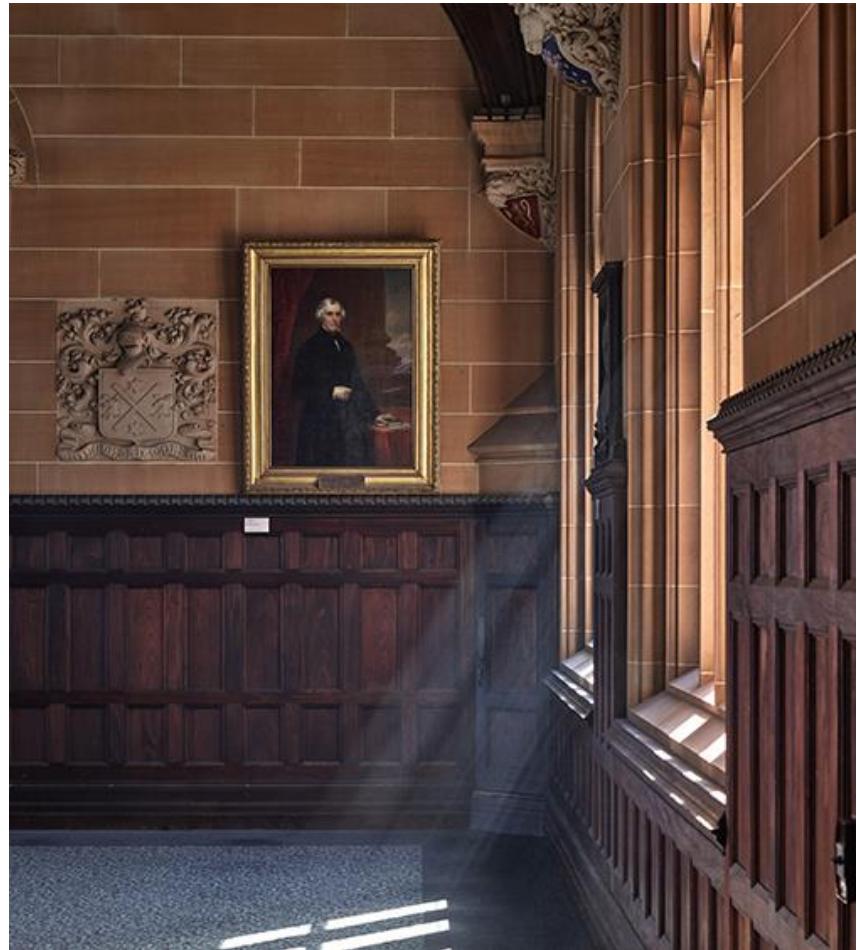
Based on material prepared by Alan Fekete



Agenda

- Ethics and Professional Frameworks
 - Ethical Responsibility, Privacy, Liability
- Intellectual Property
 - Copyright, Trademark, trade secretes
 - Patents
- Open-source Software
- Open-source Software Licensing

Ethics and Professional Frameworks



Ethics – Theory

- Ethics is a branch of philosophy!
 - What is right? What is wrong?
- Ethics vs Morals?
 - Not simple, but...
 - Morals: Principles of right and wrong that guide personal behaviour – your personal compass.
 - Ethics: Rules of conduct accepted within a social context. External.
- How do undertake ethical reasoning? Frameworks for making judgements...

http://www.diffen.com/difference/Ethics_vs_Morals

Professional Frameworks

- Most professional associations have “frameworks” guiding professional conduct
- These typically include consideration of ethical conduct
- Examples:
 - [ACS: Code of Professional Conduct](#)
 - [ACM Code of Ethics and Professional Conduct](#)
 - [IEEE CS Code of Ethics](#)

Professional Frameworks – ACS

- **The Primacy of the Public Interest**
 - You will place the interests of the public above those of personal, business or sectional interests.
- **The Enhancement of Quality of Life**
 - You will strive to enhance the quality of life of those affected by your work.
- **Honesty**
 - You will be honest in your representation of skills, knowledge, services and products.
- **Competence**
 - You will work competently and diligently for your stakeholders.
- **Professional Development**
 - You will enhance your own professional development, and that of your staff.
- **Professionalism**
 - You will enhance the integrity of the ACS and the respect of its members for each other.

https://www.acs.org.au/content/dam/acs/acs-documents/ACS%20Code-of-Professional-Conduct_v2.1.pdf

Professional Frameworks – IEEE CS

- **Public:** Software engineers shall act consistently with the public interest.
- **Client and employer:** Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest.
- **Product:** Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
- **Judgment:** Software engineers shall maintain integrity and independence in their professional judgment.
- **Management:** Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
- **Profession:** Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
- **Colleagues:** Software engineers shall be fair to and supportive of their colleagues.
- **Self:** Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Professional Frameworks – ACM

- General Ethical Principles
 - Society and human well-being, avoid harm, be honest and trustworthy, be fair, respect work, respect privacy, honour confidentiality
- Professional Responsibilities
 - More specific aspects including quality of work and processes, quality of reviews and judgement, competence
- Professional Leadership Principles
 - Those have leadership roles including quality of work, training and competence, public interest decisions to use or retire systems

<https://www.acm.org/code-of-ethics>

Scenario – Discuss

- You work for a small company, SmallCorp Consultants. The company's only contract at present is a \$6M year-long project for Bluestone Mining to write a software system to analyse geological data.
- You discover a way of designing the system that means that the project can be completed in 2 months, rather than 1 year, and at a cost of only \$1M.
- Your boss tells you to keep quiet and ignore the “better solution” as they want to keep the team working on the project as long as possible (otherwise he would have to sack most of the staff), and Bluestone has already indicated they are happy to pay the \$6M as they believe the project is worth that much.
- What do you do?

Ethical Responsibility

- Who's responsible?
- A system is developed then some errors/faults occur during operations
 - Therac-25, Ariane 5, Nissan Airbag software defect
- Independent QA team did integration testing – who is ethically responsible for any damage may occur by a fault?
- Factors include software methodology, environment, information provided

Who's Ethically Responsible?

- Responsibility for engineering and geoscience software:

“Developing software is a highly risky proposition. The software development process is a complex undertaking consisting of specifying, designing, implementing, and testing. Any small mistake or fault will cause unlimited damage to society. Professional Members contribute to the success of software development projects. However, the Association of Professional Engineering and Geoscience is primarily concerned with their responsibility for minimizing the risk of failure and protecting the public interest.”

https://en.wikipedia.org/wiki/Software_engineering_professionalism

ACM Professional Responsibilities

- Strive to achieve high quality in both the processes and products of professional work
- Maintain high standards of professional competence, conduct, and ethical practice
- Know and respect existing rules pertaining to professional work.
- Accept and provide appropriate professional review.
- Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks ↪
- Perform work only in areas of competence.
- Foster public awareness and understanding of computing, related technologies, and their consequences
- Access computing and communication resources only when authorized or when compelled by the public good.
- Design and implement systems that are robustly and usably secure.

Privacy

- ACM Professional Responsibility
 - Access computing and communication resources only when authorized or when compelled by the public good
 - Individuals and organizations have the right to restrict access to their systems and data so long as the restrictions are consistent with other principles in the Code
 - Computing professionals should not access another's computer system, software, or data without a reasonable belief that such an action would be authorized or a compelling belief that it is consistent with the public good

Privacy – Case Study

- Toysmart vs FTC
 - Online Toy store in Waltham, Mass.
 - Privacy policy on website: would not share details...
 - Filed for bankruptcy – and then sought permission to sell assets.
 - *Including customer information: names, addresses, billing info, browsing and purchasing history....*
 - Original FTC proposal was to allow this – but only selling to a buyer in a related market, who adhered to the original privacy policy
 - Disney bought the assets and destroyed the consumer information!
 - Q: Is this a problem? Why?
 - See:
 - http://itlaw.wikia.com/wiki/FTC_v._Toysmart.com
 - <http://www.computerworld.com/article/2596456/e-commerce/opinion--toysmart-case-can-set-bar-for-online-privacy.html>

Privacy – Case Study

- Google WarDriving
 - Google cars capture Street View information!
 - *But also captured WiFi data worldwide!*
 - SSID, MAC address, signal strength, but also any unencrypted data packets.
 - Google blamed a “rogue engineer”
 - *But later shown that Google managers had commissioned the wardriving program, to help them build Wi-Fi maps.*
 - Q: Aren’t they just capturing data that is freely available anyway? And surely it helps them provide a better service?
 - See:
 - [http://www.darkreading.com/risk-management/google-wardriving-how-engineering-trumped-privacy/d/d-id/1104126?](http://www.darkreading.com/risk-management/google-wardriving-how-engineering-trumped-privacy/d/d-id/1104126)

Privacy – Case Study

- ACM Professional Responsibilities
- *A system being publicly accessible is not sufficient grounds on its own to imply authorization. Under exceptional circumstances a computing professional may use unauthorized access to disrupt or inhibit the functioning of malicious systems; extraordinary precautions must be taken in these instances to avoid harm to others.”*

Scenario – Unauthorised access?

- Next semester you get a part-time job working for the University’s ICT unit doing testing on development changes to Sydney Student. As part of this you have access (for testing purposes) to a copy of all course and student data.
- Is it OK for you to see unit of study results for your friends as part of your testing?
- Is it OK for you to see the results of other students who you don’t know but who are in your course?
- Is it OK for you to see the results of students in other courses?
- Is it OK for you to see real results / personal information, if the names have been removed?

Product Liability

- “the area of law in which manufacturers, distributors, suppliers, retailers, and others who make products available to the public are held responsible for the injuries those products cause”
- Is computer software a product or a service?
 - As product (code) sold with license
 - Software as a Service; buy subscription
 - Australian Consumer Law (ACL) lists computer software as a good

<https://www.slideshare.net/aliasnetwork/software-liability>

https://en.wikipedia.org/wiki/Product_liability

Software Liability – Software Engineers

- Should software vendors be liable for software failures/defects?
- Are software developers liable for defects in their software (in Australia)?
- Software is a product, maybe strict liability or negligence
- Software is a service, generally not liable
- Software engineers should have ethical responsibilities, be competent in doing their job, communicate any issues they may observe with their managers

See:

- [Who is liable for software errors? Proposed new product liability law in Australia](#)
- [Are software developers liable for defects in their software?](#)

Case study – Liability for Unreliability

- Therac-25
 - Medical radiation therapy machine.
 - Mid-1980's : At least 6 accidents of massive overdose of radiation, and at least 3 deaths.
 - Subsequent commission found:
 - Primary reason: bad software design and development practices
 - Code was not independently reviewed
 - No analysis of possible failure modes
 - Poor documentation of error codes, and ability to override
 - Q: Should the programmers have been held criminally liable? Why?
 - Q: How do you avoid hubris?



Liability – Nissan Recall – Airbag Defect*

- What happened?
 - ~ 3.53 million vehicles recall of various models 2013-2017
 - Front passenger airbag may not deploy in an accident
- Why?
 - Software that activates airbags deployment improperly classify occupied passenger seat as empty in case of accident
 - No warning that the airbag may not function properly
 - Software sensitivity calibration due to combination of factors (high engine vibration and changing seat status)
- Q: Who is liable? Why?
- Q: How to avoid such scenarios?



<http://www.reuters.com/article/us-autos-nissan-recall/nissan-to-recall-3-53-million-vehicles-air-bags-may-not-deploy-idUSKCN0XQ2A8>
<https://www.nytimes.com/2014/03/27/automobiles/nissan-recalls-990000-cars-and-trucks-for-air-bag-malfunction.html>

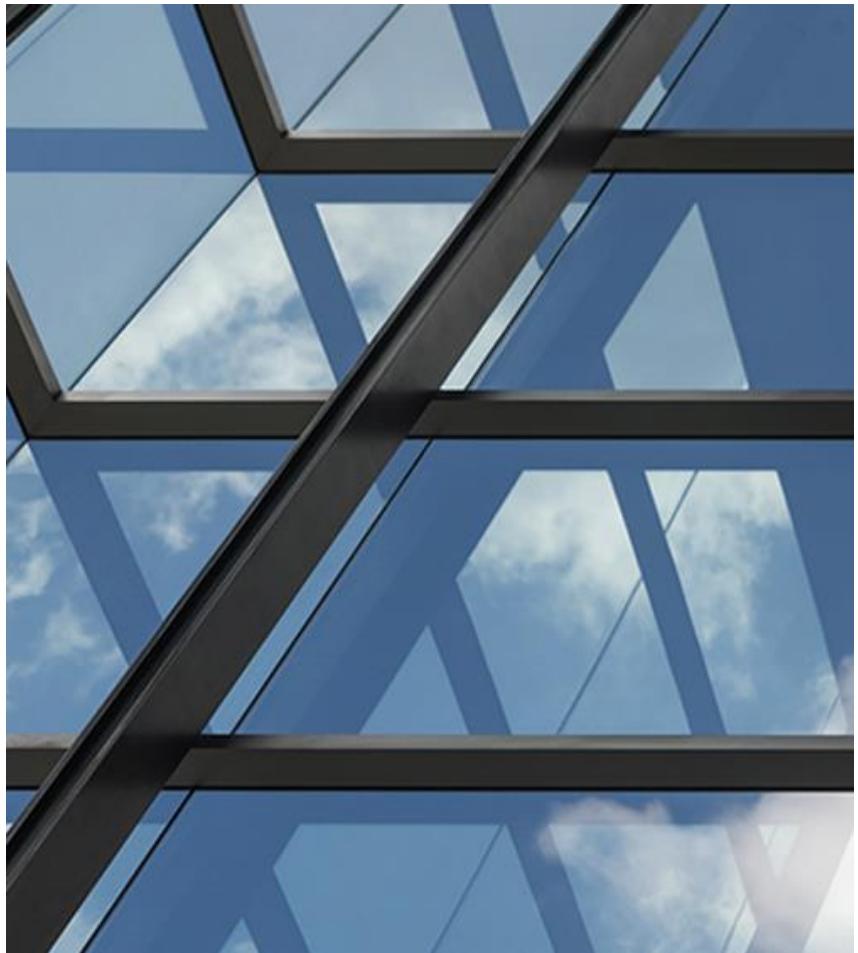
Practicing Within Your Abilities?

- Who here is a good programmer?
 - How do you know?
- Professionalism – what does this mean?
- ACM Code of Ethics:
 - 2.2 Acquire and maintain professional competence.
 - 2.3 Know and respect existing laws pertaining to professional work.
 - 2.4 Accept and provide appropriate professional review.
- Evaluate your programming proficiency – How good are you at programming?
 - Also read:
 - Hitting the High Notes

Intellectual Property



THE UNIVERSITY OF
SYDNEY



Intellectual Property (IP)

- “A category of property that includes intangible creations of the human intellect”
- IP is intangible
 - Challenging to protect – how to protect a software or design?
- IP law to protect people and business’ intellect
 - Economic incentives that lead to innovation and technological development

Intellectual Property (IP) – Types

- Copyright
 - Grant a creator of an original work exclusive right it
 - Source code, executable code, database, artistic work
 - Granted automatically in Australia, not all countries though
 - Does not protect ideas or methods employed
- Trademarks
 - Sign, design or expression which distinguishes products or services
 - Formal registration and require meeting certain criteria
- Trades secretes
 - Process, formula, practice, design, pattern which is unknown and provide a competitive advantage

Intellectual Property (IP) – Patents

- Form of IP protection which grants inventor(s) the right to exclude others from making, using or selling and importing an invention for a limited time, in exchange for the public disclosure of the invention
- Inventor(s):
 - Exclusive rights to the inventor (usually 20 years)
 - Freedom to exploit the invention (commercially and non-commercially)
- Public:
 - Enrich body of knowledge and innovation – valuable information to be shared with the public

Intellectual Property (IP) – Patents

- To be patentable, an invention must be:
 - New
 - Non-obvious
 - Useful (industrial applicability)
 - The disclosure of the invention in the patent application must meet certain formal and substantive standards
- A patent is granted for an invention, which may be described, in general, as a solution to a technical problem
- Local protection (IP Australia) vs. international protection (WIPO/PCT)

Ref: http://www.wipo.int/sme/en/documents/software_patents_fulltext.html

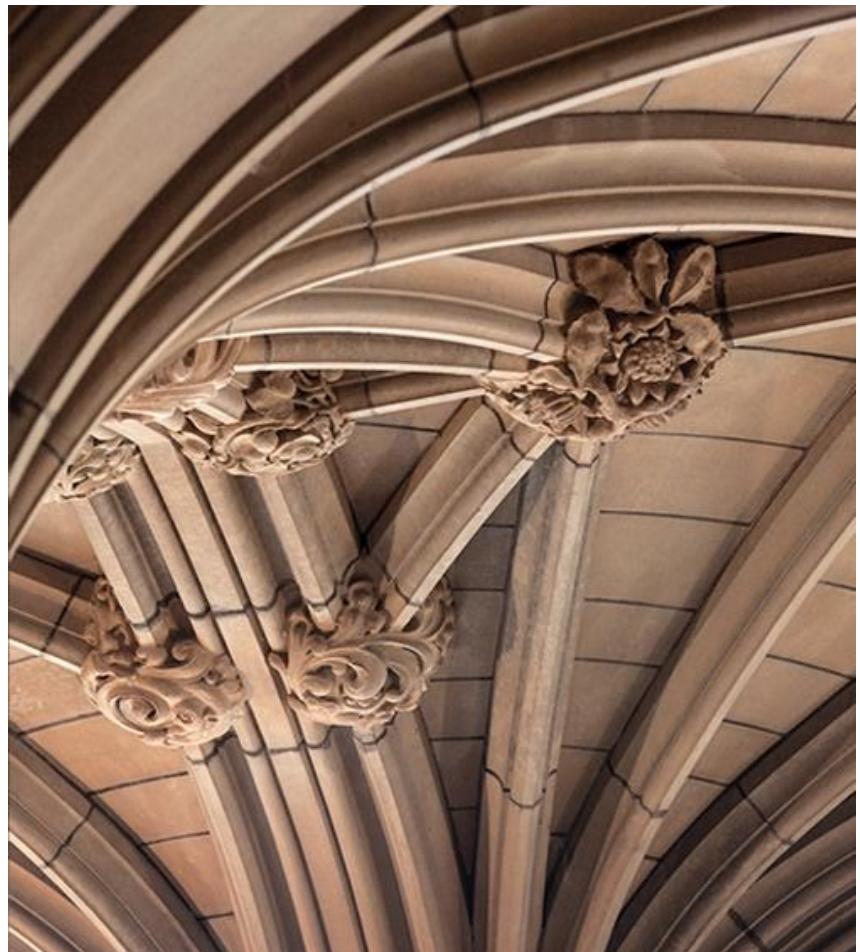
Intellectual Property – Scenario

- You accepted a 6-month short term contract job, working for a small start-up that is creating a new App that locates the cheapest place near your current location to buy consumer items. Whilst working on the project you accidentally discover that they have obtained access to proprietary code from another company (that analyses web pages to find product price information). They aren't using the code directly, but they are analysing it so they can understand the algorithm and then duplicate it in their system.
- Is this OK? Why? What would you do about this?

Licensing



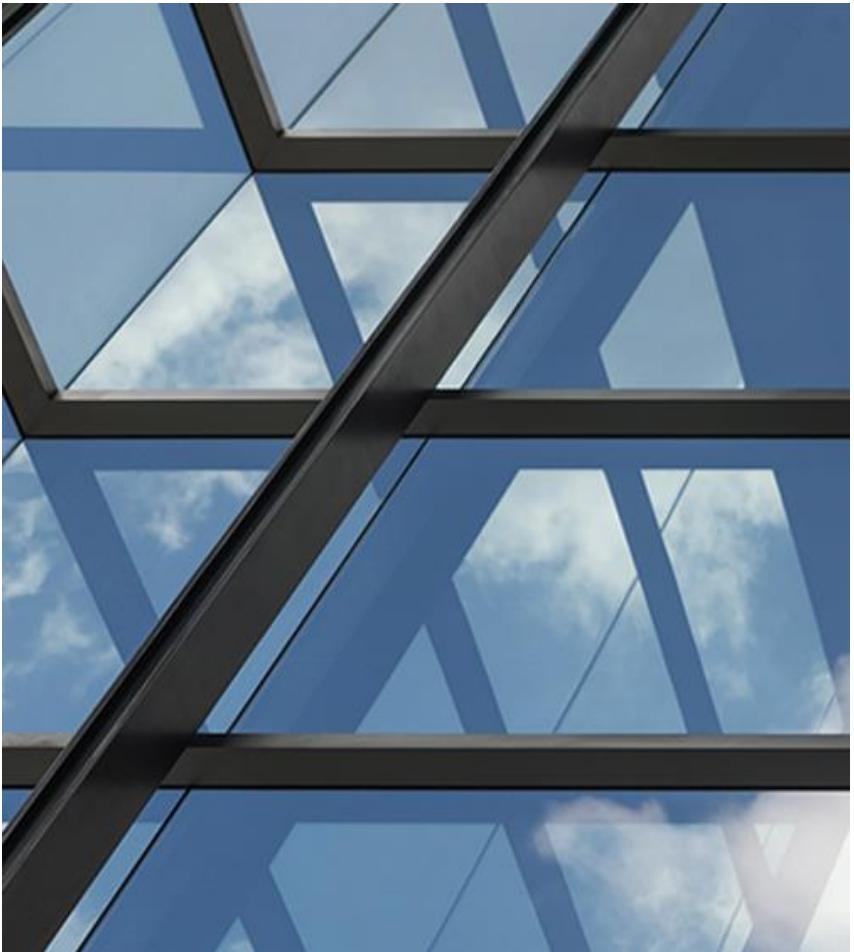
THE UNIVERSITY OF
SYDNEY



Licensing

- “Official permission to do or use something”
- Licensor grant a license to authorize a use to licensee
 - Copyrighted software or patented invention
 - Nature of use (commercial or other)
 - Condition of use
 - Duration of use
 - Return to licensor

Open-source Software



What is Open-source Software?

- **Open-source:** “something people can modify and share because its design is publicly accessible.”
- **Open-source software:** “software with source code that anyone can inspect, modify, and enhance”
- “Closed source” or “proprietary” software
 - Who can legally copy, inspect and alter proprietary software?
 - E.g., MS Office. Matlab

Open-source Software vs Free Software

- Does open-source software mean free software? Why/Why not?
- What's the difference between open-source and free software?

Open-source Software vs Free Software

- Free software refers to liberty of use, not price
 - Run, copy, study, change, distribute and improve source code
- The four essentials of freedom:
 - Run the program as you wish for any purpose
 - Study how the program works, and change it as you wish
 - Redistribute copies of the original program
 - Redistribute copies of your modified versions

More details: [Why open-source misses the point of free software](#)

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

The Open-source Initiative

- maintains the definition and trademark
- approves open-source licenses



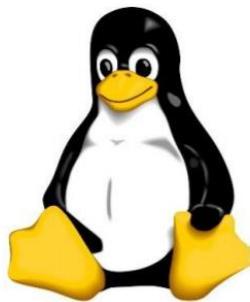
<https://osr.cs.fau.de/wp-content/uploads/2009/10/AuOS-Lecture-08-v01.pdf>

Open-source Software – License

- Terms of license but legally different from proprietary (closed-source) software
 - Copyleft licenses; must also release modified source code
 - How to use, study, modify and distribute software
 - Generally, permission to use for various purposes
 - Promote collaboration and sharing
 - Free of charge?
 - Distribute modified source code?
- Closed-source (proprietary) software
 - License agreement – legally different

<https://opensource.com/resources/what-open-source>

Open-source Software – Examples



PostgreSQL



Open-Source Software – Various Examples

- Linux
- FreeBSD, OpenBSD, and NetBSD
- Apache, which runs over 50% of the world's web servers
- BIND, the software that provides the DNS (domain name service) for the entire Internet.
- sendmail, the most important and widely used email transport software on the Internet
- Mozilla, the open-source redesign of the Netscape Browser
- OpenSSL is the standard for secure communication (strong encryption) over the Internet

<https://www.gnu.org/philosophy/open-source-misses-the-point.en.html>

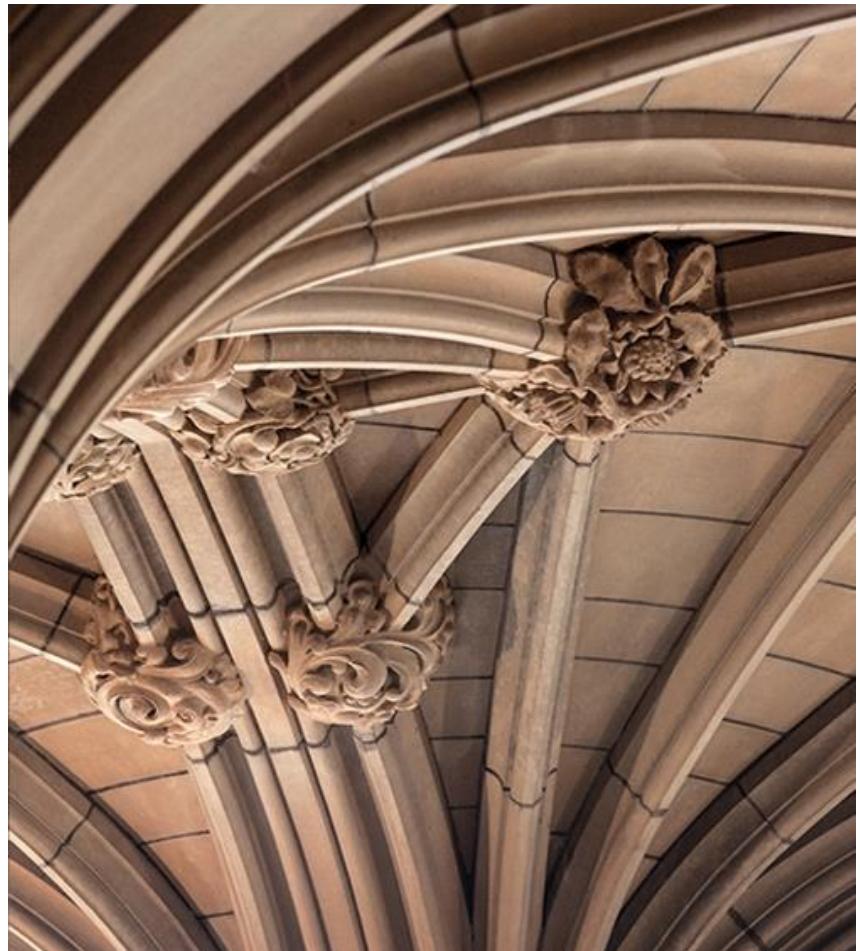
Open-source Software – Web and Tools

- Zope, and PHP, are popular engines behind the "live content" on the World Wide Web
- Languages:
 - Perl
 - Python
 - Ruby
 - Tcl/Tk
- GNU compilers and tools
 - GCC
 - Make
 - Autoconf
 - Automake
 - etc.

Ref: <https://www.gnu.org/philosophy/open-source-misses-the-point.en.html>

Licensing

Open-source Software



Open-source Software – License Types

- Open-source licenses approved by OSI (Open-source Initiative)
- Common licenses
 - [Apache License 2.0](#)
 - [BSD 3-Clause "New" or "Revised" license](#)
 - [BSD 2-Clause "Simplified" or "FreeBSD" license](#)
 - [GNU General Public License \(GPL\)](#)
 - [GNU Library or "Lesser" General Public License \(LGPL\)](#)
 - [MIT license](#)
 - [Mozilla Public License 2.0](#)
 - [Common Development and Distribution License](#)
 - [Eclipse Public License](#)



<https://opensource.org/licenses/category>

MIT License

- A short and simple permissive license
- Require preservation of copyright and license notices
- License works, modifications, and larger works may be distributed under different terms and without source code.

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

- Example: Ruby

<https://choosealicense.com/licenses/mit/>

MIT License – Use

MIT License

Copyright (c) [year] [fullname]

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.

<https://choosealicense.com/licenses/mit/>

Apache License 2.0

- Require preservation of copyright and license notices.
- Contributors provide an express grant of patent rights.
- Licensed works, modifications, and larger works may be distributed under different terms and without source code

Permissions	Conditions	Limitations
<ul style="list-style-type: none">• Commercial use• Distribution• Modification• <u>Patent use</u>• Private use	<ul style="list-style-type: none">• License and copyright notice• <u>State changes</u>	<ul style="list-style-type: none">• Liability• <u>Trademark use</u>• Warranty

- Example: PDF.js

<https://choosealicense.com/licenses/apache-2.0/>

Mozilla Public License 2.0

- Conditioned on making available source code of licensed files and modifications of those files under the same license
- Copyright and license notices must be preserved.
- Contributors provide an express grant of patent rights
- a larger work using the licensed work may be distributed under different terms and without source code for files added in the larger work.

Permissions	Conditions	Limitations
<ul style="list-style-type: none">• Commercial use• Distribution• Modification• Patent use• Private use	<ul style="list-style-type: none">• Disclose source• License and copyright notice• <u>Same license (file)</u>	<ul style="list-style-type: none">• Liability• Trademark use• Warranty

<https://choosealicense.com/licenses/lgpl-3.0/>

GNU Affero General Public License(AGPL) v3.0

- Conditioned on making available complete source code of licensed works and modifications
- Copyright and license notices must be preserved.
- Contributors provide an express grant of patent rights.

Permissions	Conditions	Limitations
<ul style="list-style-type: none">• Commercial use• Distribution• Modification• Patent use• Private use	<ul style="list-style-type: none">• <u>Disclose source</u>• License and copyright notice• <u>Network use is distribution</u>• Same license• State changes	<ul style="list-style-type: none">• Liability• Warranty

<https://choosealicense.com/licenses/agpl-3.0/>

GNU General Public License GPLv3.0

- Conditioned on making available complete source code of licensed works and modifications
- include larger works using a licensed work, under the same license
- Copyright and license notices must be preserved
- Contributors provide an express grant of patent rights.

Permissions	Conditions	Limitations
<ul style="list-style-type: none">•Commercial use•Distribution•Modification•Patent use•Private use	<ul style="list-style-type: none">•Disclose source•License and copyright notice•Same license•State changes	<ul style="list-style-type: none">•Liability•Warranty

<https://choosealicense.com/licenses/gpl-3.0/>

GNU Lesser General Public License (GNU LGPLv3)

- Conditioned on making available complete source code of licensed works and modifications under the same license or the GNU GPLv3.
- Copyright and license notices must be preserved.
- Contributors provide an express grant of patent rights.

Permissions	Conditions	Limitations
<ul style="list-style-type: none">• Commercial use• Distribution• Modification• Patent use• Private use	<ul style="list-style-type: none">• Disclose source• License and copyright notice• <u>Same license (library)</u>• State changes	<ul style="list-style-type: none">• Liability• Warranty

<https://choosealicense.com/licenses/lgpl-3.0/>

The Unlicense

- Unlicensed works, modifications, and larger works may be distributed under different terms and without source code
- Example : Youtube-dl

Permissions	Conditions	Limitations
<ul style="list-style-type: none">• Commercial use• Distribution• Modification• Patent use• Private use		<ul style="list-style-type: none">• Liability• Warranty

<https://choosealicense.com/licenses/lgpl-3.0/>

Additional resources

- Case studies
 - <http://www.onlineethics.org/CMS/profpractice/ppcases.aspx>
 - http://computingcases.org/case_materials/case_materials.html
 - <http://seeri.etsu.edu/Ethics/ecases.asp>
- Ethical Dilemmas
 - <http://www.infoworld.com/article/2607452/application-development/12-ethical-dilemmas-gnawing-at-developers-today.html>
- General concepts
 - <http://c2.com/cgi/wiki?SoftwareEthics>
 - <http://www.ibm.com/developerworks/rational/library/may06/police/>
 - <https://www.scu.edu/ethics/focus-areas/more/engineering-ethics/an-introduction-to-software-engineering-ethics/>
 - <http://courses.cs.vt.edu/professionalism/WorldCodes/EDUCOM.software.html>
 - [https://www.researchgate.net/publication/277817334 Professional Ethics of Software Engineers An Ethical Framework](https://www.researchgate.net/publication/277817334_Professional_Ethics_of_Software_Engineers_An_Ethical_Framework)
 - http://www.slate.com/blogs/future_tense/2013/09/09/software_engineers_need_a_crash_course_in_ethics.html