

INFO5992 Understanding IT Innovations

Week 6: Distributed Innovation II:

Open Data,
Free and Open Source Software,

A/Prof Jinman Kim

Semester 1, 2018



Copyright warning

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

WARNING

This material has been reproduced and communicated to you by or on behalf of the University of Sydney pursuant to Part VB of the Copyright Act 1968 (**the Act**).

The material in this communication may be subject to copyright under the Act. Any further copying or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.

UoS Outline

Week	Lecture Topics	Activity	Assessments
1. 5 Mar	UoS Introduction; Definition of IT Innovation; IT Innovation System; IT Innovation in Australia	Tute 1 – Welcome to your tutorial; Importance of innovation to a Country	Form Groups
2. 12 Mar	Introduction to Technological / IT innovation; Examples of IT innovation in industry sectors; Type and Source of Innovation	Tute 2 – Massive Open Online Courses – Enabling technologies and Peer-review	
3. 19 Mar	Dynamics of Technological / IT Innovation; Adoption of Technology; Dominant Design	Tute 3 – Dominant design in the Smartphone market	Individual Report Introduction
4. 26 Mar	Disruptive Innovation; Industry Value Chain; Value Network analysis	Tute 4 – Cognitive IT services and its value chain	Quiz intro
Easter Break			
5. 9 Apr	Distributed innovation I: Open / Closed innovation; Platform innovation; Web APIs;	Tute 5 – Web API considerations	MCQ
6. 16 Apr	Distributed innovation II: Crowd innovations; Free and Open source software;	Tute 6 – Open source Geolocation and Maps	
7. 23 Apr	Distributed innovation III: User innovation; Open Data	Tute 7 – Sharing Economy from a Distributed Innovation Context	Group presentation Introduction
8. 30 Apr	Innovation by Start-up companies and Opportunities	Tute 8 – Business Model Canvas	
9. 7 May	Organisational Culture; Structure supporting innovation	Tute 9 – Group Presentation preparations and feedback	MCQ Report Submission
10. 14 May	IT Innovation Management	Group Presentation	Group Presentation submissions
11. 21 May	Innovation ecosystem; Sydney's innovation ecosystem	Peer-Review Marking	
12. 28 May	Judging IT Innovations	Tute 10 – Developing a Judging criteria for IT Innovation project	
4. 30 Jun	UoS Review; UoS comments / questions	Tute 11 – Technology innovations in IT Management	Peer-review

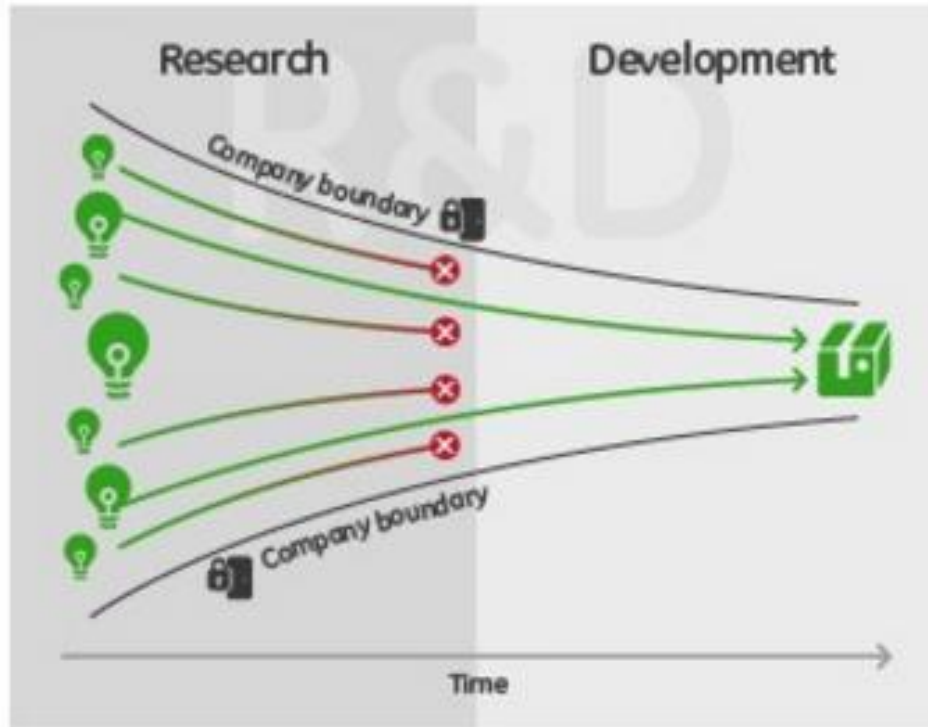
Agenda

- Distributed Innovation Part II
 - Open Data
 - Open Source
- Case Study on Open Source Maps
- Tute 6

Distributed Innovation – Open Innovation

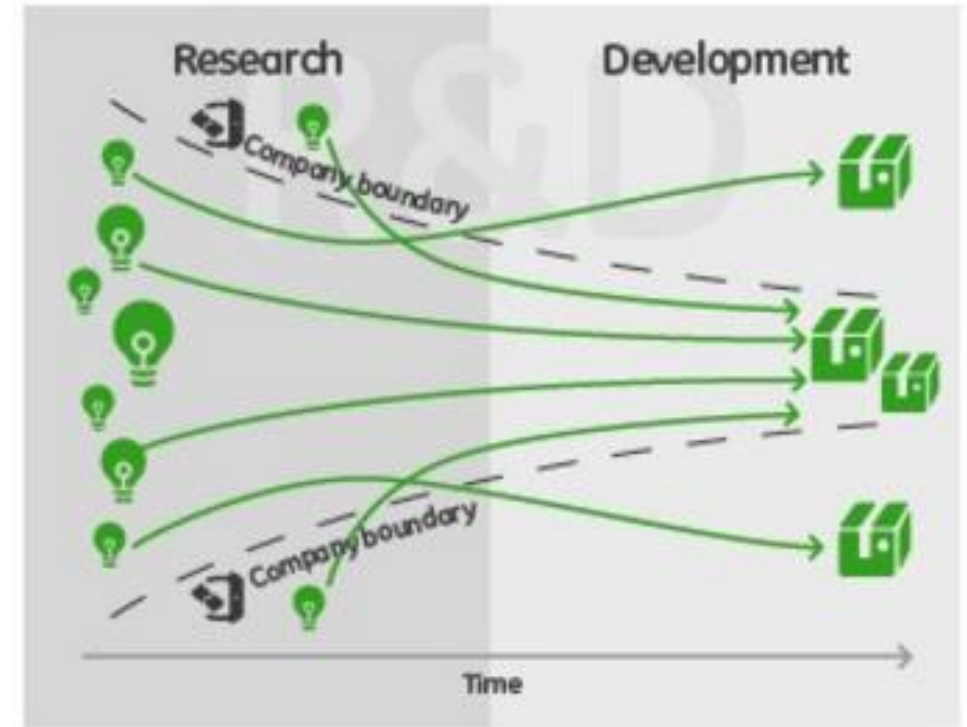
Recap: Closed and Open Innovation

Closed Innovation Concept



Source: Charts adapted from QuickMBA.com

Open Innovation Concept



VS

<http://www.geglobalresearch.com/blog/growing-middle>

Recap week5: Some approaches to distributed innovation

These are some approaches companies use to get external companies/individuals involved in their innovation:

- A. Product platforms
- B. Web APIs
- C. Crowdsourcing innovation / Crowdfunding Innovation
- D. Releasing data sets “Open data”
- E. Free and Open Source Software
- F. User innovation
- G. Platform ecosystems
- H. Accelerators, investment and others

D. Releasing data sets “Open data”

What is ‘Open Data’?

“Open means anyone can freely access, use, modify, and share for any purpose (subject, at most, to requirements that preserve provenance and openness).”

Put most succinctly:

“Open data and content can be freely used, modified, and shared by anyone for any purpose”

<https://opendefinition.org/>

Releasing data sets

- Many governments have opened up government data (“open data”)
 - In some cases, static data (eg tables of static data)
 - In some cases, live data feeds (eg an RSS feed or data service)
 - The Australian federal government <http://data.gov.au> includes:
 - Electoral boundaries
 - Crime data, census data
- NSW Government <http://data.nsw.gov.au/> - includes:
 - Bus stop data, Electricity consumption data, pollution education etc.

Releasing data sets

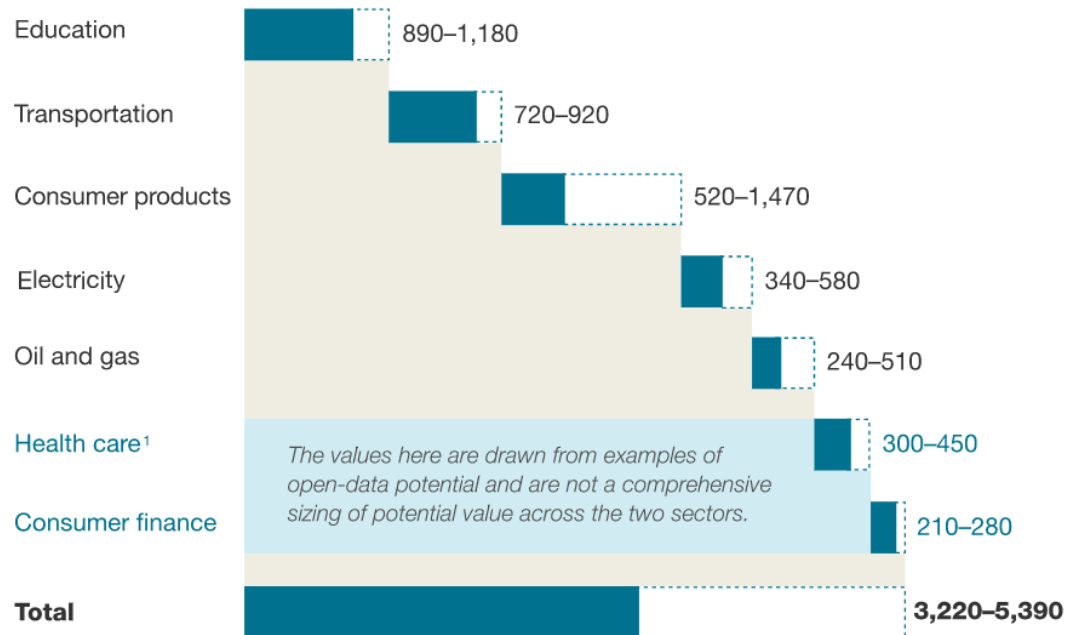
- Many communities are also building open data sets
 - E.g. www.openstreetmap.org, www.openaddresses.io
- Some companies are encouraging users to develop applications using their data
 - Examples of companies releasing data sets:
 - [GoGet car share data](#) , [Coca Cola Amatil data](#)

Value in Open Data

Exhibit

Open data can help unlock \$3 trillion to \$5 trillion in economic value annually across seven sectors.

Potential value in open data, \$ billion



¹Includes US values only.

Source: McKinsey Global Institute analysis

- **Economic value** e.g. increased efficiency, new products and services, and a consumer surplus (cost savings, convenience, better products)
- **Big data's impact** e.g., replacing or supporting human decision making
- **business opportunities** e.g., new product and services
- **Governments to play a central role**

Value in Open Data – Self-reinforcing cycle

- The benefits of open data can be self-reinforcing: they will increase as individuals perceive the advantages and help to improve the accuracy and detail of the information available.
- However, this cycle can gather momentum only if private industry and public agencies cultivate a vibrant open-data ecosystem and implement policies to protect stakeholders.
- For companies, that means putting in place the technologies and talent to collect and analyze data.
- For individuals—as both consumers and citizens—it means being vigilant, savvy providers and users of open data.

Example: Australian Government

Why Open Data?

"Data is a game-changer for government. Open data provides the intelligence for insight, invention and exploration that translate into better products and services that improve everyday life and encourage business growth."

The Hon. Victor Dominello, MP, Minister for Innovation and Better Regulation, launching the 2016 Open Data Policy



NSW Government Open Data Policy

Open Data

Open data is data that anyone can access, use, re-use or share.



Proactively released by government



Free or at the lowest cost



Well managed and trusted data sources



Up to date data sets



Available and easy to find use, re-use and share



Better, faster, safer, more open data

<http://data.nsw.gov.au/>

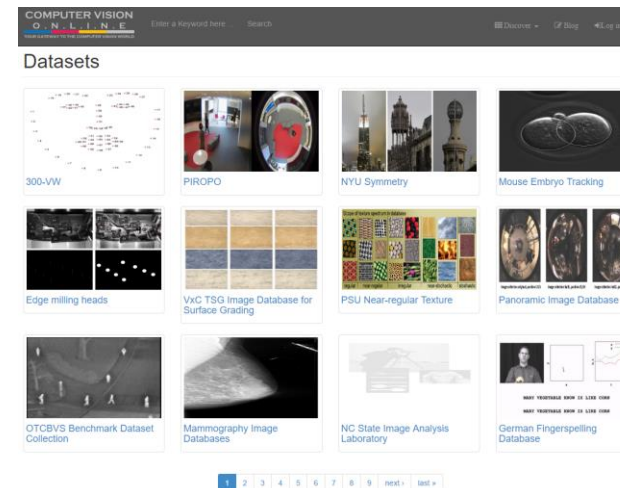
<https://www.ipc.nsw.gov.au/open-data-infographic>

Example: Computer Vision / Medical Imaging Community

- In Research community, open source and open data is important for
 - Benchmarking / Dissemination
 - Replication / Citations
 - Reputation / State of the Art
 - Building a community



<http://www.visceral.eu/>

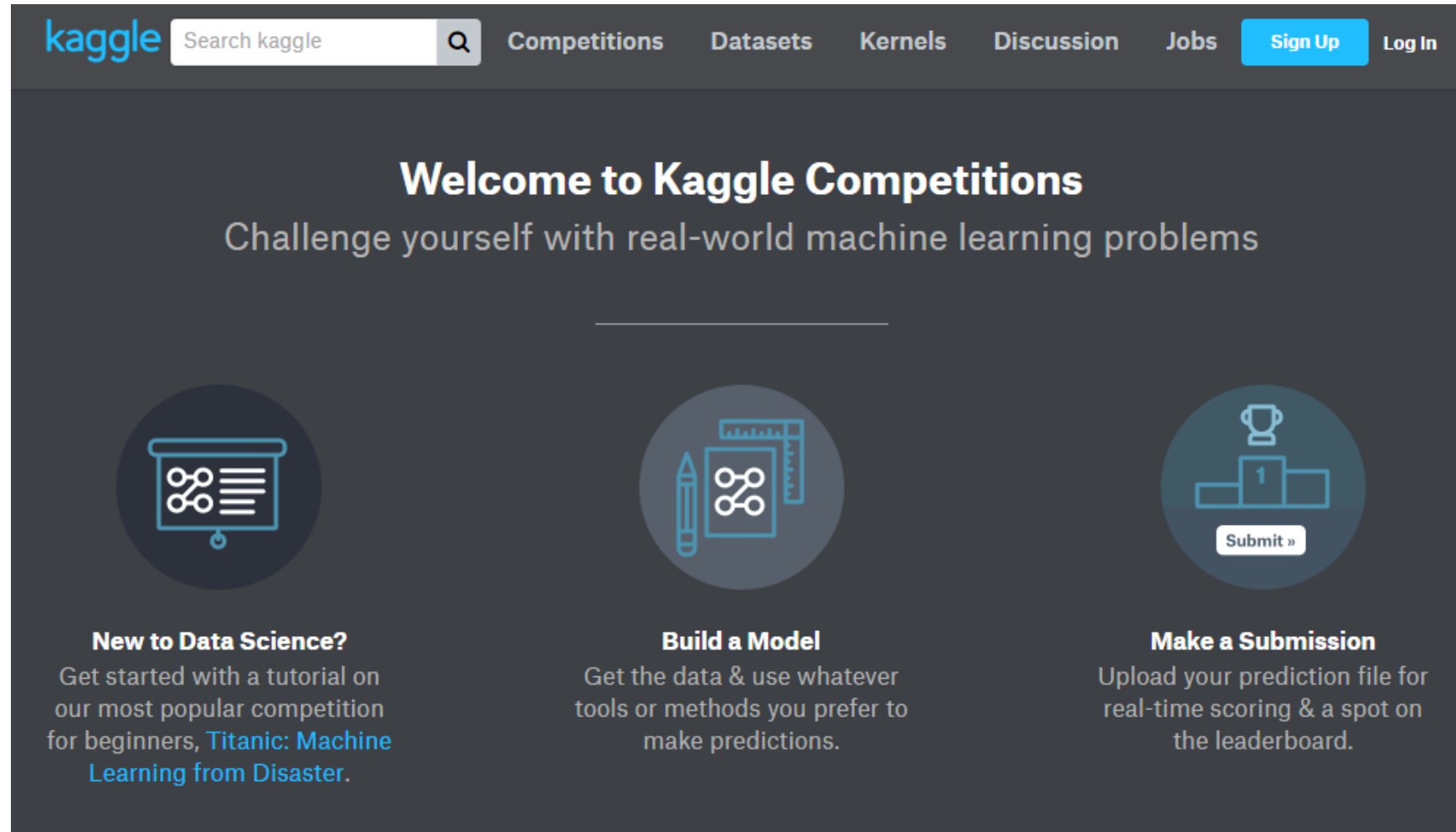


<https://computervisiononline.com/datasets>

<http://www.cvpapers.com/cvpr2017.html>

<http://www.cvpapers.com/cvpr2014.html>

Case Study – Kaggle – Big data competitions



<https://www.kaggle.com/competitions>

Kaggle story

- In 2010, **Kaggle** was founded as a platform for predictive modelling and analytics competitions on which companies and researchers post their data and statisticians and data miners from all over the world compete to produce the best models. **This crowdsourcing approach** relies on the fact that there are countless strategies that can be applied to any predictive modelling task and it is impossible to know at the outset which technique or analyst will be most effective. Kaggle also hosts recruiting competitions in which data scientists compete for a chance to interview at leading data science companies like Facebook, Winton Capital, and Walmart.
- In April 2015, Kaggle released the first version of their **Scripts product onto their platform**. Scripts allows users to write, run, and publicly share their code on Kaggle.
- In January 2016, Kaggle released their **Datasets product**, making a selection of public datasets available on Kaggle. Each datasets has Scripts enabled, as well as a dedicated forum, allowing for conversation and collaboration between data scientists and the work they are doing on each dataset.
- On 8 March 2017, Google announced that they were acquiring Kaggle.^[2] They will join the Google Cloud team and continue to be a distinct brand.^[3]

<https://en.wikipedia.org/wiki/Kaggle>

<http://www.afr.com/technology/google-buys-australias-kaggle-20170308-gutzx3>

Open collaborative innovation

Free and open source
software

Proprietary software vs Free and open source software

- Traditionally most software was proprietary
- Proprietary software is software built by or for a specific person, organization or group of organizations where:
 - The owner holds intellectual property rights over the software; and
 - The owner has total control over the software and how it is used
- In free and open source software:
 - Source code is made available
 - Source code can be changed and redistributed by others
 - (more precise definitions coming later)

Some examples

Proprietary software



(Many of these also use open source within them)

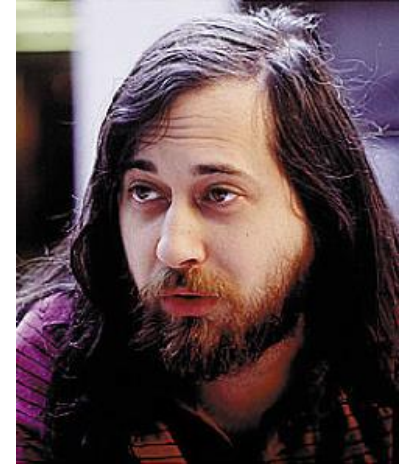
Free and open source software



(Not all versions of the above are open source)

Free and open source software: Example GNU

- Richard Stallman was a programmer at MIT AI Lab
- In the 1970s, many manufacturers (eg of copiers) supplied source code (eg drivers)
- Stallman (and others) modified software to meet their needs
- Companies started to be more protective of their source code.
- In 1980, Stallman and others were refused access to the source code for the software of the first laser printer (from Xerox)
- Stallman was not able to modify the software and this was inconvenient for him
- This helped convince Stallman that people should be free to modify all software.



Richard
Stallman

*"technical
means to a
social end"*

Free and open source software: Example GNU

- 1983: Richard Stallman started GNU
- Goal: completely “free” version of Unix (full operating system)
 - (Unix had first been released by AT&T in 1971)
- By 1990s, most components ready but not kernel or drivers



<https://www.gnu.org/home.en.html>

Free and open source software:

Example Linux

- Background to Linux – Linus Torvalds:
 - Linus wanted to build an operating system kernel to improve his skills
 - He didn't want to use Unix as it was proprietary
 - He couldn't use GNU's kernel as there was not yet a reliable kernel
 - He didn't want to use MINIX as there was a fee
 - So he wrote the Linux kernel and did this with community involvement
 - The Linux kernel was used with the rest of the GNU operating system to form GNU/Linux (usually what we know of as just "Linux")



Linus Torvalds

"Linux"

What is Free Software?

(using Free Software Foundation definition)

- Free Software Foundation (FSF)
- Started by Richard Stallman in 1985
- “Free” as in “free speech” not as in “free beer”



[https://en.wikipedia.org/wiki/Gratis_versus_libre#/media/File:Gael_RMS - free as free speech, not as free beer.png](https://en.wikipedia.org/wiki/Gratis_versus_libre#/media/File:Gael_RMS_-_free_as_free_speech,_not_as_free_beer.png)

What is Free Software?

(using Free Software Foundation definition)

- “Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it means that the program's users have the four essential freedoms:
 - The freedom to run the program, for any purpose (freedom 0).
 - The freedom to study how the program works, and change it to make it do what you wish (freedom 1). Access to the source code is a precondition for this.
 - The freedom to redistribute copies so you can help your neighbor (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. “

Stallman's Argument

- Computer software is becoming more and more critical for the running of a free society
- If that software is controlled by companies or governments, the software can be used to restrict or monitor people
- So, it is necessary that source code be available for all software
- If source code were not freely available, a limited number of very powerful people would dominate computing

Free Software Foundation <http://www.fsf.org/about/>
and
DiBona, Chris (ed) (1999). *Open Sources: Voices from the Open Source Revolution*.
O'Reilly & Associates: Sebastopol. pp.2.

“Copyleft”

- Play on word “copyright”
- “Copyleft is a general method for making a program (or other work) free, and requiring all modified and extended versions of the program to be free as well.” (Free Software Foundation)
- Example of a copyleft licence is the GNU Public License (GPL)

<http://www.gnu.org/copyleft/>



Copyleft symbol

Open Source Software

- Concept of Open Source Software started by Eric Raymond
- Launched in 1998
- He said that the term/concept “free software” was:
 - Ambiguous (e.g. confusion with freely downloadable binaries and shareware)
 - Intimidating to companies (due to ideological approach)
- He focussed on the pragmatic aspects of providing access to source code, rather than the “rights” of users.



Eric Raymond

"Given enough eyeballs, all bugs are shallow"

What is Open Source Software?

(using Open Source Initiative definition)

- To be classified as OSS, the software must be (according to its licence):
 - Freely redistributable
 - Source code must be available for free or at reasonable reproduction cost
 - Modifications and derived works must be allowed and be distributable under same terms
 - Can protect integrity of author's source code as long as allow source code patches
 - No discrimination against people/groups
 - No discrimination against fields of endeavour
 - Must not be restricted to use with a specific product
 - Must not place restrictions on other software distributed with it
 - Must be technology-neutral

Difference between Free Software and Open Source Software



Tim O'Reilly
O'Reilly Media

Richard Stallman
Free Software Foundation

Photo: Julian Cash, OSCON 2002

Difference between Free Software and Open Source Software

- According to Stallman, "Open source is a development methodology; free software is a social movement."
- Open Source covers a wider range of licence types
- More ability to mix Open Source software with proprietary software than is the case for free software
- The Open Source concept was developed to bring major software businesses and other high-tech industries into the mix.
- When avoiding distinguishing between these, people use the terms:
 - FOSS (Free and Open Source Software); or
 - FLOSS (Free/Libre and Open Source Software)

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

Free software and open source software: Examples

- **OSS and copyleft** (changes to the source must be made available to others)
 - The Linux kernel
 - MariaDB (database software based on MySQL codebase)
 - Eucalyptus (for building private clouds – company bought by HP)
- **OSS and not copyleft** (changes to the source do not need to be made available to others)
 - Apache web server
 - OpenCV (Computer Vision library originally by Intel)
 - Chromium (the core of Google Chrome web browser)

*Note: It's a bit more complicated than this as some of this software is available under multiple licences.
More later on OSS licences.*

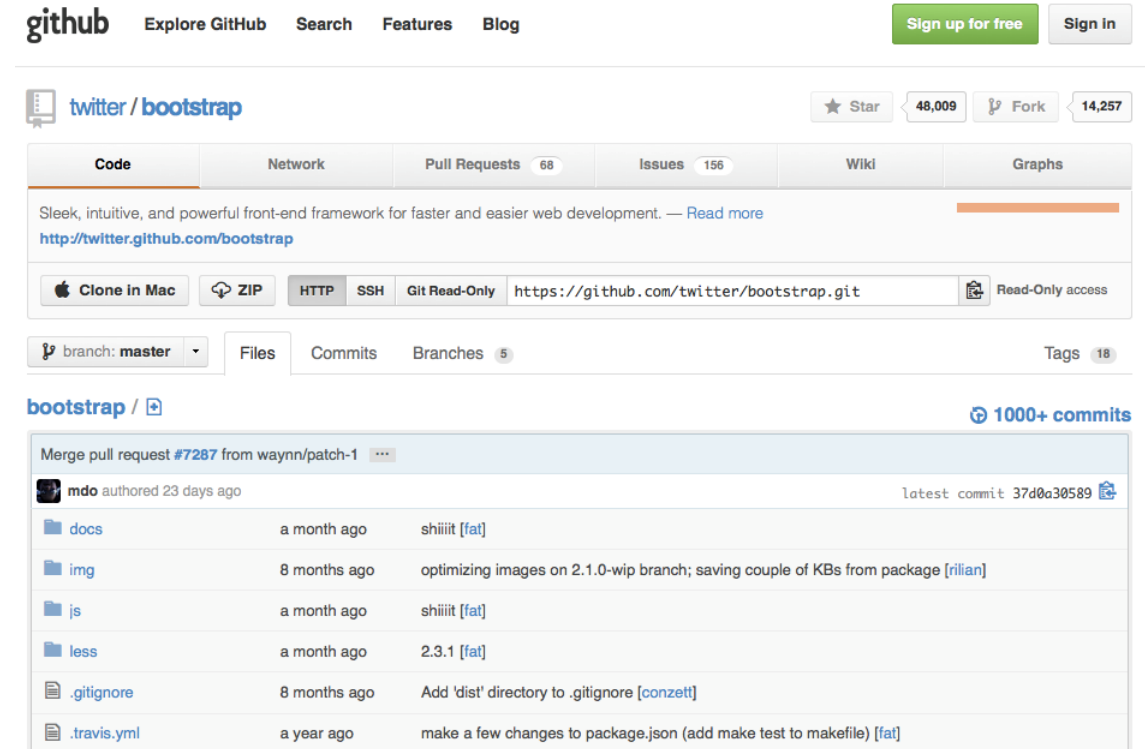
Open source hosting sites

- Offer hosting, version control, issue tracking, wikis, download support etc
- Some support code reviews etc
- Examples:
 - Github (>10 million repositories)
 - Sourceforge (>400k projects)
 - Google Code (250k projects) (closed down Jan 2016)

Github

- Github (>10 million repositories):
 - E.g. Ruby on Rails (web application framework), jquery (JavaScript query engine), node.js (evented i/o for javaScript), Diaspora (distributed social networking)
- GitHub is free to use for public and open source projects. Work together across unlimited private repositories with a paid plan.

—



Importance of FOSS for innovation

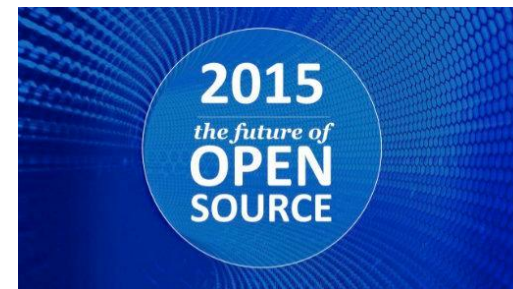
Growing importance of free and open source software

- Many companies use open source software such as for:
 - Internal IT infrastructure (eg Linux)
 - Building and running web services (eg Apache, Apache Tomcat, JBoss)
 - Building software for redistribution
- Open source software allows companies to rapidly innovate their infrastructure and services

Open Source usage in Enterprises

- Survey by North Bridge and Black Duck Software
- Input from 1 300 senior IT professionals, in 2015
- 78% of companies run all or part of their operations on open source software
 - Compared to 42% in 2010
- 3% of companies said they don't use open source software
- 64% say that their company participates in open source projects
- 66% say that they consider open source software before considering proprietary software

Source: <https://opensource.com/business/15/5/report-future-open-source-survey>
http://www.slideshare.net/North_Bridge/2015-future-of-open-source-study



Open source for innovation



http://www.slideshare.net/North_Bridge/2015-future-of-open-source-study

Importance of FOSS in R&D and startups

- Most infrastructure used in R&D and startups uses FOSS:
 - Operating systems (e.g. Linux)
 - Containers (e.g. Docker)
 - System configuration management (eg Puppet, Chef)
- Most new software is built using FOSS:
 - Software platforms (e.g. Java, Scala, Python, Ruby on Rails, node.js)
 - Software libraries/frameworks (e.g. Spring framework, glibc)
 - Software build and test automation (e.g. Jenkins, Cucumber)
- Most new software contains FOSS:
 - To reduce time and cost of development
 - To reduce testing and maintenance costs (assuming using stable FOSS)
 - To provide compatibility with other software
 - To focus on the core differentiator of your own software

Some open source business models

- Sell support and services
 - Example: Canonical (with Ubuntu)
- Sell certified version (with support and services)
 - Example: Cloudera (with Hadoop)
- Sell “enterprise edition” (effectively proprietary software)
 - Example: MySQL “standard edition” (not “community edition”)
- Dual licensing (copyleft so need commercial license if modify source)
 - Example: Digia (with Qt)
- Other advantages to the company
 - Example: Google (with Android)

Video: Tim O'Reilly on Open Source Business Models



<http://www.youtube.com/watch?v=GYarQ1r2yZo>

Example: Facebook and Open Source

- Accelerates innovation in the world. Users can build apps more quickly and Facebook can benefit from the improvements that others make to their code.
- If the company knows something will be open from the start, it just builds it better so that it can be more accessible and dependable because it's going to be used in the outside world.
- Open source provides opportunities to share challenges. It attracts the interest of people who want to work on these challenges, and as a result it helps improve the quality of the company's staff.



<https://opensource.com/business/15/7/keynote-oscon-james-pearce-facebook>

<https://code.facebook.com/>

Example: Apple and Swift programming language

- Apple is among the most closed of tech companies, yet it is doing open source with its **Swift** programming language
- Swift is open source, and they want you to help make it the best general purpose programming language available everywhere.
- The success of a software platform is proportional to the number of developers that use it.
- Apple clearly wants to attract the best developers in the world to create new apps and desktop applications, which showcase its latest iPhones, Apple Watches and MacBook devices.



<http://www.computerweekly.com/news/450296755/Why-Apple-is-wooing-open-source-developers-with-Swift>
<https://swift.org/>

Example: Microsoft and .Net

- Microsoft is porting its server-side .NET stack to Linux and Mac OS X, and is making more of that stack available as open source. With its engineers involved in more than 2,000 open source projects, you'd have to agree that open source has more than a foothold at Microsoft these days.
- Microsoft also wants to bring technologies to Linux, in large part because of Azure.
- Running a cloud platform gives Microsoft an interest in Linux that goes far beyond the open source contributions the Windows Server team has been making to the Linux kernel
- As of September 2015, more than 20 percent of the virtual machines running on Azure IaaS (Infrastructure as a service) were Linux.
 - “As we pursue our vision of the fabric and the cloud anywhere, that is as much a story about supporting Linux workloads as it is Windows workloads,” says lead architect for Windows Server, Jeffery Snover.



<http://www.zdnet.com/article/microsoft-to-open-source-more-of-net-and-bring-it-to-linux-mac-os-x/> 2014

<http://www.cio.com/article/3026664/open-source-tools/the-real-reason-microsoft-open-sourced-net.html> 2016

Open source lab model (eg Amplab at UC Berkeley)

- Berkeley Lab: Multi-year collaborative effort at UC Berkeley
- Sponsors: Amazon, Google, IBM, SAP + 19 more
- Open Source Software developed:
 - Spark (cluster computing framework), Mesos, Tachyon, GraphX, MLBase
- Companies formed:
 - Mesosphere (\$122.25M invested), Databricks (\$47M invested), Tachyon Networks (\$17M invested)



<https://amplab.cs.berkeley.edu/>

Amplab: benefits to participants



- What do the companies get:
 - Deep knowledge of technology as it is developed
 - Influence technology direction and outcome
 - Protection from disruption by other companies
 - Access to university talent pool
- What does the university get:
 - Clear focus for computer science research
 - Additional income to fund activities
 - Close interaction with market needs
 - Incubation of new businesses
 - Stronger innovation ecosystem

Challenges in using FOSS in products and services

- Meeting obligations of software licenses (ensuring appropriate notices, etc)
- Possibility of accidentally “contaminating code”
 - E.g. a programmer introduces some GPL (General Public Licence) code from the Internet into some proprietary product code and then the product is released
 - legally, the company should release the proprietary source code
- Ensuring adequate quality of the final product if it includes some open source software of unknown quality
- Avoiding security vulnerabilities in underlying code (that may already be known to hackers)

How can companies address the challenges

- Companies developing products (hardware or software) or services and using open source software should have an open source policy and controls to ensure good governance.
- According to a Gartner report, <50% of Global 2000 IT Companies were planning to implement an open source governance program by 2014.
- In a related survey, only 1 / 3 of companies surveyed had an OSS policy

Source: <http://blog.blackducksoftware.com/2011/07/29/policy-precedes-controls/>

Tools for managing open source software

- Most open source analysis tools provide ways for checking if open source software is used and check if usage conforms to a policy (eg a company OSS policy)
 - Black Duck Software (<http://www.blackducksoftware.com/>)
 - Software tools and online Knowledgebase containing open source software
- Palamida (<http://www.palamida.com/>)
- FOSSology (<http://www.fossology.org/>)
 - Developed by HP – released as open source software
 - Openhub.net (<http://code.openhub.net/>)
 - Now owned by Black Duck Software
 - Free online service for searching open source code
 - Searching of over 21b lines of FOSS code
 - For a comprehensive list of tools - http://fossology.org/links_-_related_projects

FOSS Licenses

Usage and Licenses

Obligations when using open source software

- The obligations depend on the actual software licence used by the software
- Your obligations may include:
 - Nothing (i.e. no special obligations); or
 - If you redistribute the open source software in your software:
 - Mentioning that you have used it; or
 - Redistributing any changes you made to it; or
 - Not suing other companies in relation to patents you may hold related to the features of the open source software; etc

Summary of Main open source licences

Permissive licences:
Changes need not be made available

Restrictive (copyleft) licences:
Changes must be made available

Public
domain

MIT

BSD

Apache
Software
License

GPLv2

GPLv3

AGPL

SleepyCat



- More restrictions/conditions on users
- More assurances of software staying free

Public domain

- Work in the public domain does not have intellectual property rights
 - (eg the right has expired or has been deliberately placed in the public domain)
- Examples: the English language, Shakespeare's works, Beethoven's music, many old photos for which copyright has expired
- Not commonly used for software because:
 - As software development is a recent activity, copyright hasn't expired yet
 - Author can't make disclaimer (unlike open source licences)

Massachusetts Institute of Technology Licence (MIT License)

- User can do anything with the software...
- But they must make sure that the copyright of the original author is maintained
- No warranty

Copyright (C) <year> by <copyright holders>

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.

Berkeley Software Distribution Licences (BSD Licences)

- Similar to MIT Licence but, if redistribute software is using it, it must acknowledge its use
- 4-clause (original), 3-clause (“modified”) and 2-clause (“simplified”) versions exist
- 3-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.
- Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Removed
in simplified
version

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "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.

GNU General Public Licence (GPL)



- More restrictive than MIT and BSD – it is copyleft
- You can use the code and change it, but you must release all modified code under the same licence and any other code of yours that touches it
- 2 main versions - GPL v2 and GPL v3
- <http://www.gnu.org/licenses/gpl.html>
- Clause 5 of GPL v3:
 - You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:
 - a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
 - b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
 - c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
 - d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.
 - A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

When creating open source software: How do you know what licence to use?

- Or use “Dual-licensing” – this is now very common
- Build up the market first and then provide services
- Eg:
 - Software can be licensed as GPL or proprietary licence
 - If a company doesn't want to make their changes available, they can come to you to negotiate a proprietary licence

When creating open source software: How do you know what licence to use?

Permissive licences:
Changes need not be made available

Restrictive (copyleft) licences:
Changes must be made available

Public
domain

MIT

BSD

Apache
Software
License

GPLv2

GPLv3

AGPL

SleepyCat



If:

- You want a lot of companies to adopt your software in their products/services, and
 - You don't care if they make their changes available (eg as you just want the code to be used or you have deep enough knowledge & expertise that they will come back to you):
- => use a permissive licence (eg BSD, Apache)

If:

- You want to ensure that companies (using your software in their products) make their changes available (so you and others can get them):
- =>use a restrictive licence (eg GPLv3)

Example: Open CV



- OpenCV is released under a BSD license and hence it's free for both academic and commercial use.

OpenCV license

By downloading, copying, installing or using the software you agree to this license.

If you do not agree to this license, do not download, install, copy or use the software.

License Agreement
For Open Source Computer Vision Library
(3-clause BSD License)

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

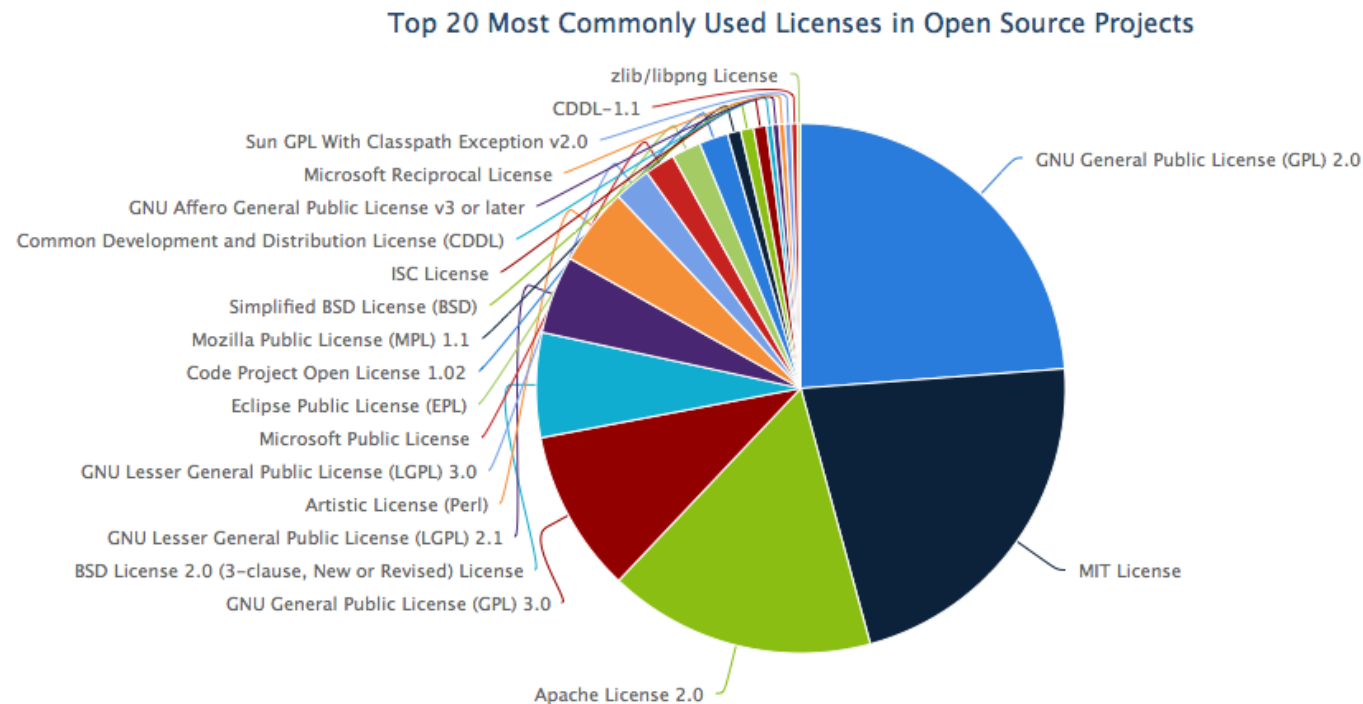
- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of the copyright holders nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors "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 holders or contributors 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.

<http://opencv.org/license.html>

Most common open source licences

- The Black Duck Software Knowledgebase contains OSS packages from:
 - Over 1.1 million open source projects from more than 8,500 sites
 - Using 2,400 different software licences



Source: <http://www.blackducksoftware.com/> (Sept 2015)

Summary

- Free software and open source software are growing in importance for:
 - Enterprise IT;
 - R&D in established companies; and
 - Startups
- One factor leading to the high rate of IT innovation is the availability of open source software
- Creating innovative new software services/startups is now accessible to us all thanks to the web and open source software

References

- Baldwin, Carliss, and Eric Von Hippel. "Modeling a paradigm shift: From producer innovation to user and open collaborative innovation." *Organization Science*, Vol 22, No 6, 2011.
- C. DiBona (ed) *Open Sources: Voices from the Open Source Revolution*, O'Reilly & Associates, 1999. Available online: <http://oreilly.com/openbook/opensources/book/>
- Karl Fogel, *Producing Open Source Software: How to run a successful free software project*, O'Reilly Media, 2005. Available free online: <http://producingoss.com/>
- R. Goldman and R. Gabriel, *Innovation Happens Elsewhere: Open Source as Business Strategy*, Morgan Kaufmann Publishers, 2005. Available free online: <http://www.dreamsongs.com/IHE/>
- Melissa A. Schilling, *Strategic Management of Technological Innovation*, 4th edition, McGraw-Hill, 2013.
- R. Stallman, *The Cathedral and the Bazaar*, <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>, 1999.
- E. von Hippel, *Democratizing Innovation*, MIT Press, 2005. . Available free online: <http://web.mit.edu/evhippel/www/democl.htm>

Case Study

Innovations with Maps



https://commons.wikimedia.org/wiki/File:World_Map_1689.JPG

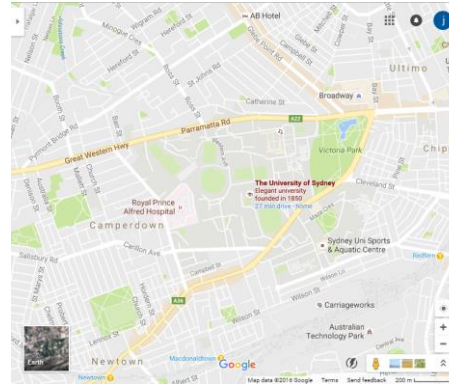
Geographic Information System

- The past decade has changed GIS immensely. With the advent of data mining, machine learning, mobile applications, the Internet of Things, social media, and other recent additions to the computing landscape, there are orders of magnitude more data available across every discipline, and the race to do useful and interesting things is in full throttle.
- Once limited to more traditional realms like land records, agriculture, natural resources, and urban planning, GIS now permeates practically every field.

<https://opensource.com/life/15/11/getting-started-web-mapping>

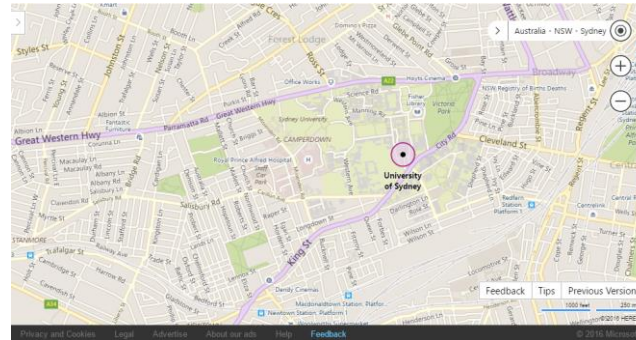
Maps?

— Google Maps



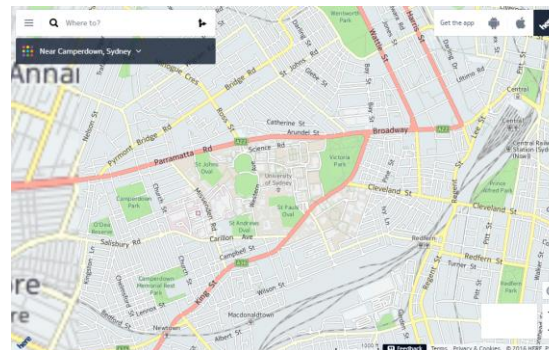
<https://www.google.com/maps/>

— Bing Maps



<https://www.bing.com/maps>

— Here WeGo Map



<https://wego.here.com/>

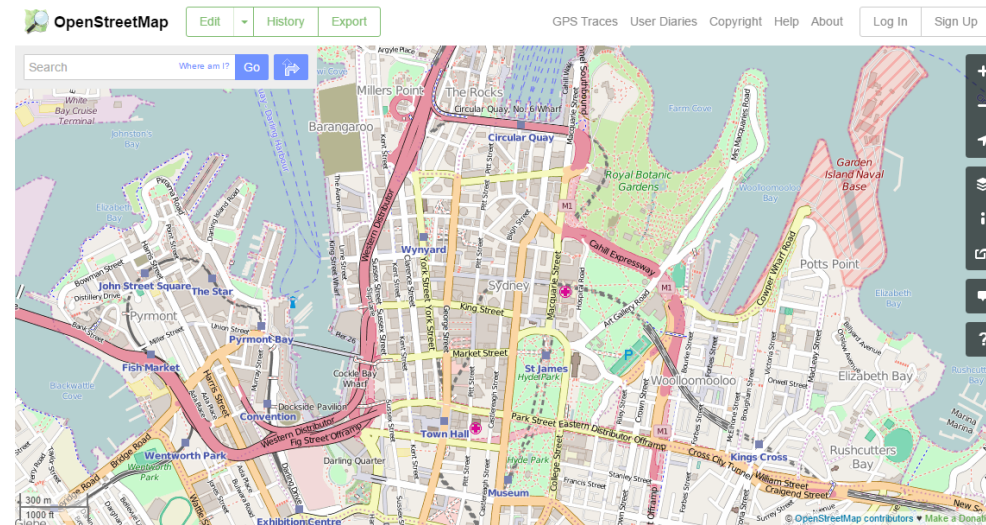
An open source map

— Open Street Map



Steve Coast

<http://www.allthepeople.net/steve-coast>



<http://www.openstreetmap.org/#map=15/-33.8680/151.2100>

M Haklay, and P Weber, “Openstreetmap: User-generated street maps”, *IEEE Pervasive Computing*, 2008 - ieeexplore.ieee.org

OpenStreetMap Foundation

- OpenStreetMap is an initiative to create and provide free geographic data, such as street maps, to anyone.
- The OpenStreetMap Foundation is an international not-for-profit organization supporting, but not controlling, the OpenStreetMap Project.
- It is dedicated to encouraging the growth, development and distribution of free geospatial data and to providing geospatial data for anyone to use and share.

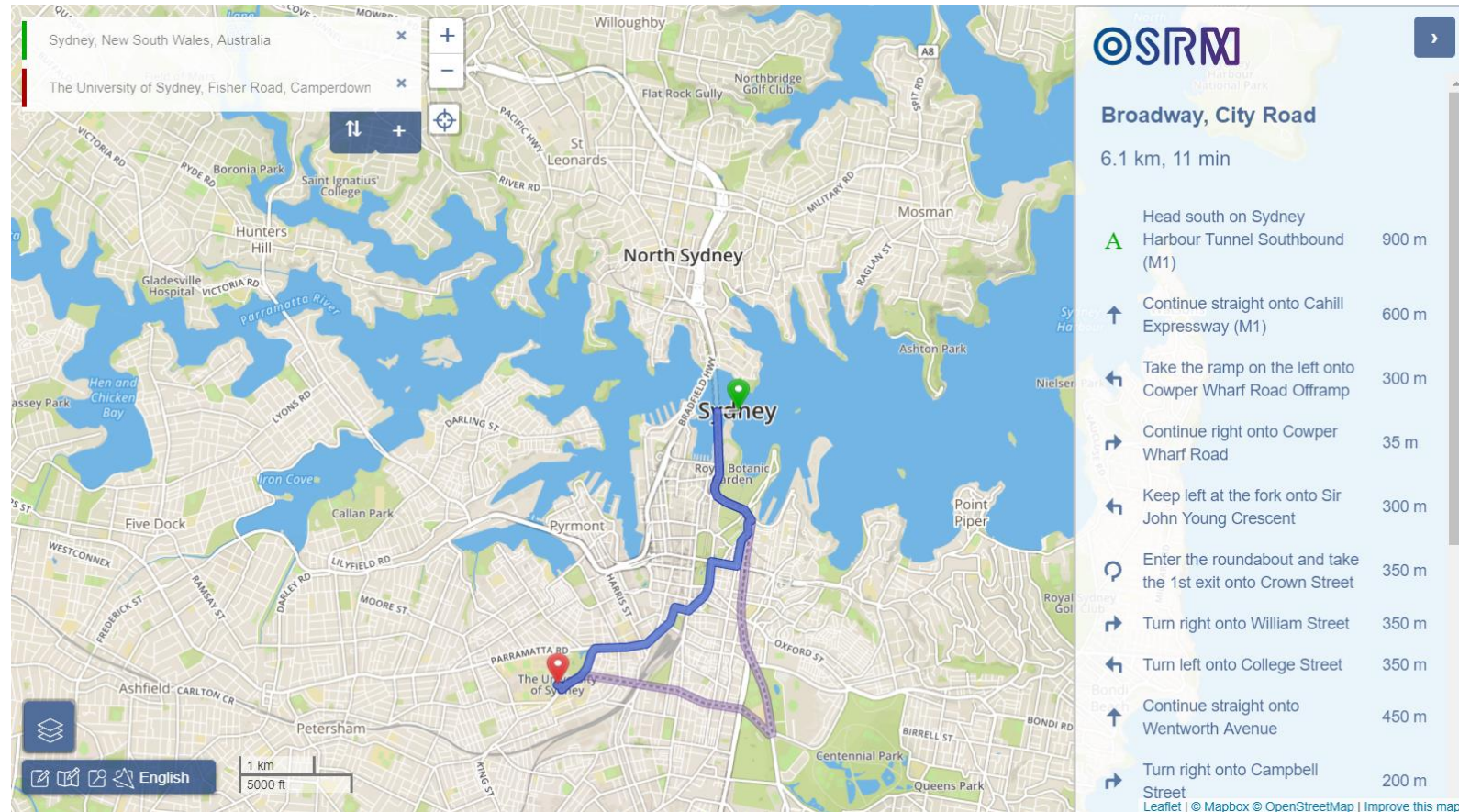


http://wiki.osmfoundation.org/wiki/Main_Page

OpenStreetMap

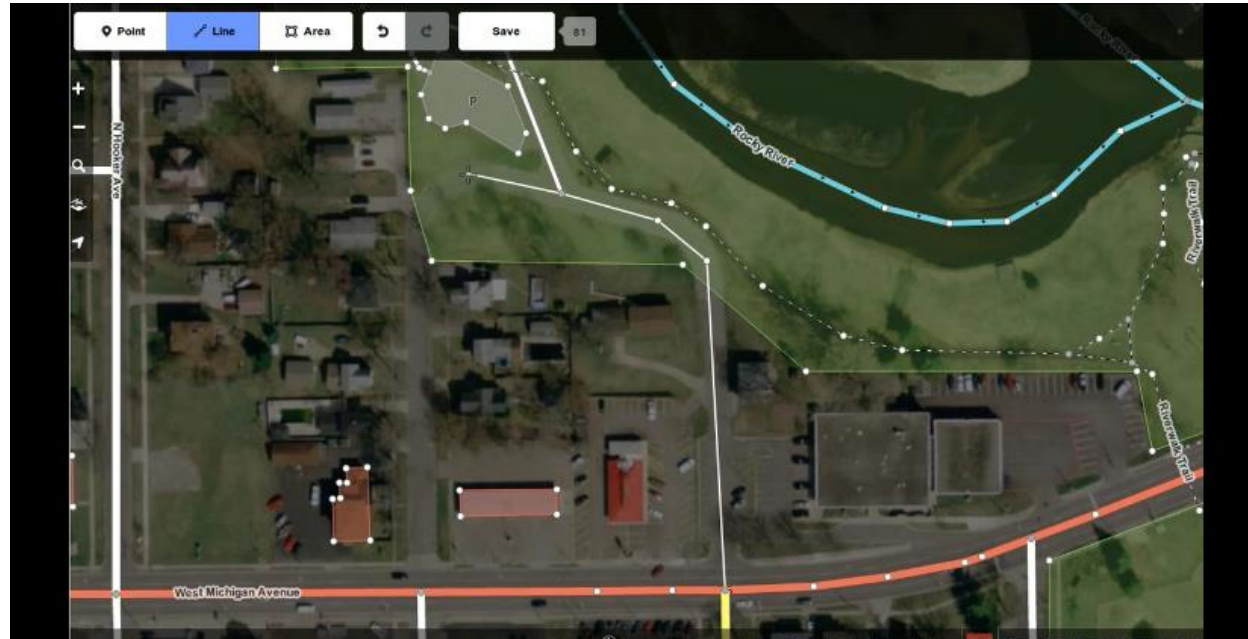
- Open source – enable innovations to happen
 - introduce services / businesses
 - for non-profit communities
- Crowd sourcing – over 2M users collecting data using manual survey, GPS and other free sources
 - Government and ‘out of copyright’ data
 - Innovating through the power of the crowd
- Open data license – share left
- Leading to many new innovations!

Open Source Routing Machine



<http://project-osrm.org/>

Mapping is easier than you think!



<https://www.mapbox.com/blog/new-map-editor-launches-openstreetmap/>

Other Open Source Maps



- A JavaScript library for image- and vector-tiled maps using SVG.

<http://polymaps.org/>



- an open-source JavaScript library for mobile-friendly interactive maps

<http://leafletjs.com/>



- A small, extensible, and free library for designers and developers who want to use interactive maps in their own projects.

<http://modestmaps.com/>



Users of Open Source Maps....

- Pokémon Go is a mobile augmented reality game by Niantic in which the player catches and collects Pokémon that "spawn" at locations all over the map. The game was released for Android and Apple iOS in July 2016.
- Niantic switched the base map from Google Maps data to OpenStreetMap data in December 2017.
- Since its initial release, many Pokémon Go players have edited OpenStreetMap in an attempt to improve their gameplay. Many players have made good-faith edits to OpenStreetMap, such as adding buildings and roads that reflect reality on the ground. On the other hand, there have also been many cases of Pokémon Go players adding spurious parks or deleting schools in an attempt to essentially tag for the renderer.
- Pokémon Go players are strongly encouraged to improve the map but are expected to follow good practice like any other mappers.

https://wiki.openstreetmap.org/wiki/Pok%C3%A9mon_Go

Tutorial 6

Understanding the Innovation
and Business model of Map
Web APIs

Tutorial Qs

- **Q1.** What are the **Distributed Innovation concepts** employed by mapping API companies? Does the model of OpenStreetMap differ from the paid APIs? We have studied the following concepts in the past two weeks:
 - Product platforms
 - Web APIs
 - Crowdsourcing innovation / Crowdfunding Innovation
 - Releasing data sets “Open data.”
 - Free and Open Source Software
 - User innovation
- **Q2.** The Map API has created an extensive and powerful Value Network. In our case study today, we learned about several companies based upon the freely available OpenStreetMap. Can you add more examples to the Value Network (services, categories of services etc)
- **Q3.** Comparing the OpenStreetMap to others, they embrace ‘user innovation’ and ‘crowdsourcing innovation’ as their core differentiator. What is your view of this and its long-term potential? Could they become at the level of Wikipedia i.e., the dominating service?
- **Q4.** [Optional / Homework] Map Web APIs has had a huge impact in many industries e.g., shopping, Airbnb, ride sharing, navigation, etc. Would you consider Map APIs to be a destructive innovation? Has it created its own Value Chain and also created a New Market? Would you consider Maps to be in an Era of ferment, where we are seeing incremental innovations, and therefore it is prime for a new disruption? What would that look like and is there companies/research targeting this change?

Playing with Maps

4

Run

Save

Tidy

Collaborate

Fiddle Meta

Untitled fiddle

No description

Add title to make the fiddle public.


External Resources

AJAX Requests

Legal, Credits and Links

JSFiddle Roadmap

suggest and vote for features



Slack - a messaging app for teams, integrating with the tools you already use.

ads via Carbon

app_code: "AJKnXv84fjrb0KIhawS0Tg",
useCIT: true,
useHTTPS: true
});
var defaultLayers =
platform.createDefaultLayers();

//Step 2: initialize a map - not specifying a
location will give a whole world view.
var map = new
H.Map(document.getElementById('map'),
defaultLayers.normal.map);

//Step 3: make the map interactive
// MapEvents enables the event system
// Behavior implements default interactions for
pan/zoom (also on mobile touch environments)
var behavior = new H.mapevents.Behavior(new
H.mapevents.MapEvents(map));

// Create the default UI components

No need for the HTML tag, it's already in the output.
No need for the META tags.
No need for the HEAD tag, it's already in the output.
For JavaScript use the panel below or the Resources panel for external files.
For external CSS files use the Resources panel on the left.

HTML

1

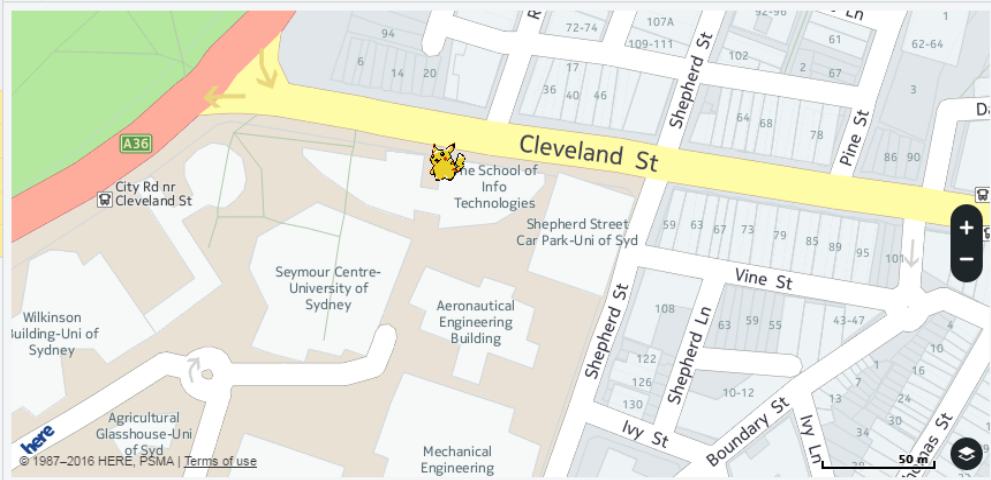
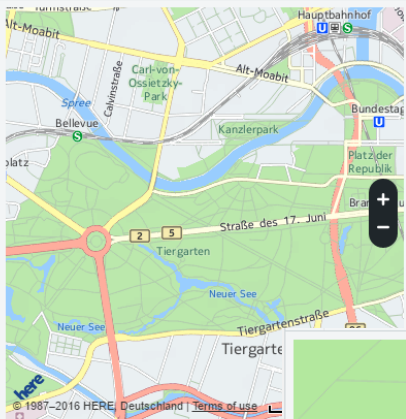
Settings

Sign in

CSS

1

JAVASCRIPT



The University of Sydney

Page 77

Assessment

Group Presentation