

Open Source Software

Lecture I Introduction to Open Source Software

Prof. Josué Obregón

Department of Industrial Engineering - ITM

Seoul National University of Science and Technology



Agenda

- Lecturer and students
- Course logistics
 - Objectives
 - Methodology
 - Contents
 - Resources
- Introduction to OSS
 - What is and why use OSS?

Lecturer presentation



- BSc in Computer Engineering,
 - Universidad de San Carlos de Guatemala



- MSc in Industrial & Management Systems Engineering
 - Business Process Management Laboratory
 - Process mining applied to SNS



- PhD in Industrial & Management Systems Engineering
 - Industrial Artificial Intelligence Laboratory
 - Interpretable Machine Learning for tree ensembles

Prof. Josue Obregon (Assistant Professor)

Tel: 02-970-7291

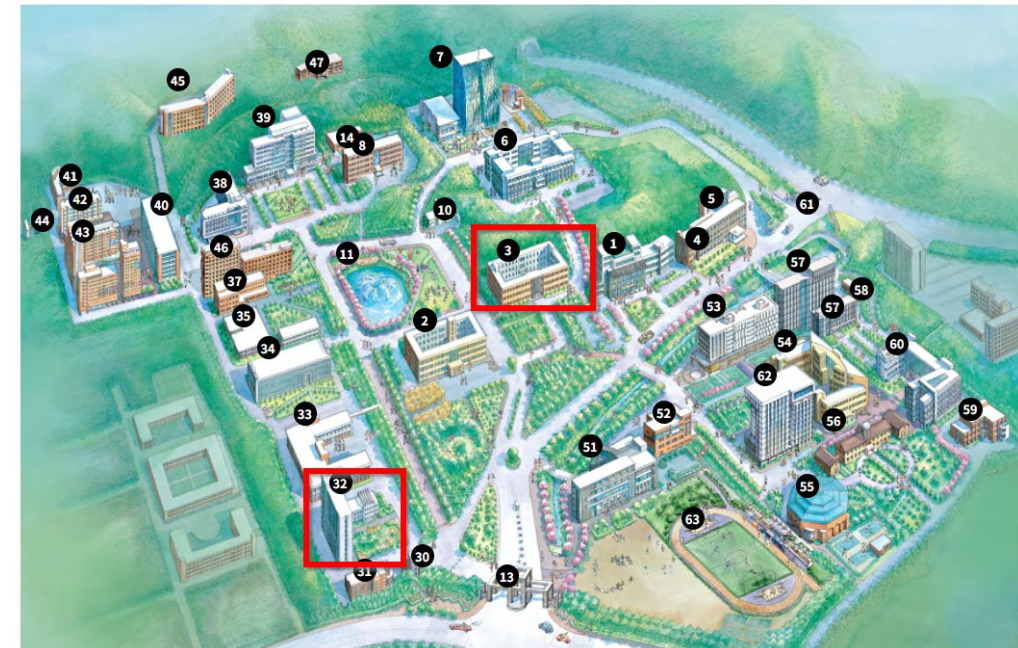
Office: Changhak Hall (3), Room 334-1

Office Hours: Monday 14:00 – 17:00

Home-page: <https://www.josue-obregon.com/>

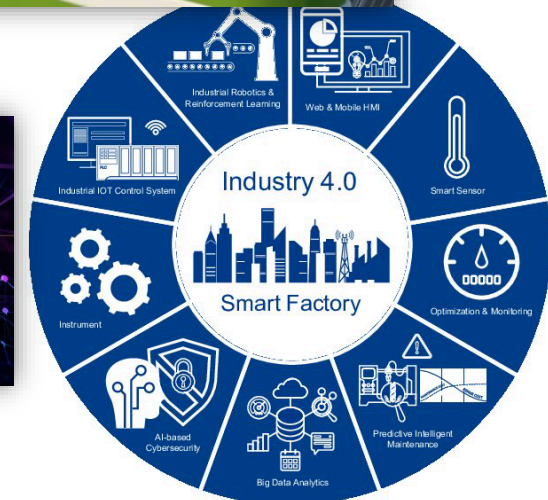
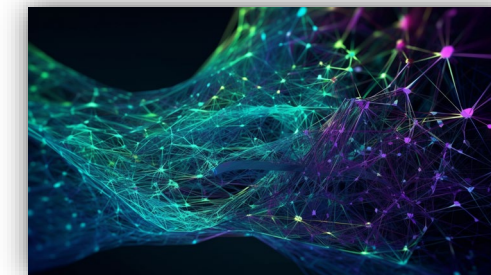
Email: jobregon@seoultech.ac.kr

Publications: [Google Scholar Profile](#)



Lecturer presentation

- EIS Lab (Explainable Intelligent Systems Lab)
- Applied Machine Learning / Deep Learning
 - Smart Energy
 - Solar energy power generation forecasting
 - Batteries SOH estimation
 - Smart Manufacturing
 - Manufacturing quality prediction
- Explainable Artificial Intelligence (XAI)
 - How can we explain the predictions of complex models??
- If you are:
 - Curious about applied AI
 - Want to learn how to read a paper and do basic research
 - Contact me!



https://www.researchgate.net/publication/333531793_Industrial_Internet_of_Things_Big_Data_and_Artificial_Intelligence_in_the_Smart_Factory_a_survey_and_perspective/figures?lo=1&utm_source=google&utm_medium=organic

<https://smartenergysystems.eu/about/>

<https://www.udacity.com/course/deep-learning-pytorch--ud188>

Students

- What do you think the term "open source" means?
- Have you ever downloaded or used software that was free?
- Why do you think some developers would want to share their code with the world for free?
- Do you think open source software is more secure, less secure, or the same compared to software that isn't open-source? Why?



Course objectives

1. Understand the principles and philosophies of open source software.
2. Gain practical experience with open source tools and communities.
3. Familiarize with the contribution process as well as develop skills to contribute to open source projects.
4. Learn project management and collaboration techniques in an open source context.
5. Experiment with several open source tools widely used in the world

Class methodology

- There are two class components: Lectures and hand-out practices to learn OSS concepts, infrastructure or tools
 - Lectures: Tuesday 10:00 – 13:00 (Room 501 Frontier Building)
 - Office Hours: Monday 14:00 – 17:00 (Room 334-I, Changhak Hall (3))

Communication



During office hours

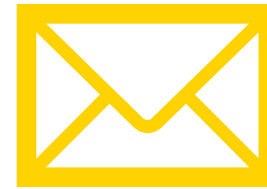
No appointment is required!

First-come, first-serve

In-person or Zoom

Be effective, more people may be waiting for their turn

Have your materials ready



Outside office hours

No walk-in

Appointments can be requested by email preferably

Be aware!

“After class” is not my office hours

- Ask your questions during class time (including attendance issues)
- If your questions are not answered in class, please come to office hours or use asynchronous communication!

Academic Integrity Violation

SeoulTech takes it seriously, so do I

You may get away with it for some time, but not forever

You would be surprised by how easily we can spot it

Both cheaters and facilitators are penalized

In-class participation

1

Talk to me

- Raise your hand and wait your turn to talk
- It is expected and **appreciated**.

2

Ask questions

- It doesn't show a weakness
- It helps everyone to grow!

3

Answer questions

4

Correct some mistakes I will eventually make

- Yeah, this happens

HW start the day they are assigned...

... not the day they are due!

I will not answer questions at night

I will not answer questions on weekends

I cannot help you 2 hours before the deadline

I cannot help you after you submitted the wrong file/answer

Evaluation

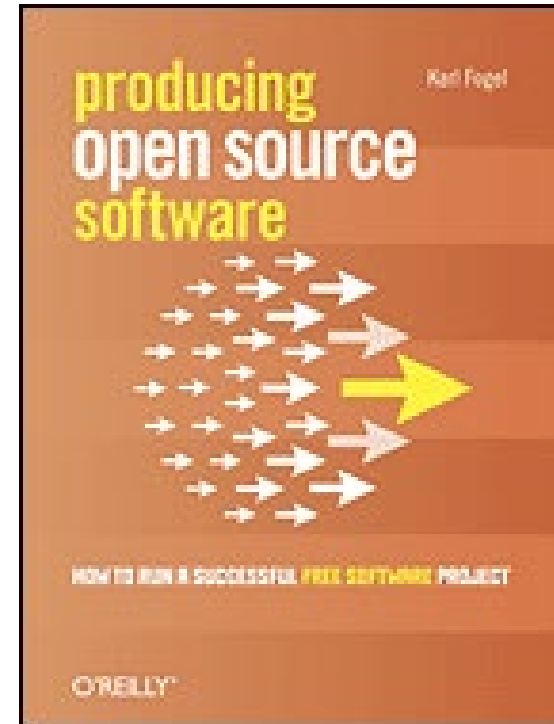
- Evaluation
 - Homework: 20% (4 or 5)
 - Team Project: 50% (divided in two parts)
 - Midterm exam: 30%

Contents

- Introduction to Opensource Software (W1) (HW1)
- Creating an OSS project (W2) (Recorded Lecture)
- Version Control with Git and GitHub I (W3-W4) (HW2)
- Code Review (W5) (HW3)
- Packaging releasing and Daily Development (W6)
- How to Contribute to OSS (W7) (Project Statement Release)
- Midterm Exam (W8)
- Social and Political Infrastructure of OSS Projects (W9)
- Commercial OSS (W10)
- Case Study 1: Linux (W11) (HW 4?)
- Mid-term Project Presentation (W12)
- Case Study 2: Wikipedia (W13) (HW 5)
- TBD (W14)
- Final Project Presentation (W15)

Readings and resources

- Course material posted on e-class system
 - <http://eclass.seoultech.ac.kr>
- Textbook
 - Fogel, Karl: Producing Open Source Software (Second Edition)
 - <https://producingoss.com/>

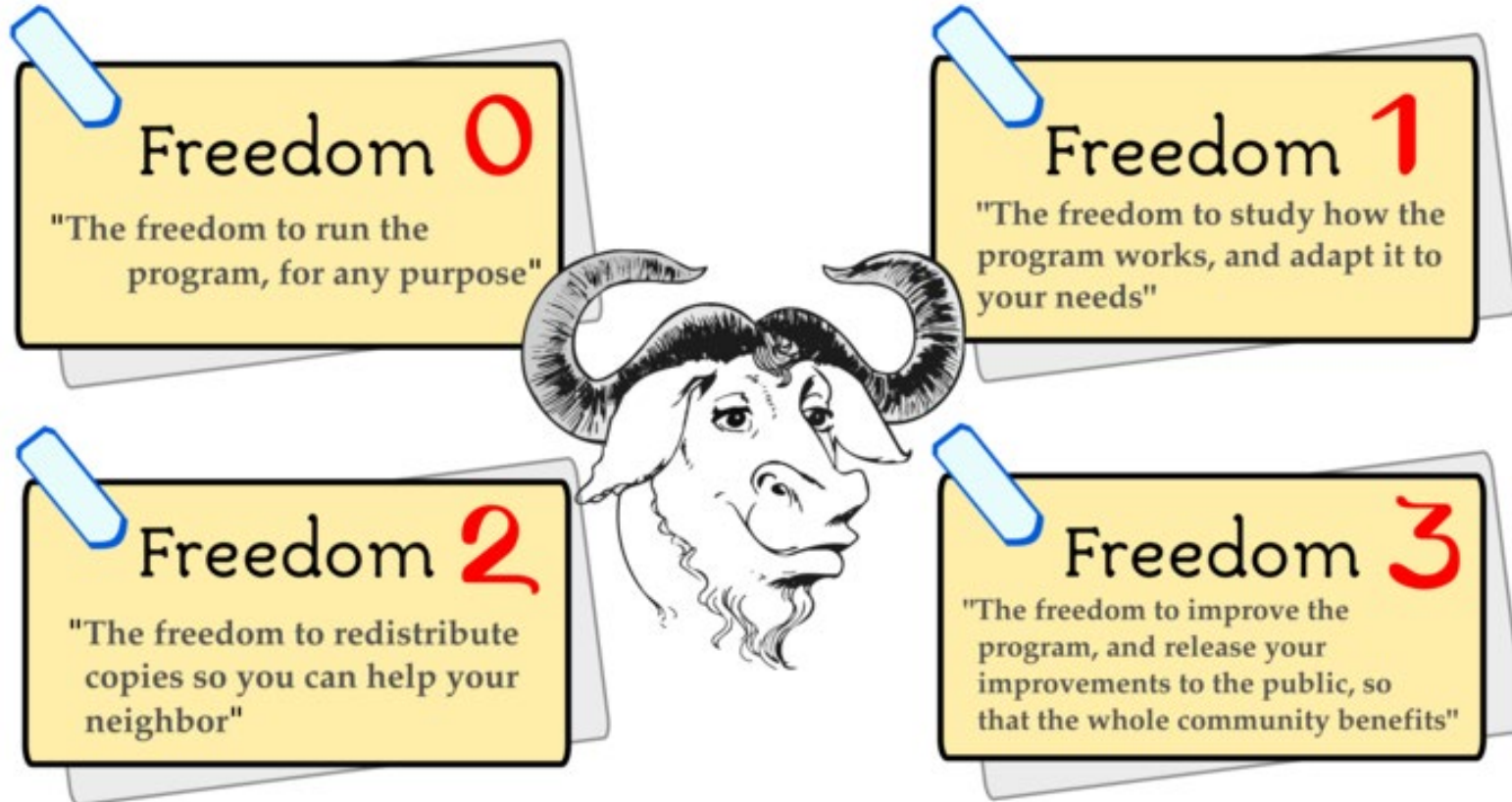




Open Source or Free Software

Software that ensures that the users have...

The Four Freedoms



written by Richard Stallman, founded the Free Software Foundation (FSF) and launched the GNU Project.

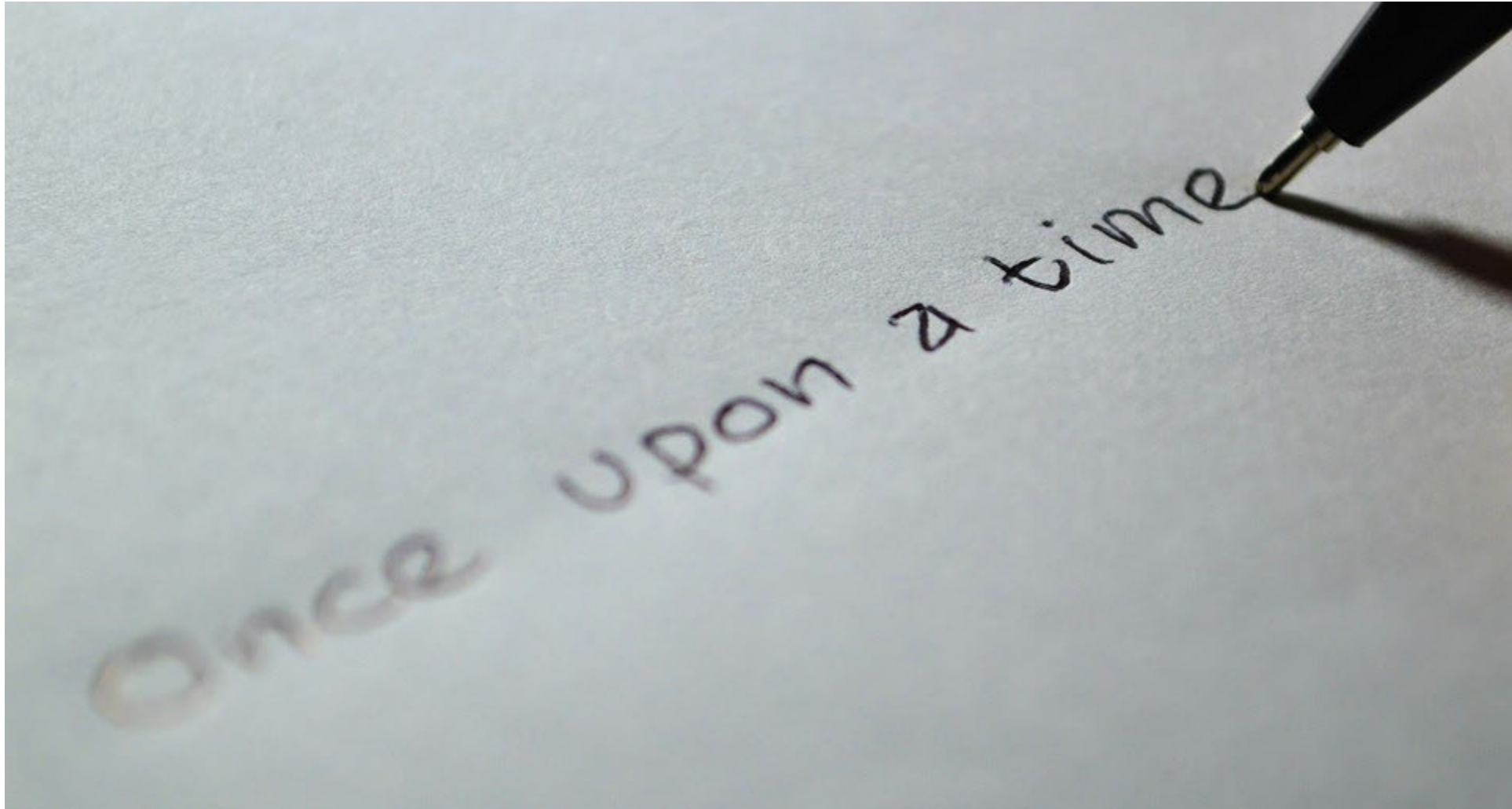
Proprietary (Closed Source) Software

- Only the owners of such software projects have full legal access to the source code involved.
- For this reason, this model is also called closed source.
- To use proprietary software, end users must accept a license restricting their rights
 - Often through a confusing click-through box presented after a very long and dense description during install → such licenses usually violate the four freedoms

Price is not the point

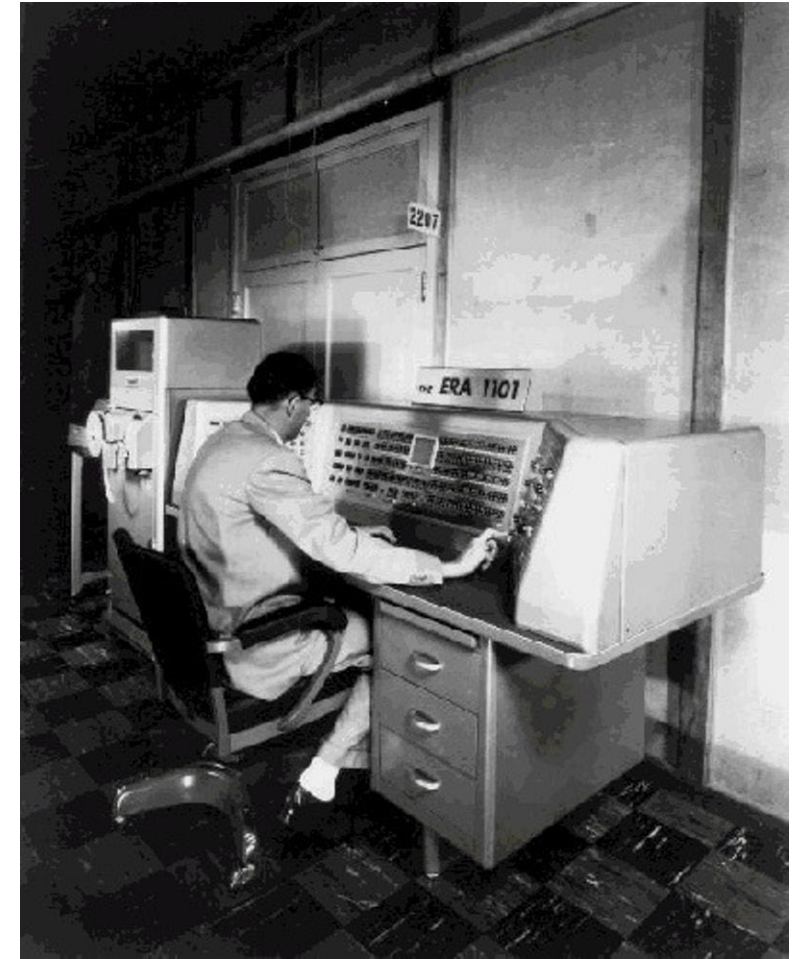
The difference with OSS and proprietary models has nothing to do with price. One can charge as much as one wants for an open product and one can provide a proprietary product for free. The license differences have to do with redistribution, modification, reuse of code, etc.

A little bit of history



The 1950s and Earlier...

- Software arose from researchers, both academic and corporate
- Distributed openly and cooperatively
- Source always distributed, binaries less often
- Software not seen as separate commodity
- Software bundled for free with hardware
- Licensing was sloppy



The 1960s

- Major aspects of computer science and software rapidly developed both in academia (MIT, UC Berkeley) and industrial research labs (Bell Labs, Xerox)
 - 1964: BASIC high-level programming language was developed at Bell Labs
- 1968: ARPANET emerges, eventually leads to Internet, essential for developers and researchers to communicate, share, collaborate
- 1969: UNIX born at Bell Labs (AT&T) and given for free to universities and research centers
- 1969: IBM forced by US government to break software and hardware apart and sell/distribute separately, due to unfair business practices.



The 1970s

- 1972: C programming language is developed at Bell Labs by Dennis Ritchie
- 1976: emacs released by Richard Stallman and Guy Steele. Versions by other authors proliferate, including proprietary ones. GNU emacs not released until 1985
- 1978: First version of TeX released by Donald Knuth, an open source typesetting system often used for publishing journal articles and books, still in widespread use, usually in the LaTeX version.



Dennis Ritchie stands over Ken Thompson as he works on in 1972 at Bell Labs.

The 1980s (I)

- 1980: Usenet begins as the ancestor of user forums and the World Wide Web
- 1980: Stallman problem with Xerox printer
 - Stallman refused access to the source code for the printer driver
- 1982: GNU project announced by Richard Stallman

We did not call our software "free software", because that term did not yet exist; but that is what it was. Whenever people from another university or a company wanted to port and use a program, we gladly let them. If you saw someone using an unfamiliar and interesting program, you could always ask to see the source code, so that you could read it, change it, or cannibalize parts of it to make a new program.

(from <https://www.gnu.org/gnu/thegnuproject.html>)



Stallman and the GNU mascot

The 1980s (II)

- 1984: X Window System released out of MIT, with X11 protocol released in 1987 (now run by X.Org)
- 1985: Free Software Foundation (FSF) founded by Richard Stallman
- 1985: GNU Emacs text editor was released
- 1987: gcc released (now known as Gnu Compiler Collection)
- 1987: Perl released by Larry Wall.
- 1989: GPL 1.0 was created by Stallman (copyleft license)



The 1990s (I)

- 1991: Linux begun by Linus Torvalds
- 1992: 386BSD release
- 1992: Yggdrasil Linux /GNU/X release



Linus Benedict Torvalds



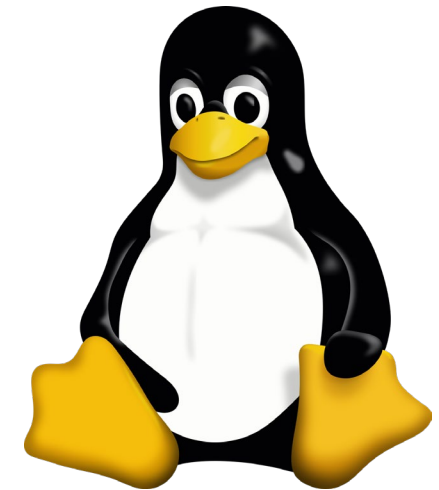
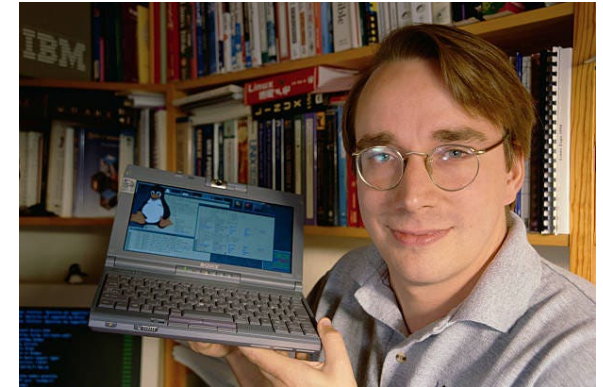
Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

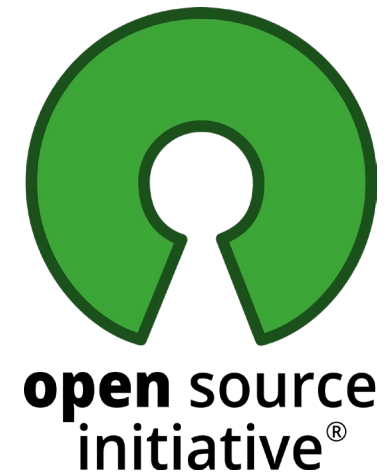
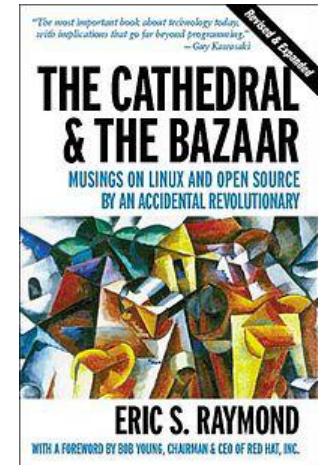
Linus (torv...@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).



The 1990s (II)

- 1993: Debian (non-commercial Linux distribution) and Red Hat are founded.
 - Red Hat was the first company built on open source to become very large
- 1997 – Eric Raymond writes “The Cathedral and the Bazaar”
 - Advantages of Open Source
 - Development in a decentralized way
- 1998: Open Source Initiative (OSI) was founded



Free software philosophical perspectives

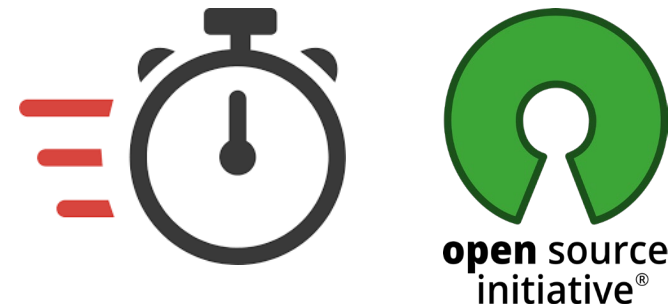
Idealistic

- Here, free means as in freedom, not beer. There is a profound belief that all software should be open for ideological and ethical reasons, not just technological ones.



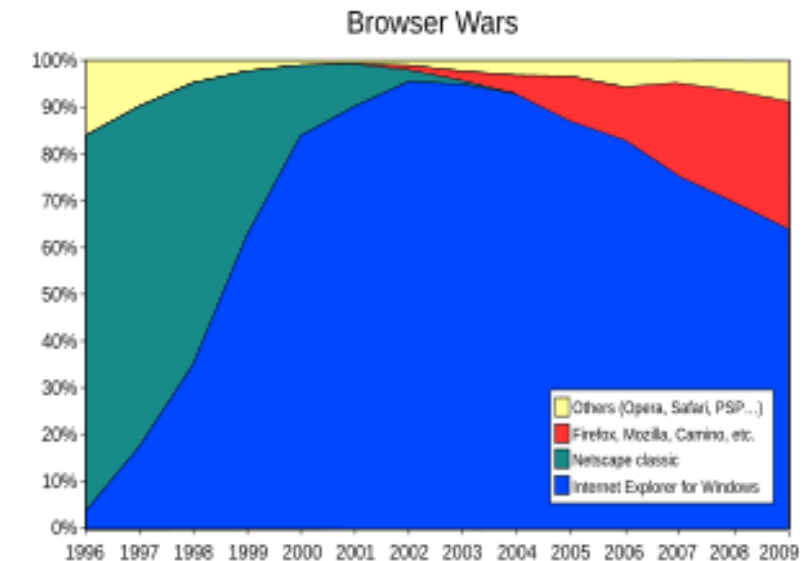
Pragmatic

- Here, the primary considerations are technical ones, including faster and better development involving more contributors and review, easier debugging, etc.



The 1990s (III)

- 1998: Netscape open sources its browser, which will later become Firefox (browser war)
- 1999: Sourceforge is launched
- 1999: OpenOffice released (eventually forks into LibreOffice).



The 2000s

- 2002: Blender released as an open source project
- 2003: Firefox released
- 2004: Ubuntu releases its first version, which Canonical builds on top of Debian
- 2005: Git released by Linus Torvalds
- 2007: Android released, based on Linux kernel; first devices on the market in 2008
- 2008: Chromium released by Google; basis of Google Chrome.



Questions

- What is open source about?
 - Price?
 - License?
 - Social Movement?
 - Hobby?
 - Market trend?



Why Open Source???

Why Open Source?

- Incentive to collaboration → human relationship
 - OSS fosters collaboration among developers globally. It builds a sense of community and encourages sharing of ideas
- Technical education
 - OSS provides hands-on experience, allowing students and developers to learn by contributing to real-world projects
- “Low-cost” access to “high-end” technology
 - OSS provides access to high-end technology without the financial barriers of proprietary software
- Decentralizes the “power” of the software
 - OSS decentralizes control by giving users the freedom to modify, improve, and redistribute software.

Why Open Source?

- Less duplicated code (and cost)
- Less bugs → More quality
 - “Given enough eye-balls all bugs are shallow”
- Customization and improvements
 - Users can adapt OSS to their specific needs by customizing it.

“Problems”

- Quality assessment of the solutions
 - Not all OSS projects have the same level of quality, so is hard to assess the quality of an OSS project
- Sustainability
 - Many OSS projects are maintained by volunteers, and sustaining long-term development can be challenging without proper funding or corporate backing.
- “Intellectual property”, trademarks, and patents
 - More on this, later
- Reputation (FUD- Fear, Uncertainty and Doubt)
 - Despite the growth of OSS, some organizations or individuals still hold reservations due to misconceptions about OSS’s reliability, security, or long-term viability

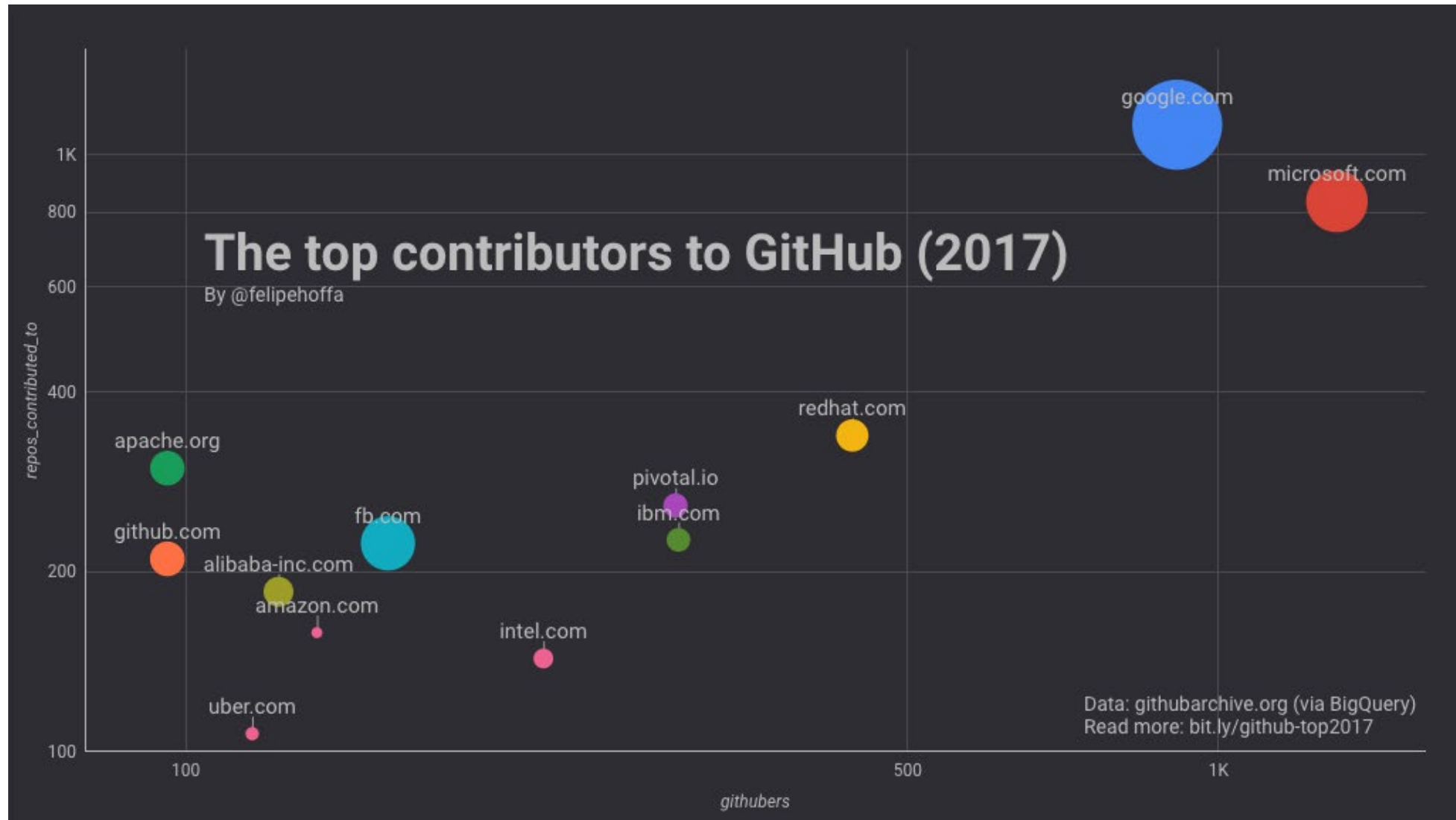
Therefore...

- Open source is about...
 - Freedom
 - License
 - Code as a way to share knowledge
 - Community!

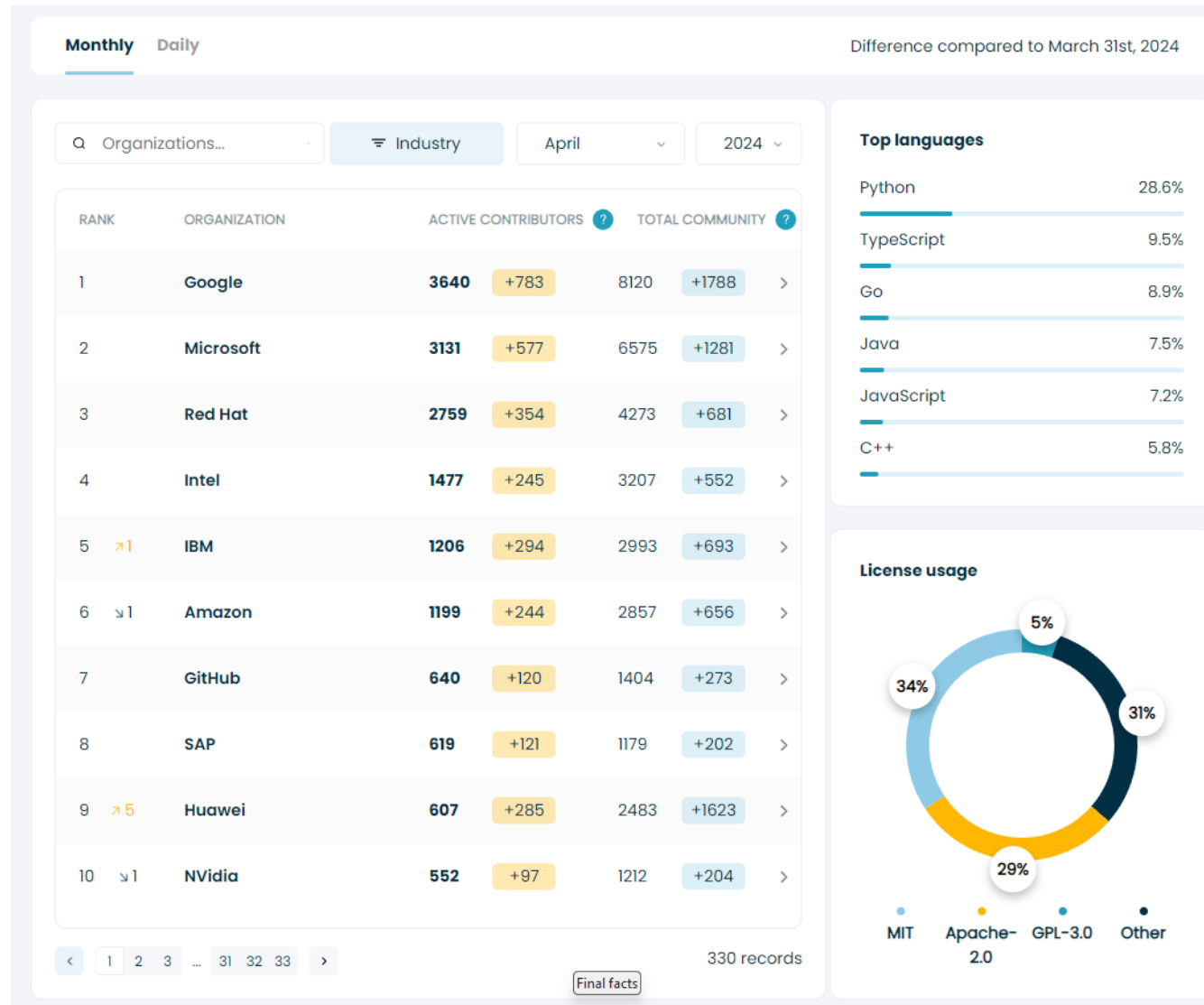


Final facts

And... Open source leads the way



Opensource Contributor Index



Why are companies joining OSS?



Branding/ Marketing

- Positioning in the communities
- Visibility
- Showcasing technological achievements
- Standardizing piece of technology



Attracting big talent

- Developers love
- Bypassing recruitment/ training steps
- Hiring tool

Reputation



Networking

- Access to big name companies
- Surrounded with peer company groups
- Getting hired



Building verifiable trust

- Showcasing expertise
- Build user fidelity



Fostering adoption

- Commercialize around their OSS project
- No vendor locking is attractive to users
- Get your technology adopted



Engineering need

- Where adopted technologies/ standards live
- Independence from licensing/ certification costs
- Build software more rapidly
- Neutrality and no locking policies



Business depends on OSS

- Keep dependency healthy
- Invest in security
- Guarantee of no company monopoly
- Governance and leadership
- Avoiding technical debt

Business Advantage



Closer channels

- Bigger reach to customers and users
- Readily R&D activity of the community
- No middleman



Coopetition

- Resource sharing
- Bigger market share
- Collaborate on the hard problem
- Better end product



Innovation

- Getting familiar with project interesting in the future
- Leveraging OSS ecosystem to guide business

Reciprocity



Sustain the



ecosystem

Sharing experiences

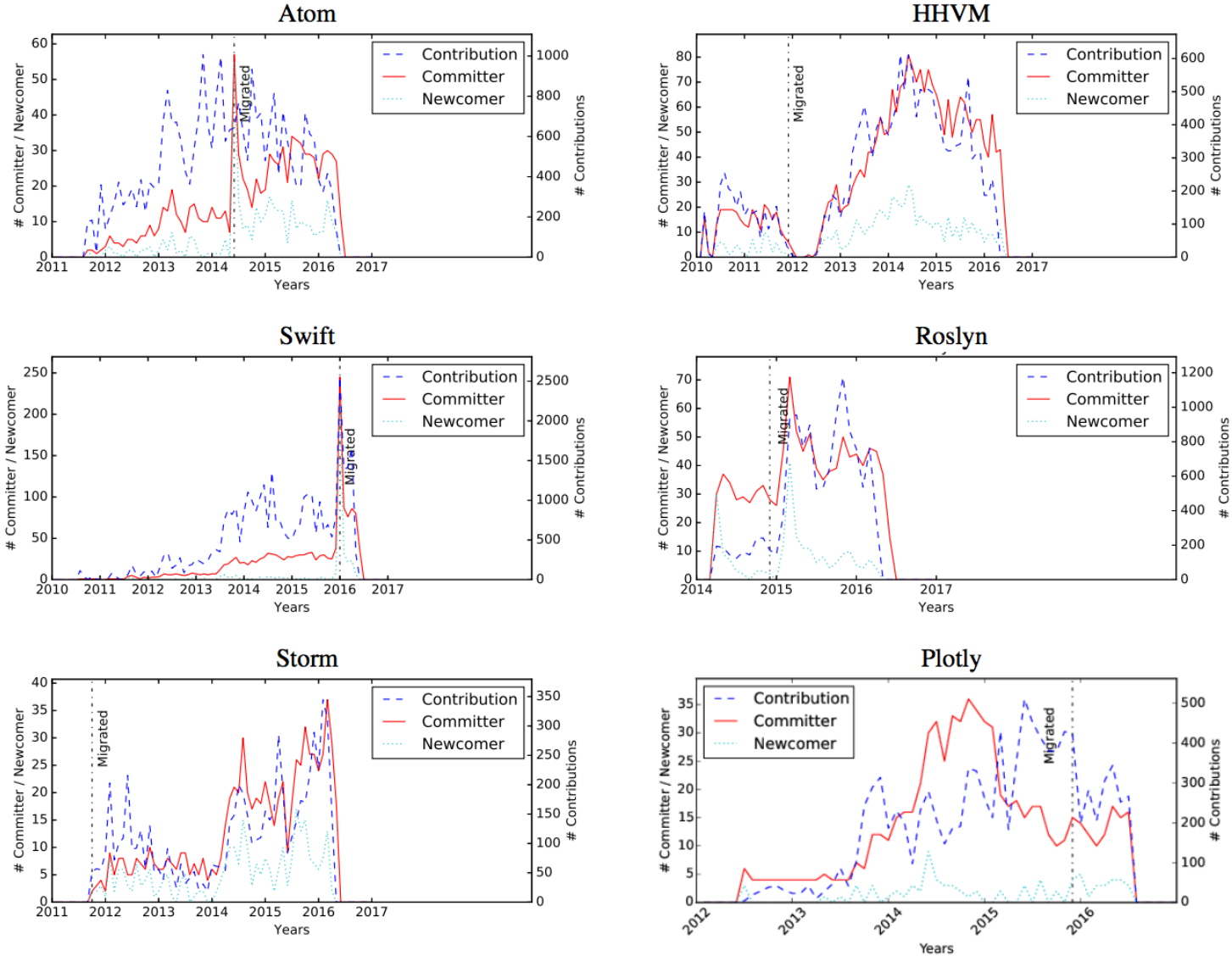
Founder Ideology



GUIZANI, M. et al. Rules of Engagement: Why and How Companies Participate in OSS. ICSE 2023

Why are companies Open-sourcing?





Pinto, Steinmacher, Dias and Gerosa, On the Challenges of Open-Sourcing Proprietary Software Projects, **Empirical Software Engineering**, 2018

All About Community!

- OpenStack:
 - 1.7M lines of code
 - 19 programming languages
 - 17K community members
 - 5K+ code contributors
 - 38K e-mail messages
- Mozilla Firefox:
 - 13.5M lines of code
 - 37 programming languages
 - 4K+ contributors (1K+ in the last 12 months)
 - 4,231 years of effort (COCOMO model) - first commit in 2002
- Swift:
 - 445K lines of code
 - Over the past 12 months, 400+ developers



<https://www.openhub.net/p/openstack>
<https://opensource.com/business/14/6/openstack-numbers>
https://www.openhub.net/p/apple_swift



Next Week

Creating an OSS project