

# **OpenStreetMap**

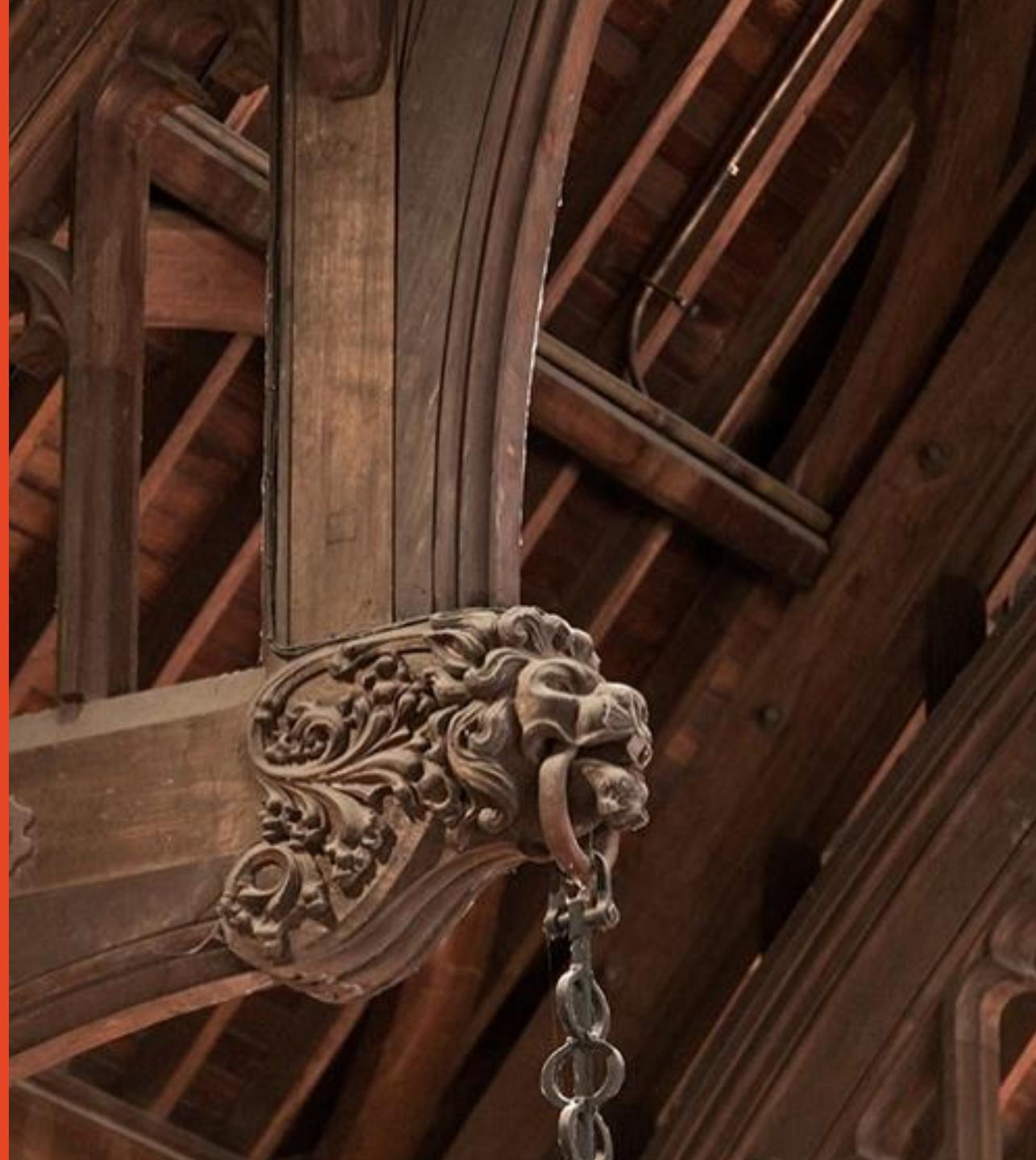
- Crowd sourcing
- Open source
- User innovation
- Leading to new innovations?

# **INFO5992 Understanding IT Innovations**

**Week 6: Distributed Innovation:  
User Innovation,  
Free and Open Source Software,  
Platform Ecosystem**

A/Prof Jinman Kim

Semester 1, 2017



# 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
1. 6 Mar	UoS Introduction; Definition of Innovation; Innovation System; Innovation in Australia	N/A
2. 13 Mar	Introduction to Technological / IT innovation	<b>Tute 1 – Massive Open Online Courses – Enabling technologies and Peer-review</b>
3. 20 Mar	Dynamics of Technological / IT Innovation; Source of Innovation; Adoption of Technology; Dominant Design	<b>Tute 2 – Design Dominance in the Smartphone market</b>
4. 27 Mar	Disruptive Innovation; Industry Value Chain; Value Network analysis	<b>Tute 3 – Innovative Tech Practice – Cognitive services</b> <i>Group Presentation Introduction – Topics Released</i>
5. 3 Apr	Distributed innovation I: Open / Closed innovation; Platform innovation; Web APIs; Crowdsourcing / crowdfunding	<i>Mid-semester Quiz</i> <i>Group Presentation – Topic Selection</i> <i>Individual Assignment Introduction</i>
6. 10 Apr	Distributed innovation II: User innovation; Free and Open source software; Open Data	<i>Peer-review Introduction</i> <b>Tute 4 – Innovative Tech Practice – Open source Geolocation and Maps</b>
<i>Easter (Break)</i>		
7. 24 Apr	Platform ecosystems	<i>Group Presentations I – IT Innovation Case Studies</i> <i>Peer-review of Group Presentations</i>
8. 1 May	Group Presentations II – IT Innovation Case Studies	<i>Peer-review of Group Presentations</i>
9. 8 May	Group Presentations III – IT Innovation Case Studies	<i>Peer-review of Group Presentations</i>
10. 15 May	Innovation in Industry sectors ( <b>Lawrence – Microsoft*</b> <b>Dr Ashnil Kuamr</b> )	<b>Tute 5 – Judging IT Innovation (Example in the Healthcare sector)</b>
11. 22 May	Innovation ecosystem; Sydney's innovation ecosystem Organisational Culture; Structure supporting innovation ( <b>Bill Simpson – Data61</b> )	<b>Tute 6 – Sharing Economy</b> <i>Individual Assignment Submission</i>
12. 29 May	Innovation by Start-up companies and Opportunities	<b>Tute 7 – Business Model Canvas</b>
13. 5 Jun	Organisational Culture; Structure supporting innovation UoS Review	<i>UoS comments / questions</i>

# **Agenda**

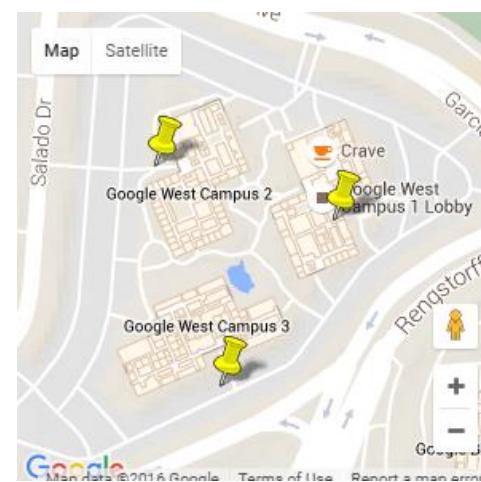
- Distributed Innovation Part II
- Feedback on Presentation topics; Peer Review Procedure Explanation
- Tute 4

# Tutorials

- Massive Open Online Courses – Enabling technologies and Peer-review
- Design Dominance in the Smartphone market
- Innovative Tech Practice – Cognitive services
- **Innovative Tech Practice – Open source Geolocation and Maps**
- Judging Innovation (*Example in the Healthcare sector*)
- Sharing Economy
- Business Model Canvas

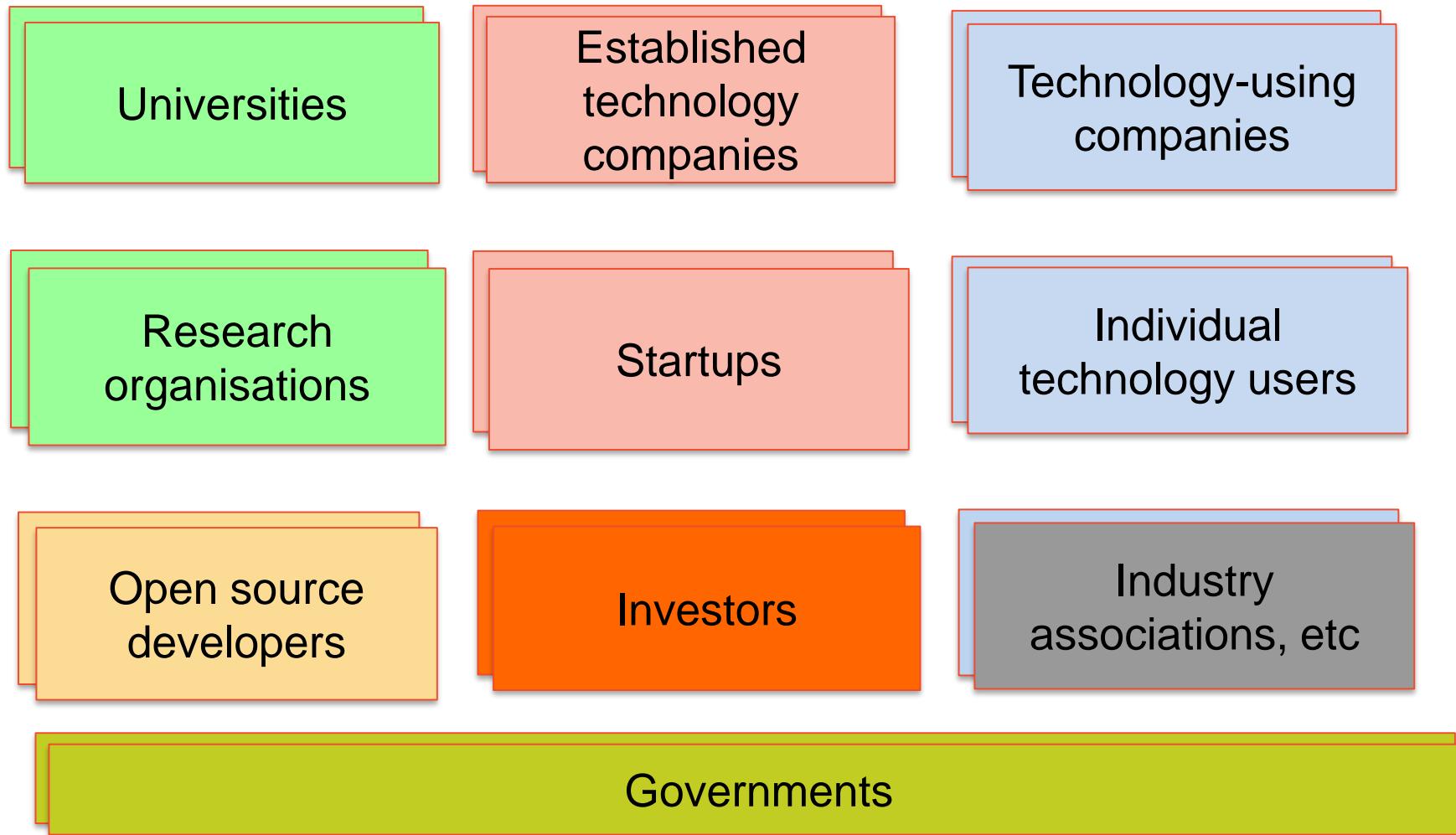


Features:	
Feature Name	Value
Description	{ "type": 0, "captions": [ { "text": "a man swimming in a pool of water", "confidence": 0.7850108693093019 } ] }
Tags	[ { "name": "water", "confidence": 0.9996442794799805 }, { "name": "sport", "confidence": 0.9504992365837097 }, { "name": "swimming", "confidence": 0.906281828803101, "hint": "sport" }, { "name": "pool", "confidence": 0.8787588477134705 }, { "name": "water sport", "confidence": 0.631849467754364, "hint": "sport" } ]
Image Format	jpeg
Image Dimensions	1500 x 1155
Clip Art Type	0 Non-clipart
Line Drawing Type	0 Non-LineDrawing
Black & White Image	False

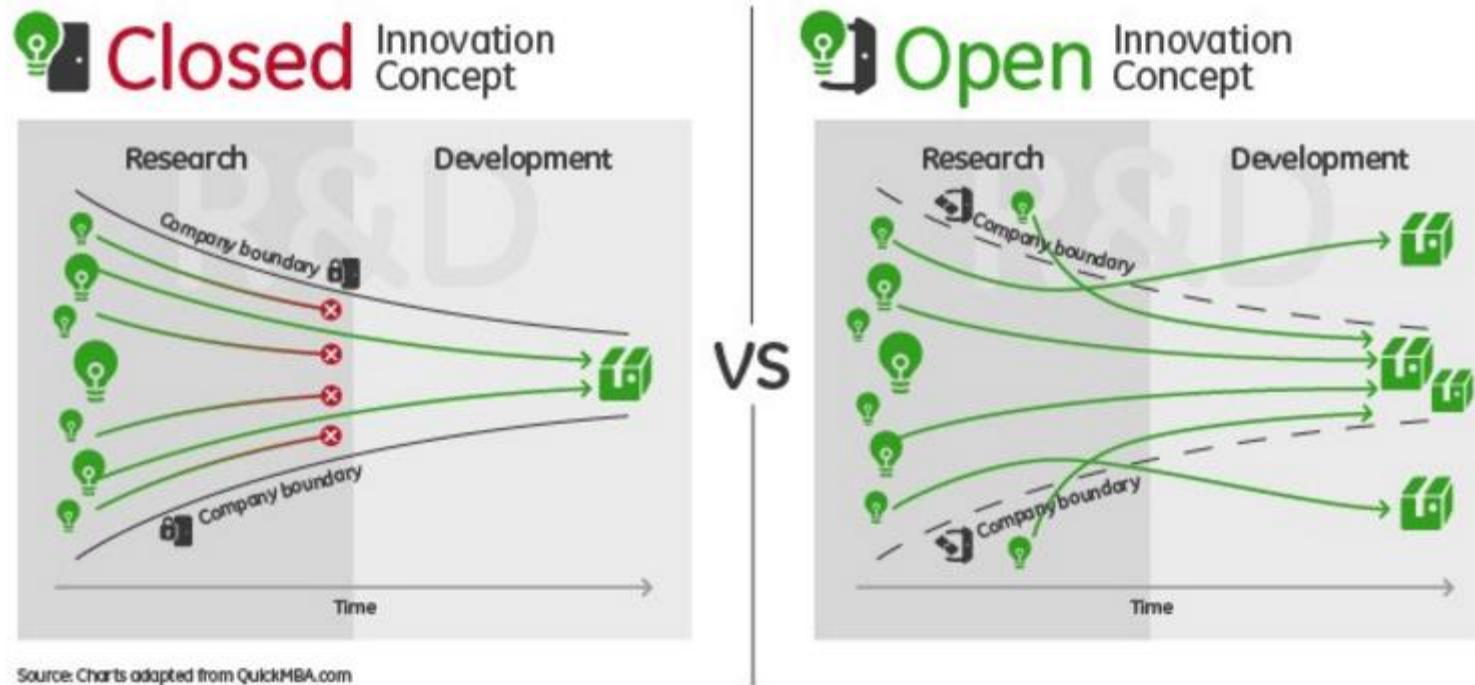


# User Innovation

# Recap week1: IT innovation ecosystem (some key parts)



# Recap week5: The classic innovation funnel: “Closed innovation”



<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

## Traditional model of innovation: “Producer innovation”

- Producer makes product/service for consumers
- Designs for innovations come from producer companies
- Producer innovators profit from many users of the same product/service
- **Assumption that a producer serving many customers can afford to invest more in innovation than a single user innovating for themselves**
- To encourage this investment, typical innovation policy allows producer to “protect” innovation through patents

Source: Baldwin and von Hippel (2011)

## **“User innovation” definition**

- **User innovation** is the idea that more users and consumers than suppliers are the innovators of new products.
- Eric von Hippel was one of the first to notice this trend and explore it.
- Products made by manufacturers (or software companies) are typically developed to meet a wide range of the needs of a wide range of people.
- Therefore, when a particular user experiences needs that are not yet felt by the majority of consumers, they make the adjustments themselves to meet their own needs.
- Often, these ideas are then fed back to the companies from these users in the hope that the product will then be produced for them.
- **These ideas can also lead to new companies being formed, especially with IT products**

# The importance of user innovation: Examples



Eric Von Hippel (MIT)

- Approx 80% of the most important scientific instrument innovations were by users (von Hippel, 1976)
- Many product innovations in sports are innovations by users (von Hippel, 2005)
- Many innovations in IT are innovations by users



[http://www.flickr.com/photos/tz1\\_1zt/112072422/](http://www.flickr.com/photos/tz1_1zt/112072422/)



[connorbaxter.com](http://connorbaxter.com)

# Examples of user innovation in IT

- The World Wide Web
  - created by a worker at a scientific research agency so that the scientists could communicate better
- Many Firefox add-ons
  - Many developers start by developing a plug-in for their own use and then make it available to others
- Apache server modules
  - often originally implemented by a web server administrator
- A lot of open source software is user innovation
  - but not all, many companies also release open source

## Example of user innovation: Apache web server

- In 1994, the most popular web server was “httpd” by Rob McCool at NCSA (same place as Mosaic – most popular web browser at the time)
- This was available as open source
- Many httpd users (webmasters) modified the server code for their own sites
- Rob McCool left NCSA in mid 1994
- Eight httpd users emailed each other to discuss using each others changes
- In 1995, they created a common code base
- By 1996, it was the world’s most used web server
- It still is today



<http://brian.behlendorf.com/>

Brian Behlendorf  
Primary developer of  
Apache

# Example of user innovation: MySQL



<http://dev.mysql.com/doc/refman/5.1/en/history-of-mysql.html>

- We started out with the intention of using the mSQL database system to connect to our tables using our own fast low-level (ISAM) routines. However, after some testing, we came to the conclusion that mSQL was not fast enough or flexible enough for our needs. This resulted in a new SQL interface to our database but with almost the same API interface as mSQL. This API was designed to enable third-party code that was written for use with mSQL to be ported easily for use with MySQL.
- MySQL is named after co-founder Monty Widenius's daughter, My.
- MySQL was bought by Sun Microsystems for \$1 billion in 2008

# Example of user innovation: Yammer



A screenshot of the Geni website showing several features: 1. "Work Together" section with buttons for "Add Brother", "Add Sister", "Add Wife", and "Add Husband". Below it, a "Work Together" description encourages users to invite relatives to collaborate. 2. "Find Your Ancestors" section with a search interface for "First Name" and "Last Name", showing results for "John" Smith and "John" Smith. 3. "Connect to New Relatives" section showing a list of "Thomas's children" with profiles for Robert Whitaker and Robert Whitaker. 4. "Stay Organized" section featuring a photo of a woman and a text box for "Type a narr" (narration). Below these sections are descriptions for each feature.

<https://www.geni.com/>

- Powerful Genealogy Tools to Help You Grow Your Tree
- Find Your Ancestors
- Connect to New Relatives
- ...

## Example of user innovation: Yammer



David Sacks

[http://www.socaltech.com/interview\\_with\\_david\\_sacks\\_geni\\_and\\_yammer/s-0017613.html](http://www.socaltech.com/interview_with_david_sacks_geni_and_yammer/s-0017613.html)

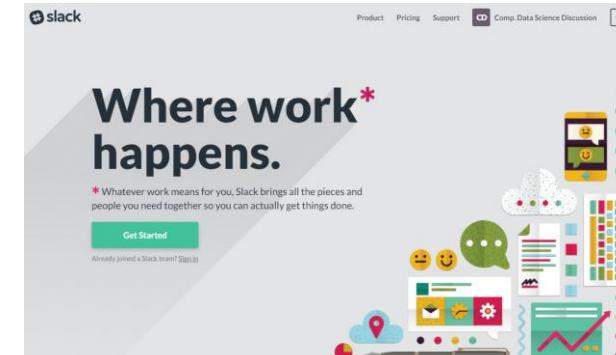
<https://www.yammer.com/>

- Yammer started as an internal productivity tool at Geni. We basically built the tool to help people stay connected, and we've been using it internally for six months.
- We have about 30 employees at Geni, and have about 20,000 messages on yammer.
- It's been incredibly successful at Geni, and is the center of the company's culture. We recently decided we should spin it out into a separate company, so that other companies can use the product as well. About a month ago, we spun it out and premiered it at TechCrunch50, as you know, and won that event.
- Yammer was bought by Microsoft for \$1.2 billion in 2012
- Yammer is now used by more than 200,000 companies (source: yammer.com)

# Example of user innovation: Slack



Glitch is Dead, Long Live Glitch!  
Art & Code from the Game Released into Public Domain  
<https://www.glitchthegame.com/>



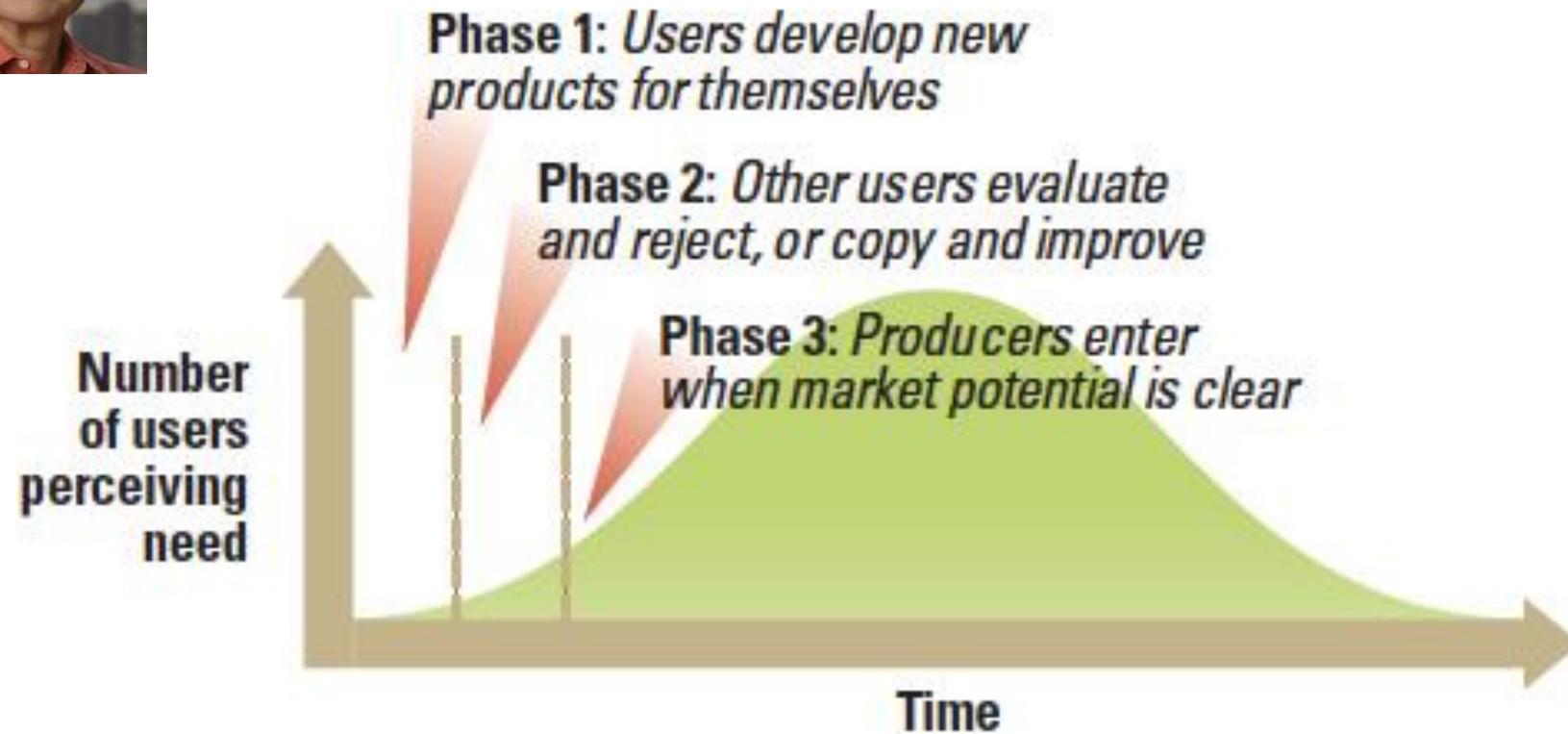
<https://slack.com/>

- Glitch was a browser-based massively multiplayer online game
- Created by Tiny Speck (co-founded by Flickr co-founder Stewart Butterfield)
- Glitch launched Sept 2011
- Slack was developed as an internal tool for team communication
- Glitch shut down Dec 2012
- Company focused on Slack as a product for others
- Have received \$540M in investment for Slack

# User Innovation



Eric Von Hippel  
MIT Sloan School of Management

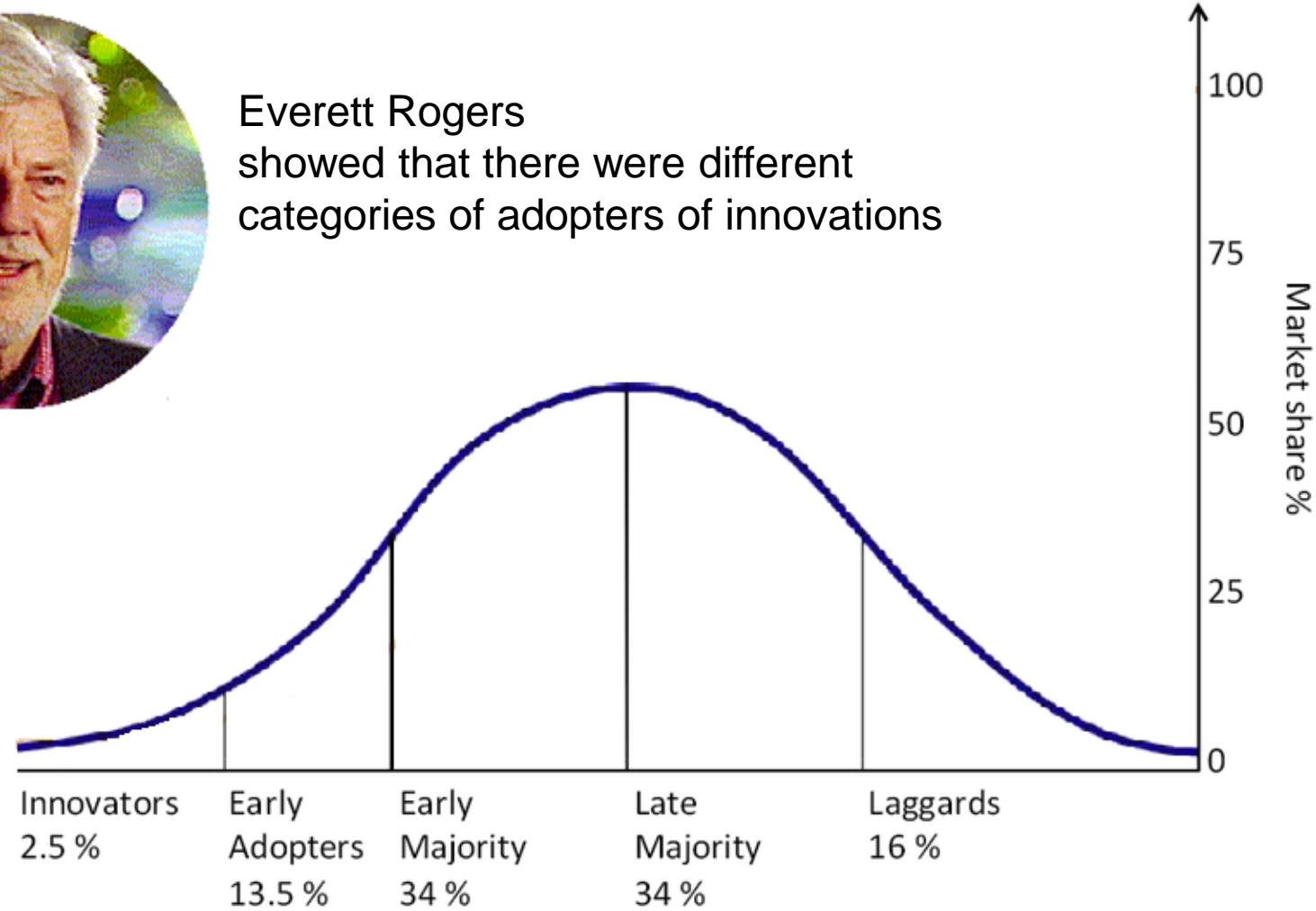


<http://sloanreview.mit.edu/article/the-user-innovation-revolution/>

## Recap week 3: Diffusion of Innovation



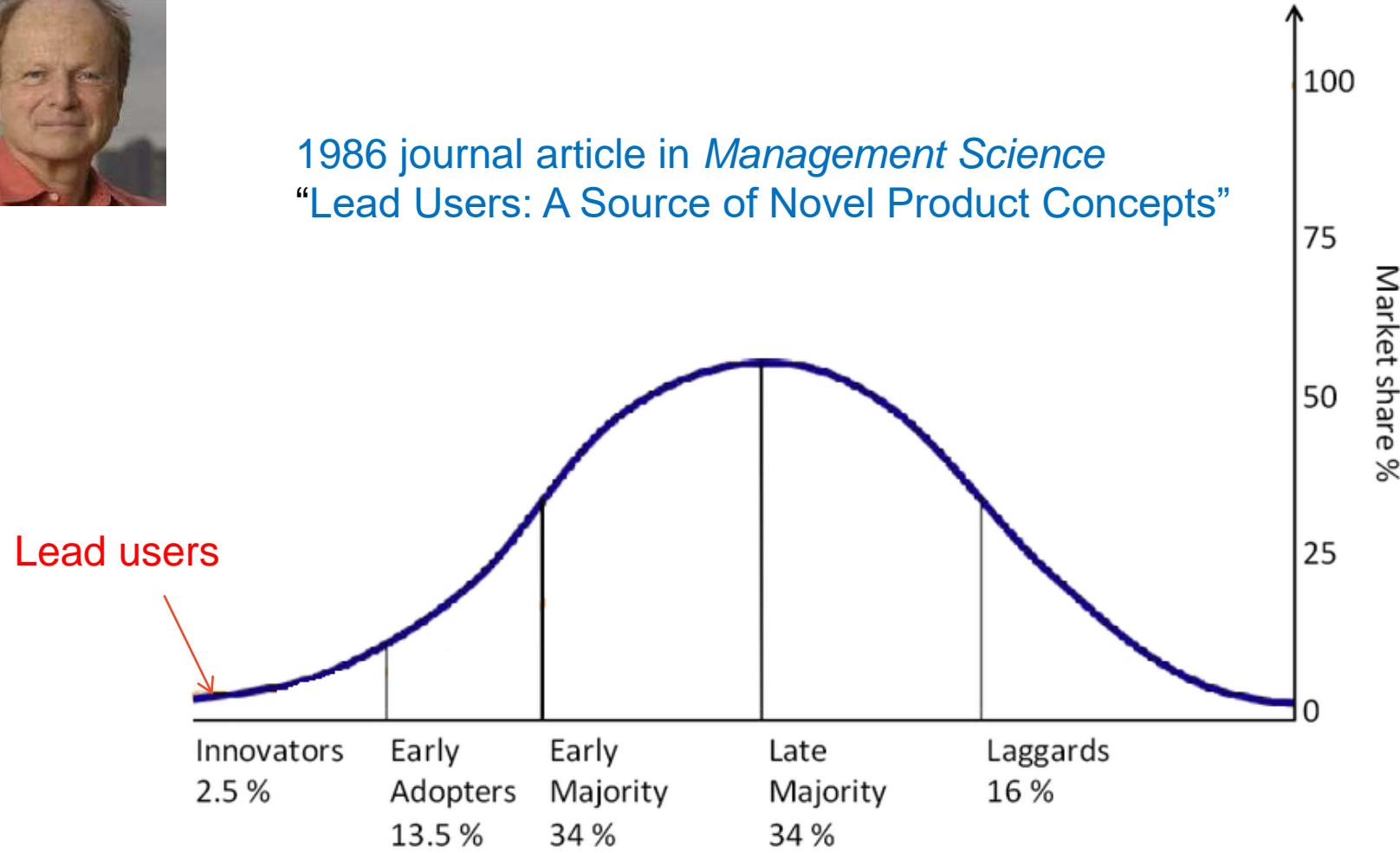
Everett Rogers showed that there were different categories of adopters of innovations



# Lead users

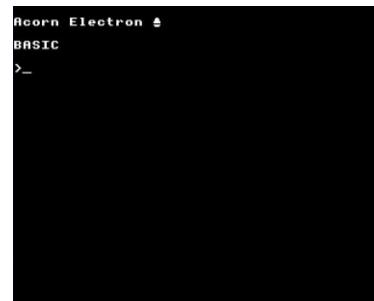


1986 journal article in *Management Science*  
“Lead Users: A Source of Novel Product Concepts”

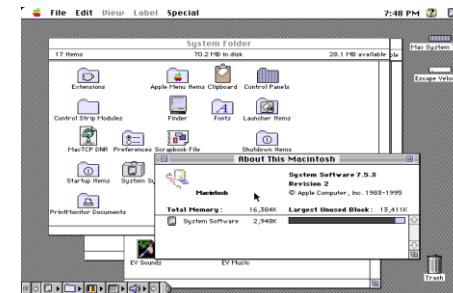


# Lead users

- In some product categories (eg cleaning products), market research focuses on typical users (eg with interviews, focus groups)
- The feedback and opinions of typical users can be useful in developing new products.
- For IT and other high tech industries, typical users are not so effective
  - Eg they often suffer from “**functional fixedness**”



Typical users not likely to suggest



- Involving lead users often leads to more effective innovation
- Lead users may be individuals, companies or communities

# How to identify lead users

- According to Von Hippel...
- Lead Users:
  - Face the needs that will be general in the market, but months or years before the general marketplace realise the needs.
  - Will benefit significantly by obtaining a solution to those needs, and...
  - Spend resources trying to solve those needs
  - Are at the leading edge of trends and so are very knowledgeable about “state of the art”
  - Note: Lead users are not usually a company’s “lead customers” – they are usually not satisfied with current products so have had to create their own



## **Users as source of innovation**

- “Lead users” can provide concepts for products, services, processes and features to help companies innovate
- Sometimes, lead users actually do the innovation themselves (i.e. “user innovation”)

# User-led Innovation – an alternative perspective

- **Users insights can't predict future demand:** The users themselves often have no idea if they will like a breakthrough product before they start using it.
- **User-centered processes stifles creativity:** “The user-centered process is created as linear rational process for innovation and that’s why it’s so popular among managers... creating something new is a chaotic, unpredictable, frustrating, and very, very hard process. And most of all, it’s the result of extraordinary efforts and visions of a few extremely talented people.”
- **User focus makes companies miss out on disruptive innovations:** “Focusing on users will lead companies to make incremental innovations that typically tend to make the products more expensive and complicated and ironically, in the long run, less competitive.”
- **User-led design leads to sameness:** “Even if user insights were useful, it isn’t a competitive advantage. Even the most advanced users studies are now widely available.”

<http://www.forbes.com/sites/stevedenning/2011/02/15/user-led-innovation-cant-create-breakthroughs/#35edb5c75a9b>  
<http://www.fastcodesign.com/1663220/user-led-innovation-cant-create-breakthroughs-just-ask-apple-and-ikea>

## User-led Innovation – Apple?

- Apple has a good track record when it comes to its consumer products, and that's because of how they define "insanely great products." To quote Jobs again:

*"It's not about pop culture, and it's not about fooling people, and it's not about convincing people that they want something they don't. We figure out what we want. And I think we're pretty good at having the right discipline to think through whether a lot of other people are going to want it, too. That's what we get paid to do."*
- In other words, Apple makes products that they themselves want to use. They are their own leading-edge customers.

<https://hbr.org/2012/12/to-stay-ahead-of-disruptions-curve>

# User innovation: becoming an even bigger force in innovation – eg “Maker movement”



Image source: <http://spotlight.macfound.org/blog/entry/craftsmanship-is-dead-long-live-maker-culture/>



Image source: <http://robodino.org/>

*“The maker movement, as we know, is the umbrella term for independent inventors, designers and tinkerers”*

<http://time.com/104210/maker-faire-maker-movement/> 2014

# **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



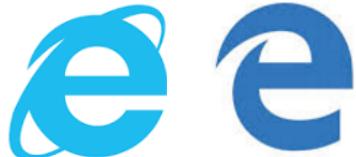
SAP ERP



IBM DB2



Oracle Database



Internet Explorer

(Many of these also use open source within them)

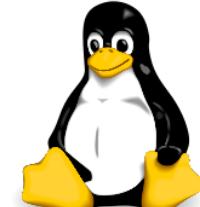
Free and open source software



Android



Chrome



Linux



Firefox



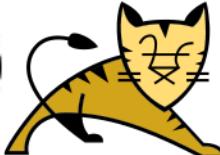
Apache  
HTTP Server



Alfresco



Hadoop



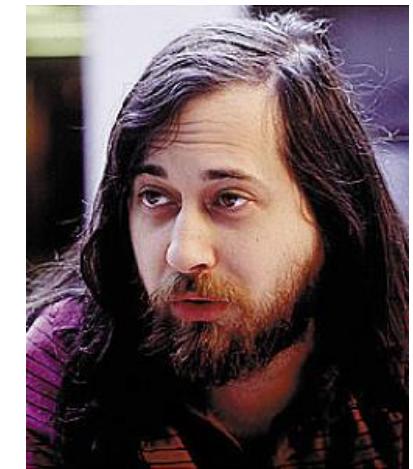
Tomcat



(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 (GNU's Not Unix)
  - 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 (eg 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.

The screenshot shows the GitHub repository page for 'twitter/bootstrap'. At the top, there's a navigation bar with links for Explore GitHub, Search, Features, and Blog. On the right, there are buttons for Sign up for free and Sign in. Below the navigation, the repository name 'twitter/bootstrap' is displayed, along with statistics: 48,009 stars, 14,257 forks, and 156 issues. A tabs menu includes Code (which is selected), Network, Pull Requests (68), Issues (156), Wiki, and Graphs. Below the tabs, a brief description states: 'Sleek, intuitive, and powerful front-end framework for faster and easier web development.' with a 'Read more' link. It also shows the URL <http://twitter.github.com/bootstrap>. Under the 'Code' tab, there are buttons for Clone In Mac, ZIP, HTTP, SSH, Git Read-Only, and the repository URL. A note indicates 'Read-Only access'. Below these are buttons for branch: master, Files, Commits, Branches (5), and Tags (18). A prominent section titled 'bootstrap / +' shows a merge pull request from waynn/patch-1. It lists several commits by mdo, including changes to docs, img, js, less, .gitignore, and .travis.yml. The latest commit is dated 23 days ago and is authored by mdo. The commit message for .travis.yml mentions making a few changes to package.json. The entire interface is clean and modern, typical of GitHub's design.

# **Importance of FOSS for innovation**

# Importance of free and open source software

## Relevance to IT careers

### A. Enterprise IT

Understand IT innovations so you can analyse likely impact from new technologies and plan for their adoption

### B. R&D of IT technologies

Understand IT innovations so you can lead the development of new technologies within an established organisation

### C. IT start-up

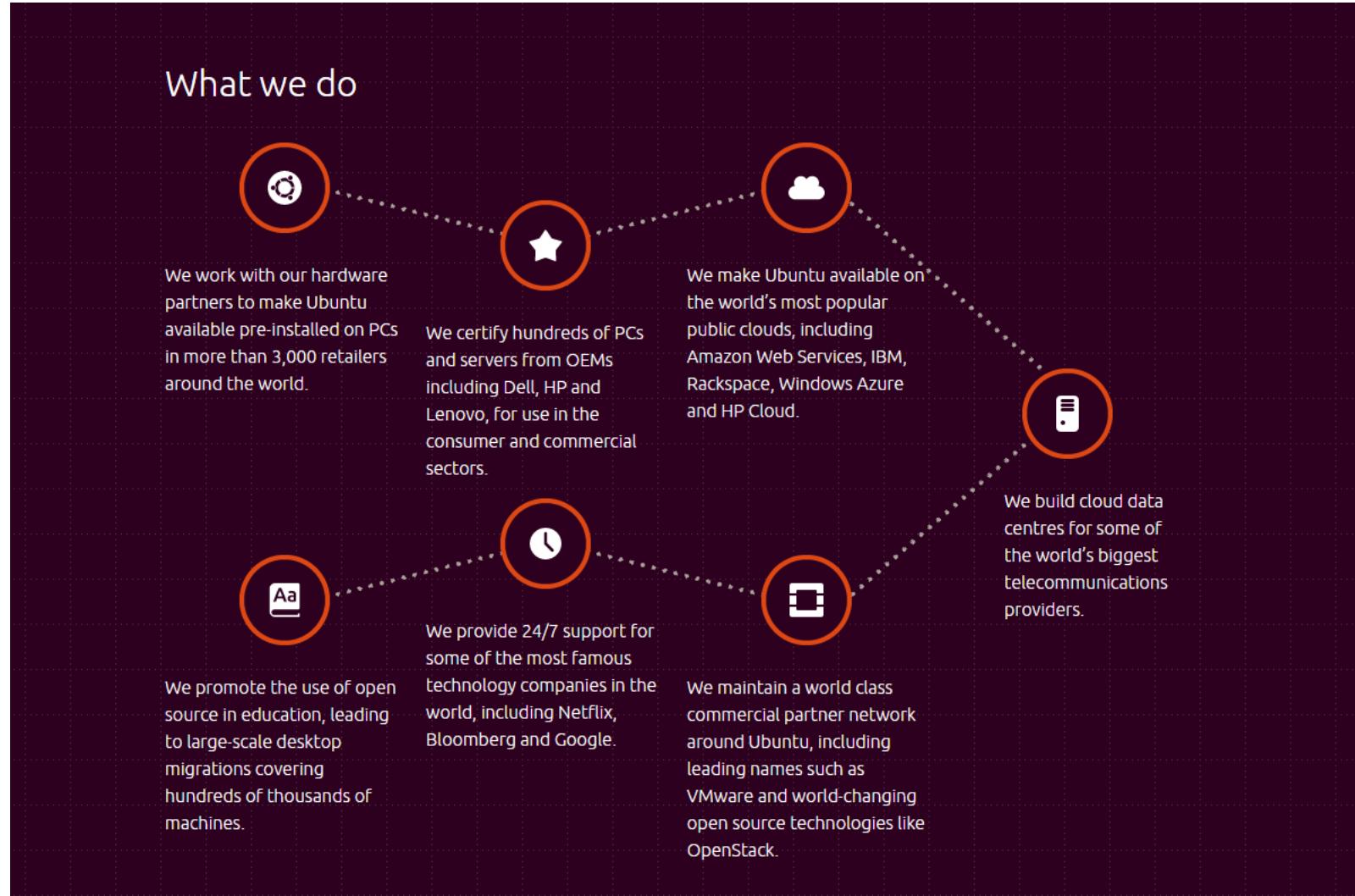
Understand IT innovation so that you can create a software start-up company

## 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

# Example: Canonical

- It is our mission to make open source software available to people everywhere. We believe the best way to fuel innovation is to give the innovators the technology they need.*

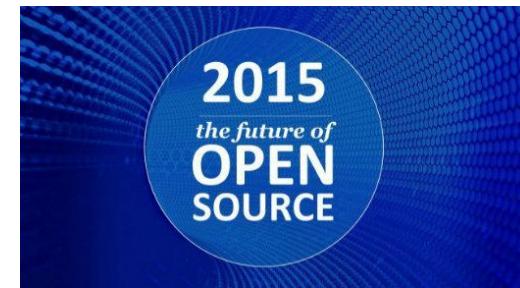


<http://www.canonical.com/>

# Open Source usage in Enterprises

- Survey by North Bridge and Black Duck Software
- Input from 1300 senior IT professionals
- 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](http://www.slideshare.net/North_Bridge/2015-future-of-open-source-study)



# Importance of free and open source software

## Relevance to IT careers

### A. Enterprise IT

Understand IT innovations so you can analyse likely impact from new technologies and plan for their adoption

### B. R&D of IT technologies

Understand IT innovations so you can lead the development of new technologies within an established organisation

### C. IT start-up

Understand IT innovation so that you can create a software start-up company

# Open source for innovation



[http://www.slideshare.net/North\\_Bridge/2015-future-of-open-source-study](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 (eg Linux)
  - Containers (eg Docker)
  - System configuration management (eg Puppet, Chef)
- Most new software is built using FOSS:
  - Software platforms (eg Java, Scala, Python, Ruby on Rails, node.js)
  - Software libraries/frameworks (eg Spring framework, glibc)
  - Software build and test automation (eg 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)
- **Cloud service using mostly open source**
  - **Examples: Facebook, Twitter**
- **Other advantage 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”
  - Eg a programmer introduces some GPL 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 final product if some 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](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

Public  
domain

MIT

BSD

Apache  
Software  
License

Restrictive (copyleft) licences:  
Changes must be made available

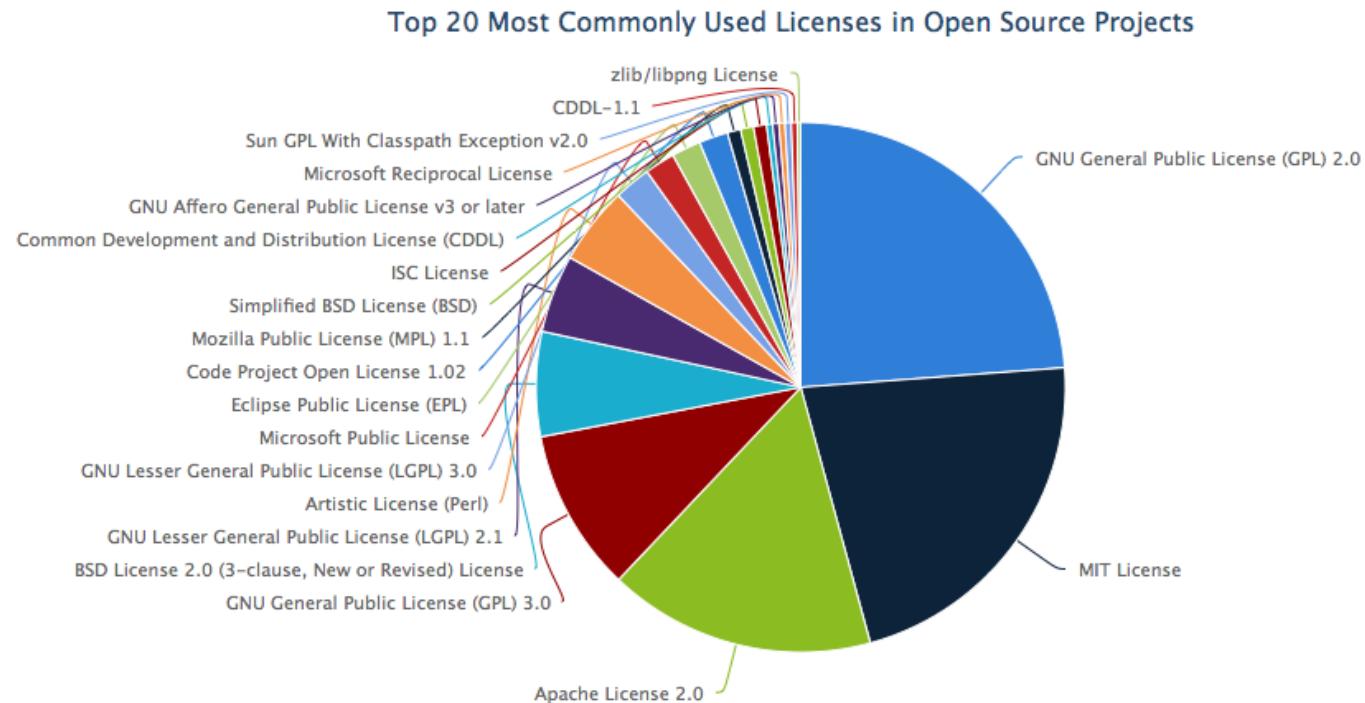
GPLv2    GPLv3    AGPL    SleepyCat



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

# 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



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

# 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)



***Free as in Freedom***

- 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

Public  
domain

MIT

BSD

Apache  
Software  
License

Restrictive (copyleft) licences:

Changes must be made available

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>

# Summary

- Three forms of innovation:
  - Producer innovation
  - User innovation (growing substantially in IT)
  - Open collaborative innovation (growing substantially in IT)
- User innovation important part of IT innovation and growing
- Many innovative companies watch lead users and user-innovators
- 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*, 4<sup>th</sup> 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/democ1.htm>

# Case Study

## Maps, Open Source and User Innovation



[https://commons.wikimedia.org/wiki/File:World\\_Map\\_1689.JPG](https://commons.wikimedia.org/wiki/File:World_Map_1689.JPG)

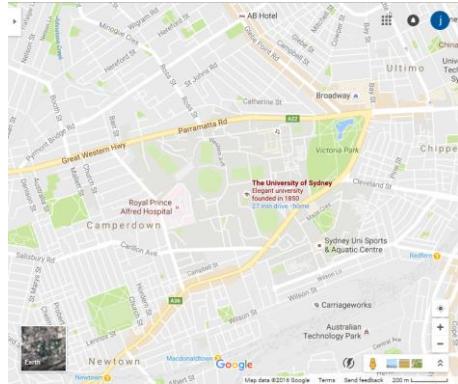
# Geographic Information System

- 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>

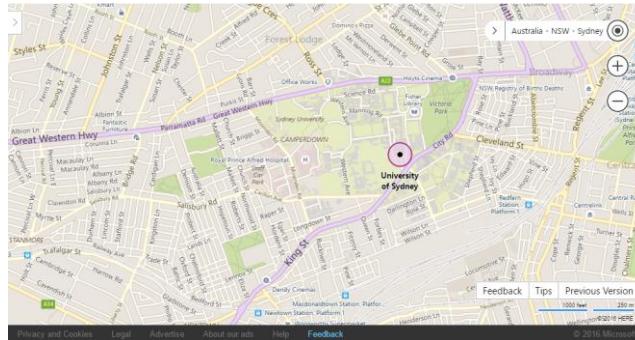
# Maps?

- Google Maps



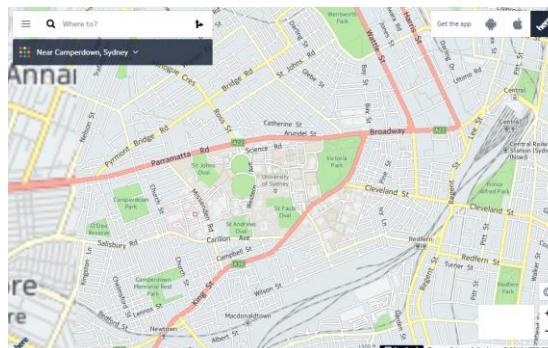
<https://www.google.com/maps/>

- Bing Maps



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

- Here WeGo Map



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

X Tracks > Track file No. 27583

Track uploader: ToeBee

Location: 38.90116 | -101.75262

Info

Track uploader:	ToeBee
Date of recording:	Nov 18, 2016
Track no.:	27583
Platform:	Unknown
Total no. of images:	1 475
OBD:	No

14 715 Points

Coverage	Photos	Distance	Points
0	1 468	34.46 km	14 680
1	7	0.07 km	35

1 / 1475

Load full resolution photos

Main Street

Wallace County High School

40

+

-

← →

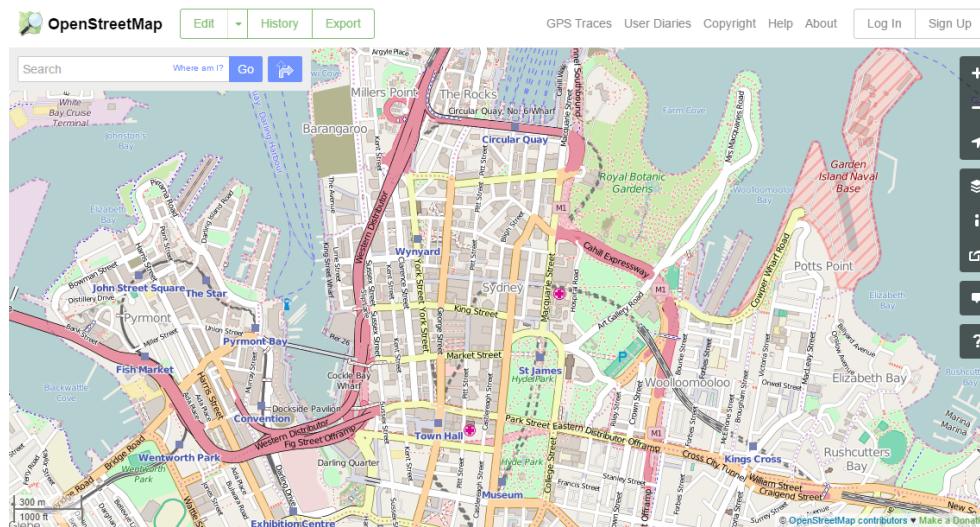
# An open source map

## — Open Street Map



Steve Coast

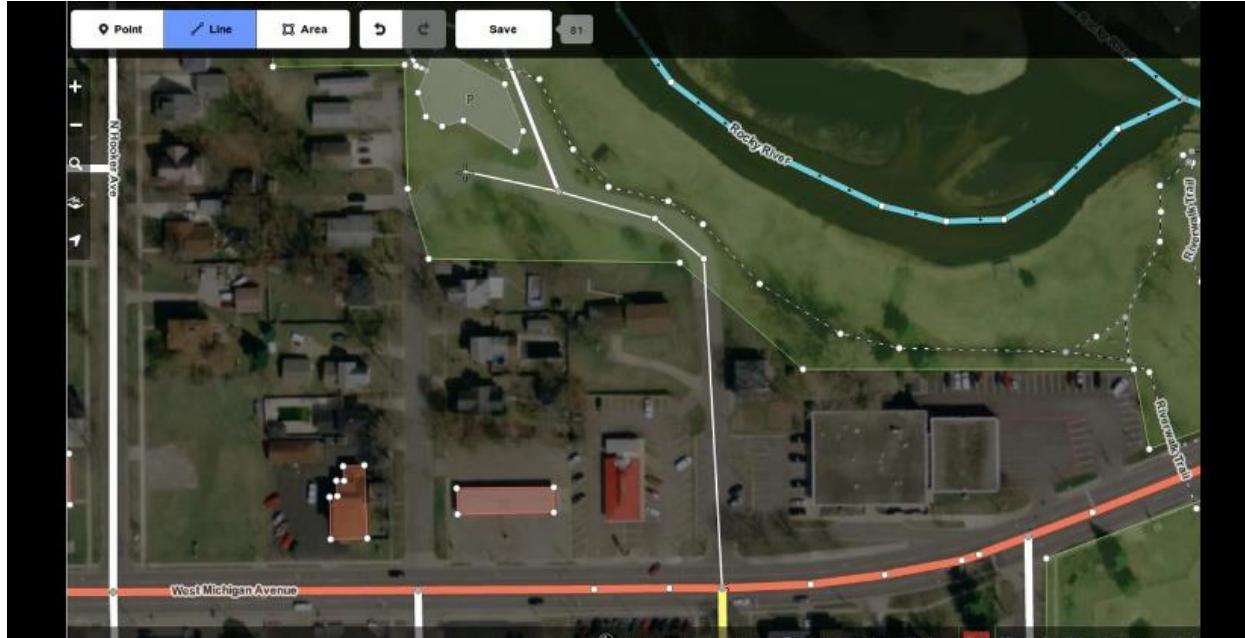
<http://www.allthepeople.net/steve-coast>



<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](http://ieeexplore.ieee.org)

# Mapping is easier than you think!



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

# 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](http://wiki.osmfoundation.org/wiki/Main_Page)

# OpenStreetMap

- Open source – enable innovation 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!

## 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/>

# **Tute 4**

## Maps and Distributed Innovation

# Maps

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

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

CSS

JAVASCRIPT



1 of 3

TEAM MAGENTA

What is the **Distributed Innovation Concept(s)** 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

- 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

Enter Answer Here

**SUBMIT ANSWER**

# Group Presentation

Group	Preference 1	Preference 2	Preference 3	Remarks	Preference	Presentation Order (week)
Groups 3	Autonomous Vehicle	3D/4D Printing	Virtual Assistant		2	13th (week 9)
Groups 4	Human Augmentation /implantable sensors	Quantum Computing	Big Data		3	20th (week 9)
Groups 5	Sharing Economy	Multi-modal Interaction: Gesture/Speech/Brain Control	Block Chains		1	8th (week 8)
Groups 6	Commercial Drones /Autonomous Driving	Smart Home	Sharing Economy		1	3rd (week 7)
Groups 7	Autonomous Vehicle	Smart Home	Personal Analytics		3	21th (week 9)
Groups 8	Augmented/Virtual/Mixed Reality	Smart Home	Quantum Computing		3	19th (week 9)
Groups 9	Commercial Drones /Autonomous Driving	Cognitive Services	Sharing Economy	Submitted by two students; using the 1st one	2	18th (week 9)
Groups 10	IoT Platforms	Smart Home	Virtual Assistant		3	15th (week 9)
Groups 11	IoT Platforms	3D/4D Printing	Sharing Economy		2	9th (week 8)
Groups 12	Autonomous Vehicle	3D/4D Printing	Block Chains		3	16th (week 9)
Groups 13	IoT Platforms	Augmented Reality	Sharing Economy		1	14th (week 9)
Groups 14				Missing submission	0	1st (week 7)
Groups 15	Virtual Reality (VR)	3D/4D Printing	Sharing Economy		3	17th (week 9)
Groups 16	Commercial Drones /Autonomous Driving	Smart Home	Human Augmentation /Implantable sensors		2	6th (week 8)
Groups 17	Commercial Drones /Autonomous Driving	Augmented/Virtual/Mixed Reality	Smart Home		1	4th (week 8)
Groups 18	Virtual Reality (VR)	3D/4D Printing	Block Chains		1	5th (week 8)
Groups 19	Human Augmentation /Implantable sensors	Smart Home	Virtual Assistant		1	10th (week 8)
Groups 20	Commercial Drones /Autonomous Driving	Multi-modal Interaction: Gesture/Speech/Brain Control	Big Data		2	11th (week 8)
Groups 21	Commercial Drones /Autonomous Driving	Cognitive Services	Human Augmentation /Implantable sensors		2	12th (week 8)
Groups 22	Augmented/Virtual/Mixed Reality	3D/4D Printing	Smart Home		1	2nd (week 7)
Groups 23	IoT Platforms	Smart Home	Virtual Assistant		1	7th (week 8)

Average Preference	1.80952381
Number of Groups	21

Selected Topics	Count
IoT Platforms	4
Commercial Drones /Autono	9
Autonomous Vehicle	N/A
Augmented/Virtual/Mixed Re	7
Augmented Reality	N/A
Virtual Reality (VR)	N/A
Human Augmentation/implant	4
Sharing Economy	7
Smart Home	9
3D/4D Printing	7
Multi-modal Interaction: Ges	2
Cognitive Services	2
Quantum Computing	2
Virtual Assistant	4
Block Chains	3
Big Data	2
Personal Analytics	1

- Topics now available – come up with a presentation Title!
- Happy to discuss with your group on an interesting industry / case studies
- Know your group's week and order – come prepared