

EN.601.210: Choose Your Own Open Source Software Adventure

Stephen R. Walli

Welcome!

- Course Objectives
- Syllabus
- Tools, Labs, etc.
- Introduce the Big Ideas

What is this course all about?

Objectives

- Understand how liberally licensed, collaboratively developed software fits into the world of a professional programmer and researcher.
- Be able to identify healthy open source projects and tools you can use in your projects and research without compromising your work.
- Have an approach to download, build, and use an open source project, and make modifications for their own use.
- Understand the difference between programming in the small, and the tooling and process to support software engineering at scale.
- Be aware of the high-level business and people issues (ownership, licenses, non-profits, community)

Syllabus

- A brief history of software collaboration from 1950 to 2020.
- What IS open source software? (The engineering economics of collaboration.)
- What healthy open source software projects look like. (On ramps and selfishness.)
- No scale without discipline (Software engineering 101)
- What is this madness called licensing and why you might care (a little).
- Scaling up even more (The role of non-profits in an open source enabled world)
- Governance is not what you think it is (Why voting is problematic, and engineers hum in the Internet Engineering Task Force)
- Codes of conduct, bad behavior, diversity, and other bits of social science.
- Why your open source software startup will fail.

Marking and Grades

- As this is an intersession course: Pass/Fail
- Attendance & Engagement & Lab 2 report
- Engagement == Use the class forum to ask questions

Tools, Tech, etc.

- All the class notes, lab instructions, etc. are on GitHub
<https://github.com/jhu-ospo-courses/JHU-EN.601.210>
- Labs depend on using Docker to create a virtual Ubuntu environment
<https://github.com/jhu-ospo-courses/JHU-EN.601.210/tree/main/labs#technology>
- [Blackboard](#) will be used for announcements, forums, grading

The Big Ideas

Meme #1: We have shared software since we have written software.

Meme #2: Writing good software is hard work.

Meme #3: There is no scale without discipline.

Meme #4: Software is inherently dynamic.

Meme #5: You always get more than you give.

Meme #6: Free-loaders means you're doing it right.

Meme #7: Don't confuse products with projects.

Meme #8: Don't confuse customers with community.

Meme #9 Companies shared liberally-licensed software long before the OSD.

Meme #10 Software freedom & open source are
different discussions.

Meme #11: Every project needs a license.

Meme #12: Non-profits have their place.

Lesson 1: A brief history of software collaboration from 1950 to 2020

Objectives

- Understand the breadth of history of software collaboration that culminated in the open source community
- Understand the key points along the timeline getting us to 2020
- Understand the open source definition formally and informally

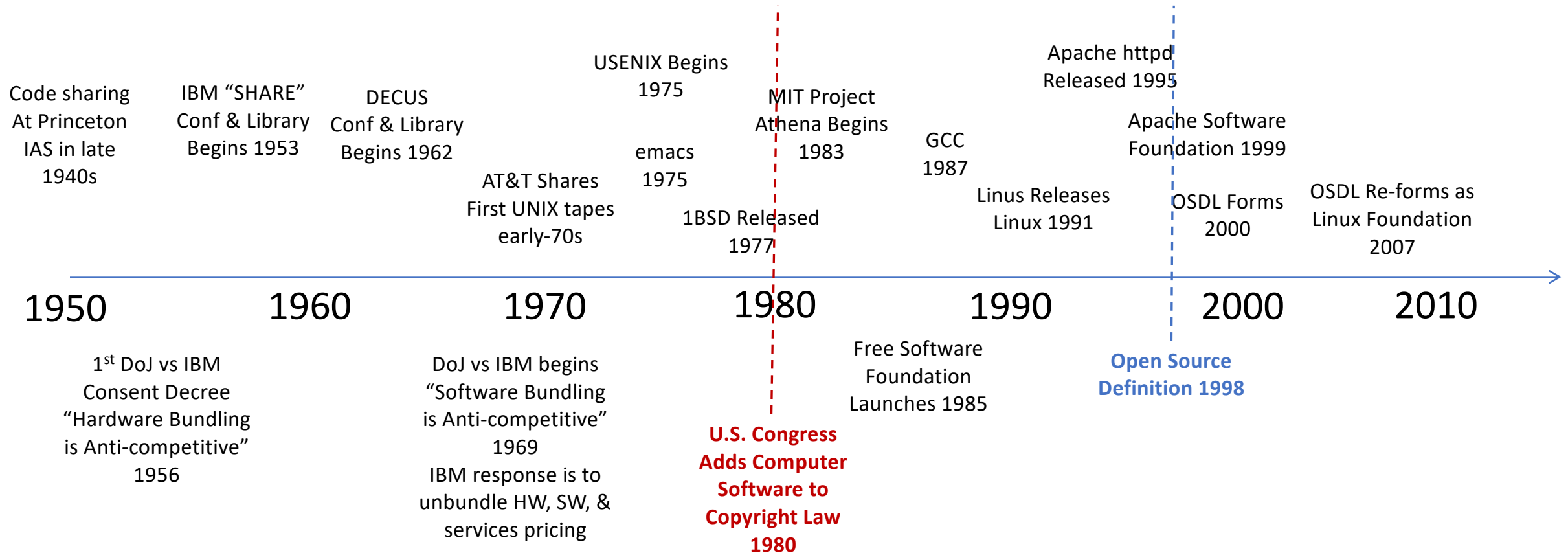
Notes:

<https://github.com/jhu-ospo-courses/JHU-EN.601.210/tree/main/lessons/1#notes>

We've known
how community
works since you
had a campfire
and I wanted to
sit beside it



We've collaborated on software since we've written software



Intellectual Property and Industrial Scale

Early experiments under copyright

- Project Athena at MIT started the MIT/X11 licensing experiment
- Free Software ethics began in early 1980s & led to copyleft & the GPL
- The Berkeley Software Distribution (BSD) begun at UC Berkeley led to the BSD license variants that began to show up in other collaborative projects like the early Apache web server project.
- The Perl language licensed under the Artistic License
- Mozilla started in 1998 when Netscape published its browser

Free Software

- The freedom to run the program as you wish, for any purpose (freedom 0).
 - The freedom to study how the program works and change it so it does your computing as you wish (freedom 1). [Access to the source code is a precondition for this.]
 - The freedom to redistribute copies so you can help others (freedom 2).
 - The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. [Access to the source code is a precondition for this.]
-
- Software freedom described in terms of rights of the software user

Open Source Software and the OSD

- Linus Torvalds releases Linux (1991) under the GPL
- Perl community + CPAN is an enormous community
- Apache httpd 1995 (still drives a third of the Web content)
- BSD lawsuit
- Netscape and Mozilla 1998
- Designed to be encompassing of all the major licenses and projects
- Debian Free Software Guidelines
- OSD described in terms of attributes of the software license

The Open Source Definition

1. Free Redistribution
2. Source Code
3. Derived Works
4. Integrity of The Author's Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License
8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must Be Technology-Neutral

The Open Source Definition

1. Free Redistribution
2. Source Code
3. Derived Works
4. Integrity of The Author's Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License
8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must Be Technology-Neutral

Victory! The Last 20+ Years

- Linux ubiquity
- Ethics versus engineering debates rage on
- All programming languages and almost all developer tool chains
- The Internet and the Web run on OSI-licensed software (LAMP)
- Almost all modern work in AI and the IoT is OSI-licensed