

Open Source

17-313 Spring 2023

Background

Subversion 1.5: A case study in open source release mismanagement

Authors:  [Hyrum K. Wright](#),  [Dewayne E. Perry](#), [Authors Info & Claims](#)

FLOSS '09: Proceedings of the 2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development • May 2009 • Pages 13–18 • <https://doi.org/10.1108/FLOSS.2009.5071354>



SOFTWARE ENGINEER INSIDER



WHAT IS SOFTWARE ENGINEERING?

SOFTWARE ENGINEER EDUCATION

For Hyrum Wright, Open-Source Software Coined Career Doors

By [Rishi Desai](#)



Hyrum Wright, PhD

Volunteering experience—every young person who's got it for it, most often through internships and fellowships. If you're into engineering, though, then there might be no better place to get into open-source software development. That's according to Hyrum Wright, who is now happily working as a software engineer in the Pittsburgh office of Google. The path to his present-day job started, he says, with the hours that he logged weekly



Links from the source

- Comment
- Weekly Edition
- Archives
- Search
- Contact
- Security
- Events calendar
- Legal documents

LWN FAQ

Write for us

Submit

Return to the

Discussion forum

EPISODE 70: HYRUM WRIGHT - SUBVERSION

Subversion recently became part of the Apache Software Foundation, and in a recent members of that project became members of the ASF.

User: Password: [Log In](#) [Subscribe](#) [Register](#)

Subversion 1.7.0 released

Posted October 12, 2010 by [corbe](#)

From: Hyrum Wright <hwright@apache.org>
To: announce <announce@apache.org>, announce <announce@subversion.apache.org>, Subversion <subversion.apache.org>, users <users@subversion.apache.org>
Subject: Apache Subversion 1.7.0 Released
Date: Tue, 11 Oct 2010 14:08:55 -0500
Message-ID: <20101011081149gl-jp-vj@wht.wright@subversion.apache.org>
Archive-link: [article](#) [thread](#)

I'm happy to announce the release of Apache Subversion 1.7.0. This is a major new feature release of Subversion, containing many new features, bugfixes and general usability improvements. The full extent of the changes is too large to summarize here, and I encourage



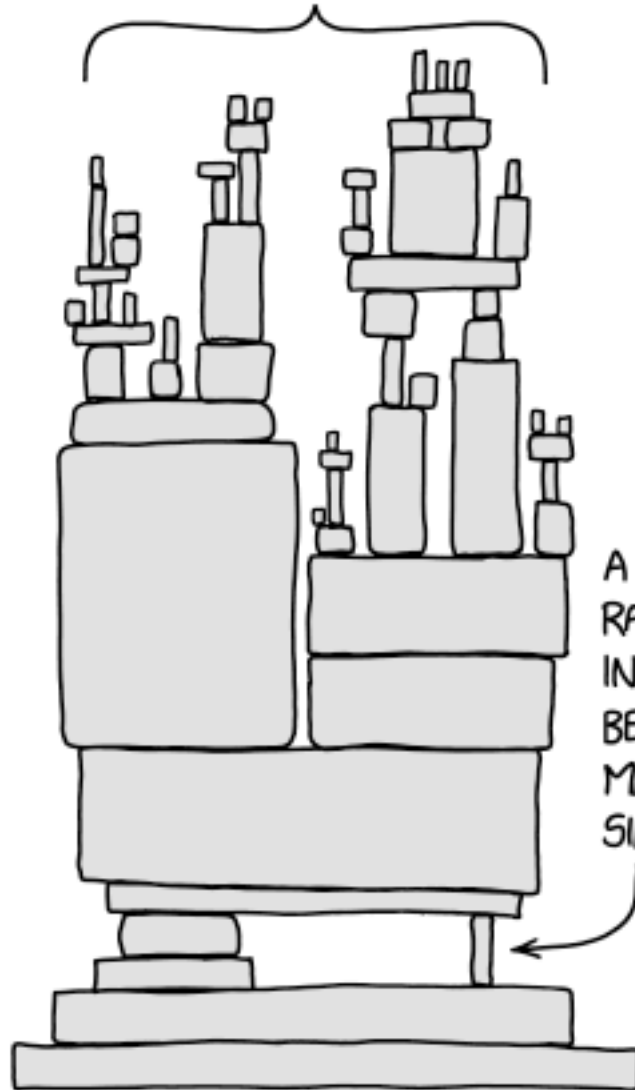
Open Source

aka Free Software

aka Free and Open Source Software



ALL MODERN DIGITAL
INFRASTRUCTURE



A PROJECT SOME
RANDOM PERSON
IN NEBRASKA HAS
BEEN THANKLESSLY
MAINTAINING
SINCE 2003

“Free as in speech”



Open Source

Anchor

Do you want to support Open Source? [Book a meeting with us now.](#)

Innovator

DataStax

Premier

Bloomberg
Engineering

 **indeed**

 **Red Hat**



Maintainer





















Supporter









Free Software vs. Open Source

- Free software origins (70-80s ~Stallman)
 - Cultish Political goal
 - Software part of free speech
 - free exchange, free modification
 - proprietary software is unethical
 - security, trust
 - GNU project, Linux, GPL license
- Open source (1998 ~O'Reilly)
 - Rebranding without political legacy
 - Emphasis on internet and large dev/user involvement
 - Openness toward proprietary software/coexist
 - (Think: Netscape becoming Mozilla)



Free Software vs. Open Source

- 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 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.

The Open Source Definition

free-redistribution

Introduction

Open source doesn't just mean access to the source code. The distributor's terms of open-source software must comply with the following criteria:

1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost; preferably downloading via the internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code

The license may restrict source code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups

The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License

The rights attached to the program must apply to all in whom the program is redistributed with the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software.

The Cathedral and the Bazaar



WHEN YOU PROGRAM OPEN SOURCE,
YOU'RE PROGRAMMING

COMMUNISM



A REMINDER
from
YOUR FRIENDS AT MICROSOFT

Perception:

- Anarchy
- Demagoguery
- Ideology
- Altruism
- Many eyes



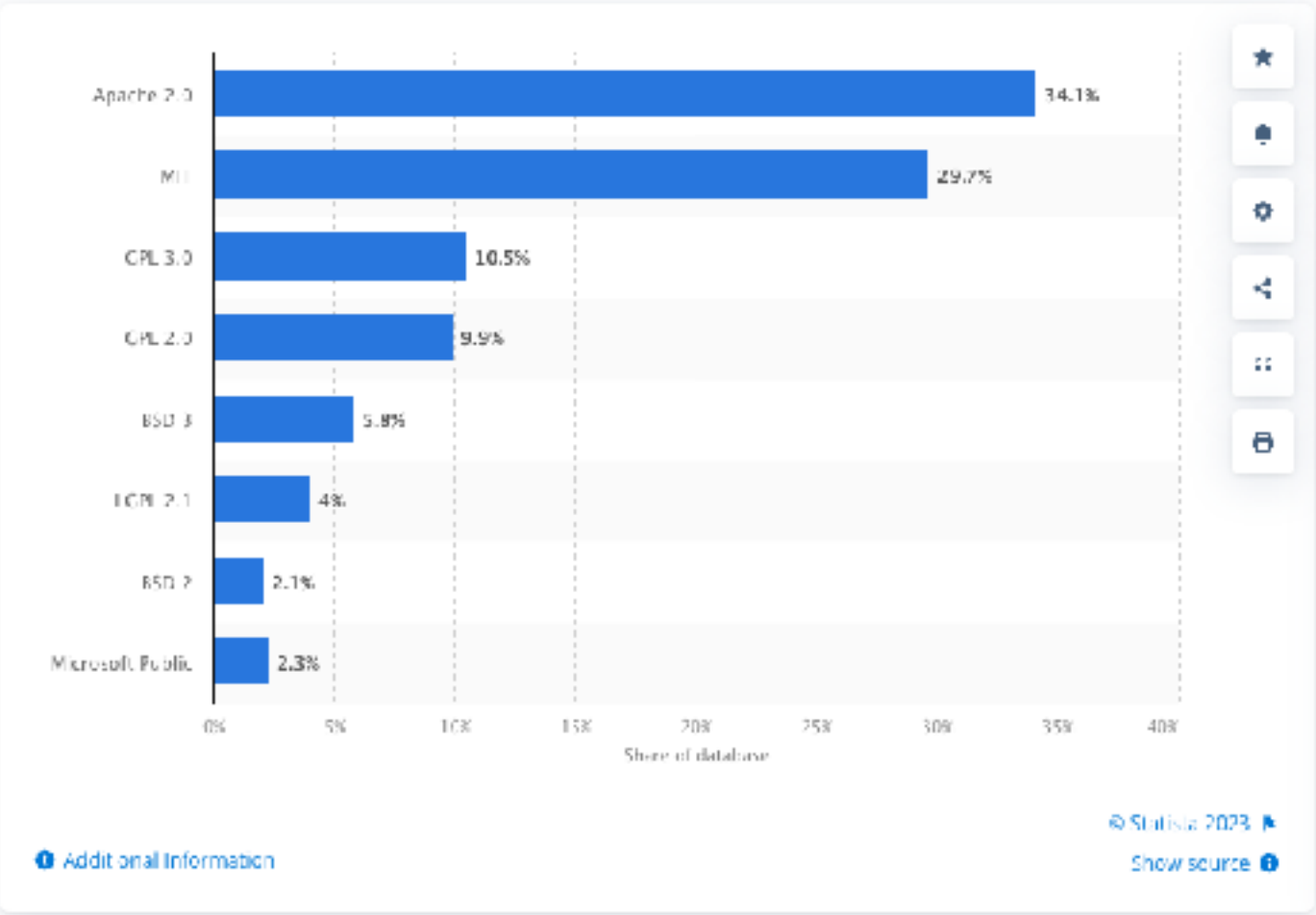
Using Open Source: Licenses

Note: IANAL (I am not a lawyer)

Why learn about licenses?

- Companies will avoid certain licenses – commonly the copyleft licenses
- Specific licenses may provide competitive advantages
- You may eventually want to release open source software or become more involved in an open source project

Most popular open source licenses worldwide in 2021



GNU General Public License: The Copyleft License

- Nobody should be restricted by the software they use. There are four freedoms that every user should have:
 - the freedom to use the software for any purpose,
 - the freedom to change the software to suit your needs,
 - the freedom to share the software with your friends and neighbors, and
 - the freedom to share the changes you make.
- Code must be made available
- Any modifications must be relicensed under the same license (copyleft)



GPL 2.0 and 3.0 – Addresses free software problems

- 2.0 - Court ruling cannot nullify the license and if a court decision and this license contradict in distribution requirements, then the software cannot be distributed
- 3.0 – patent grant and prevent Tivoization
- Not compatible with each other; Can't copyleft both at the same time – phrase: “GPL Version 3 or any later version”

Which license to choose?

The screenshot shows a PowerPoint presentation with a browser window open to choosealicense.com/licenses/. The browser displays two license options: Apache and GPL.

Apache License

A permissive license that also provides an express grant of patent rights from contributors to users.

Required	Permitted	Forbidden
<ul style="list-style-type: none">License and copyright noticeState Changes	<ul style="list-style-type: none">Commercial UseDistributionModificationPatent UsePrivate UseSublicensing	<ul style="list-style-type: none">Hold LiableUse Trademark

[View full Apache License 2.0 license >](#)

GPL

GNU Affero GPL v3.0 GNU GPL v2.0 **GNU GPL v3.0**

Required	Permitted	Forbidden
<ul style="list-style-type: none">Disclose SourceLicense and copyright noticeState Changes	<ul style="list-style-type: none">Commercial UseDistributionModificationPatent UsePrivate Use	<ul style="list-style-type: none">Hold Liable

The PowerPoint slide on the left is titled "Apache License" and contains the following text:

- Apache
- Simple to use, with no copyleft
- Permitted to derive source code
- License each derivative to use under a license
- Patent rights: contribution to use the work

Slide 33 of 42. "Office Theme".

Dual License Business Model



- Released as GPL which requires a company using the open source product to open source it's application
- Or companies can pay \$2,000 to \$10,000 annually to receive a copy of MySQL with a more business friendly license

Risk: Incompatible Licenses

- Sun open sourced OpenOffice, but when Sun was acquired by Oracle, Oracle temporarily stopped the project.
- Many of the community contributors banded together and created LibreOffice
- Oracle eventually released OpenOffice to Apache
- LibreOffice changed the project license so LibreOffice can copy changes from OpenOffice but OpenOffice cannot do the same due to license conflicts

MIT License

- Must retain copyright credit
- Software is provided as is
- Authors are not liable for software
- No other restrictions

LGPL

- Software must be a library
- Similar to GPL but no copyleft requirement

BSD License

- No liability and provided as is.
- Copyright statement must be included in source and binary
- The copyright holder does not endorse any extensions without explicit written consent

Apache License

- Apache
 - Similar to GPL with a few differences
 - Not copyleft
 - Not required to distribute source code
 - Does not grant permission to use project's trademark
 - Does not require modifications to use the same license

Producing Open Source

- Like companies, different project have different:
 - Structures
 - Cultures
 - Organizations
- Many open source projects are supported by a legal entity, such as a foundation.

Example: Apache



WHAT MAKES THE APACHE WAY SO HARD TO DEFINE?

The Apache Way is a living, breathing interpretation of one's experience with our community-led development process. Apache is unique, diverse, and focused on the activities needed at a particular stage of the project's lifetime, including nurturing community, building awareness. What is important is that they embrace:

- **Earned Authority:** all individuals are given the opportunity to participate, but their influence is based on publicly earned community. Merit lies with the individual, does not expire, is not influenced by employment status or employer, and is not project cannot be applied to another). [More on merit.](#)
- **Community of Peers:** individuals participate at the ASF, not organizations. The ASF's flat structure dictates that roles are equal weight, and contributions are made on a volunteer basis (even if paid to work on Apache code). The Apache community with respect in adherence to our [Code of Conduct](#). Domain expertise is appreciated; Benevolent Dictators For Life are di participation.
- **Open Communications:** as a virtual organization, the ASF requires all communications related to code and decision-making asynchronous collaboration, as necessitated by a globally-distributed community. Project mailing lists are archived, public
 - dev@ (primary project development)
 - user@ (user community discussion and peer support)
 - commits@ (automated source change notifications)
 - occasionally supporting roles such as marketing@ (project visibility)

...as well as restricted, day-to-day operational lists for Project Management Committees. Private decisions on code, policies, or project discourse and transactions must be brought on-list. [More on communications and the use of mailing lists.](#)

- **Consensus Decision Making:** Apache Projects are overseen by a self-selected team of active volunteers who are contributors. Projects are auto-governing with a heavy slant towards driving consensus to maintain momentum and productivity. We establish at all times, holding a vote or other coordination may be required to help remove any blocks with binding decisions. [More on decision making and voting.](#)
- **Responsible Oversight:** The ASF governance model is based on trust and delegated oversight. Rather than detailed rule-governance is principles-based, with self-governing projects providing reports directly to the Board. Apache Committers reviewed commits, employing mandatory security measures, ensuring license compliance, and protecting the Apache community from abuse. [More on responsibility.](#)

Producing Open Source: Governance

- Consensus
 - Apache
 - Rust
 - (many others)
- Dictator
 - Python
 - Linux
- Corporate

All Our Patent Are Belong To You

Elon Musk, CEO · June 12, 2014

Yesterday, there was a wall of Tesla patents in the lobby of our Palo Alto headquarters. That is no longer the case. They have been removed, in the spirit of the open source movement, for the advancement of electric vehicle technology.

Tesla Motors was created to accelerate the advent of sustainable transport. If we clear a path to the creation of compelling electric vehicles, but then lay intellectual property landmines behind us to inhibit others, we are acting in a manner contrary to that goal. Tesla will not initiate patent lawsuits against anyone who, in good faith, wants to use our technology.

When I started out with my first company, Zip2, I thought patents were a good thing and worked hard to obtain them. And maybe they were good long ago, but too often these days they serve merely to stifle progress, entrench the positions of giant corporations and enrich those in the legal profession, rather than the actual inventors. After Zip2, when I realized that receiving a patent really just meant that you bought a lottery ticket to a lawsuit, I avoided them whenever possible.

At Tesla, however, we felt compelled to create patents out of concern that the big car companies would copy our technology and then use their massive manufacturing, sales and marketing power to overwhelm Tesla. We couldn't have been more wrong. The unfortunate reality is the opposite: electric car programs (or programs for any vehicle that doesn't burn hydrocarbons) at the major manufacturers are small to non-existent, constituting an average of far less than 1% of their total vehicle sales.

At best, the large automakers are producing electric cars with limited range in limited volume. Some produce no zero emission cars at all.

Given that annual new vehicle production is approaching 100 million per year and the global fleet is approximately 2 billion cars, it is impossible for Tesla to build electric cars fast enough to address the carbon crisis. By the same token, it means the market is enormous. Our true competition is not the small trickle of non-Tesla electric cars being produced, but rather the enormous flood of gasoline cars pouring out of the world's factories every day.

We believe that Tesla, other companies making electric cars, and the world would all benefit from a common, rapidly-evolving technology platform.

Technology leadership is not defined by patents, which history has repeatedly shown to be small protection indeed against a determined competitor, but rather by the ability of a company to attract and motivate the world's most talented engineers. We believe that applying the open source philosophy to our patents will strengthen rather than diminish Tesla's position in this regard.

Ecosystem has Changed

February 2, 1978

In Open Letter to Hobbyists

As you know, we have a hobby market right now in the land of good software, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?

Alas, a year ago, Paul Baker and myself, recognizing the early success of experts, started writing BASIC for the Microsoft BASIC. Through the initial work took only two months, the three of us have spent most of the last year documenting, improving and adding features to BASIC. Now we have 4C, 5B, EXTENDED, TOP and BASIC BASIC. The value of the computer that we have used exceeds \$20,000.

The response we have gotten from the hundreds of people who say they are using BASIC are all very positive. We certainly think we are ahead. However, if most of these "users" never know BASIC (less than 10% of all BASIC owners have bought BASIC), and if the amount of royalties we have received from sales is negligible

Do you think you can't do it by creating software to get back at guys who have been so successful? Some of these guys make money selling software. The royalty paid to us, the normal, the 10% and the overhead make it a break-even operation. The thing you do is to prevent your software from being written. The way all that to do is to prevent your software from being written. The way all that to do is to prevent your software from being written. The way all that to do is to prevent your software from being written.

What about the guys who do sell BASIC BASIC, aren't they making money on hobby software? Yes, but those who have been successful do so in the land of the good. They are the ones who give hobbyists a real name, and would be happy to see it. If you are not making any money at all.

I would appreciate letters from any one who wants to pay us, or has a suggestion or comment. You can write us at 1140 Microsoft Dr., #1140 Albuquerque, New Mexico, 87104. We would prefer to have you being able to hire the programmer and design the hobby market with your software.

Bill Gates
General Partner, Microsoft

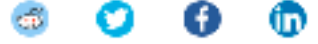
“...most of you steal your software...”

Redmond top man Satya Nadella: 'Microsoft LOVES Linux'

Open-source 'love' fairly runneth over at cloud event



20 Oct 2014 at 23:45, Neil McAllister



Risks of *not* open sourcing something?



Producing Open Source: Getting Involved

1. Find a project
2. Start Writing Code
3. ?
4. Profit!



Producing Open Source: Skills

- Written communication
 - Email, chat, and design documents are core to asynchronous work
- “Thick” skin
- Technical ability
- Political ability