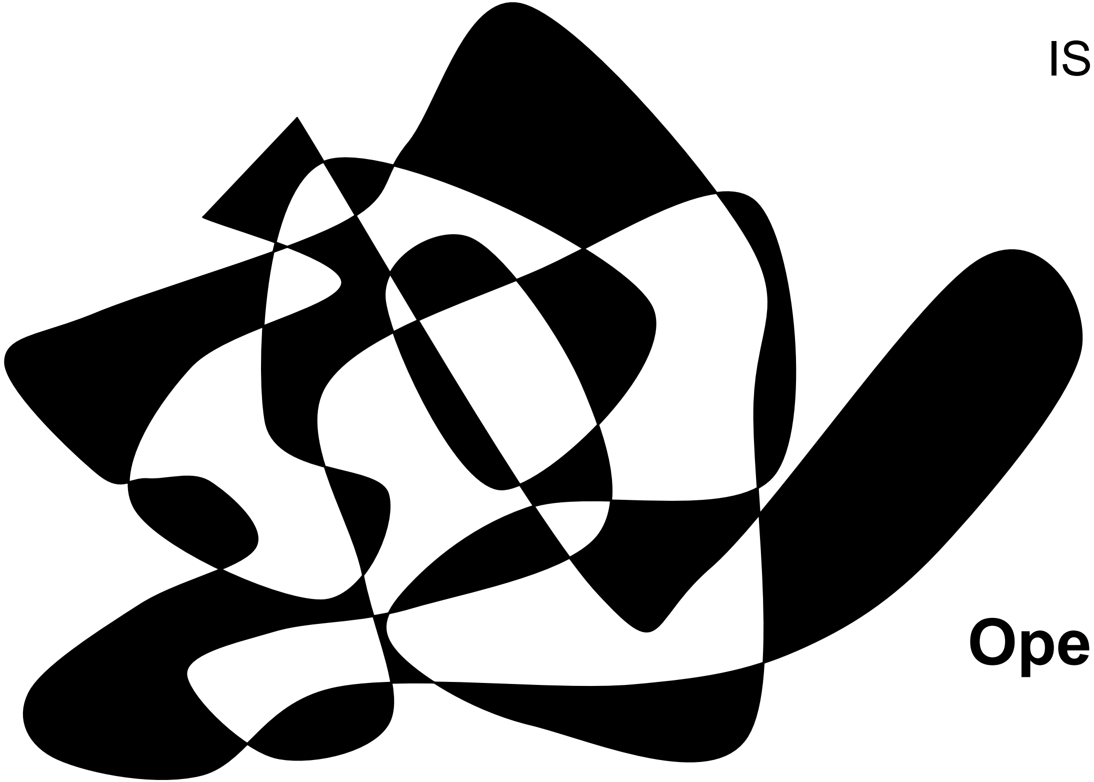
Advanced Project Management

IS 594, Section PJ



Open-source and Working Open

History of Open Source

Origins in the 1950s and 1960s: software in the public domain (Academic Labs).

Worked against intellectual property rights of corporations, rise of proprietary software.

Public domain software lead to hacking culture (1980s).

Public domain works: UNIX (Bell Labs, 1970), Tex (Donald Knuth, 1978).

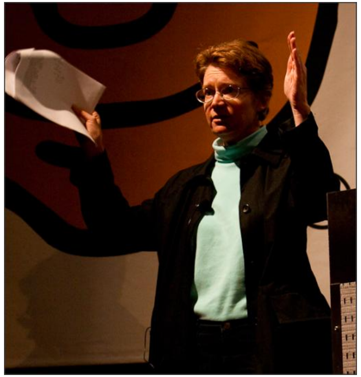
Copyleft licensing: GNU project (Richard Stallman, 1983).

Open-source Software in the Public Consiousness

Richard Stallman founded the Free Software Foundation (FSF) in 1985. Free access (not financially), stressed user freedom.

https://www.gnu.org/philosophy/open-source-misses-the-point.en.html

Christine Peterson (Foresight Institute) proposed the term “open-source” in 1998: make concept accessible and more specific to a broader community of users.



https://opensource.com/article/18/2/coining-term-open-source-software

Defining Open Source

Denoting software for which the original source code is made *freely available* and may be *redistributed and modified*.

Software is “free” or “libre” in accordance with the Free Software Definition and Definition of Free Cultural Works.

Defining Open Source

Denoting software for which the original source code is made *freely available* and may be *redistributed and modified*.

Software is “free” or “libre” in accordance with the Free Software Definition and Definition of Free Cultural Works.

Five Pillars of Open Source

Five things make a software release “open source”:

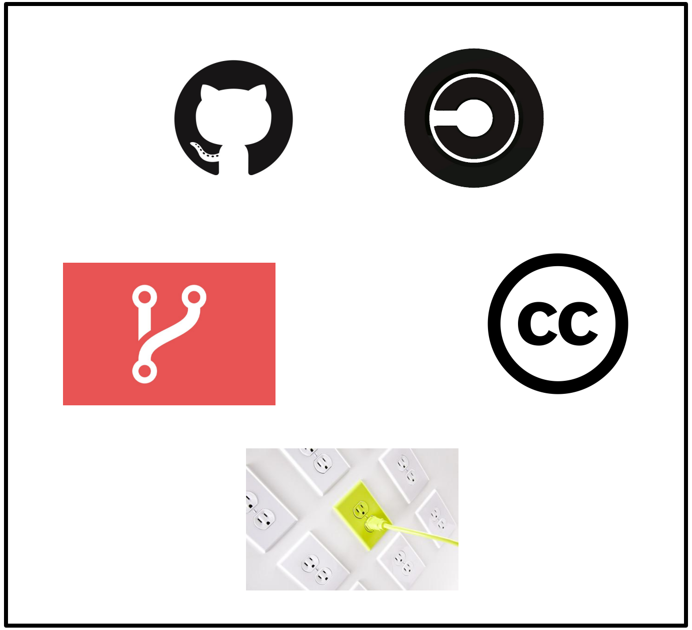
1) Publicly-available source code.

2) Copyleft licensing.

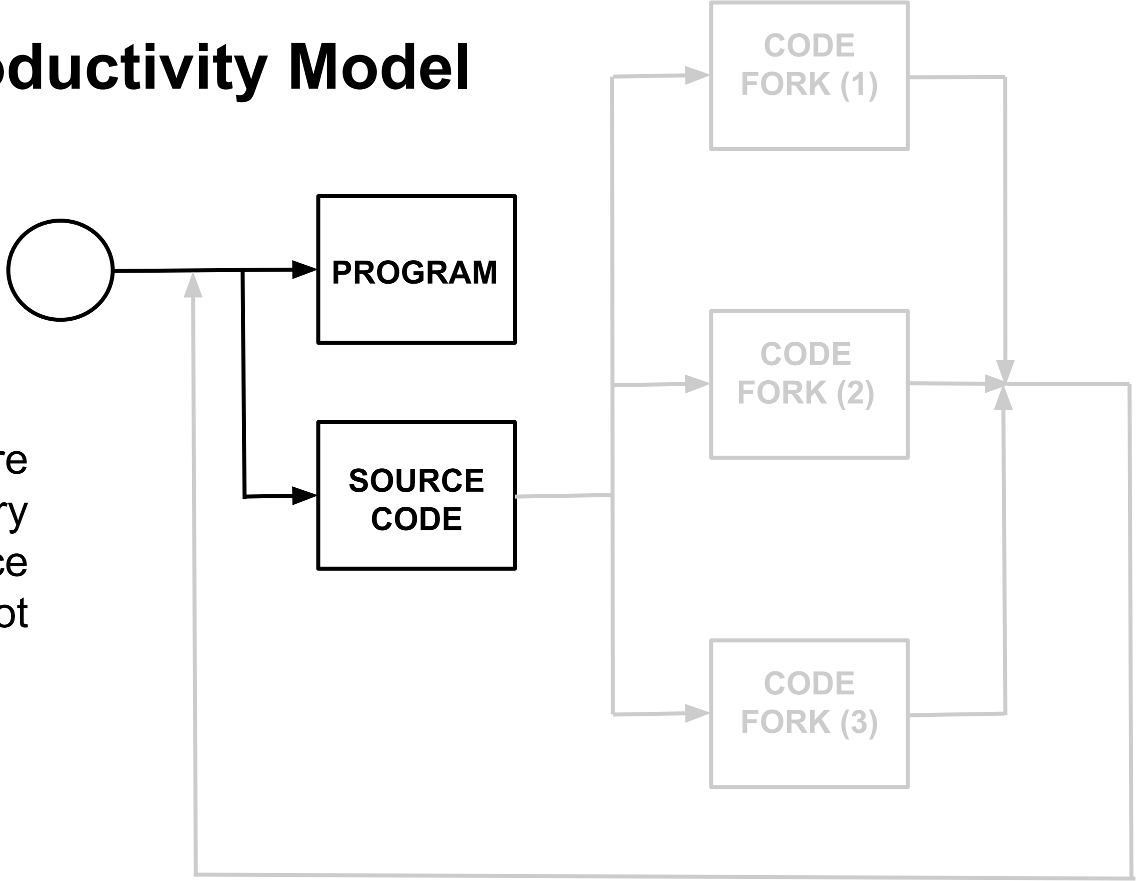
3) Free distribution and remixing.

4) Technology neutral.

5) Version-control.

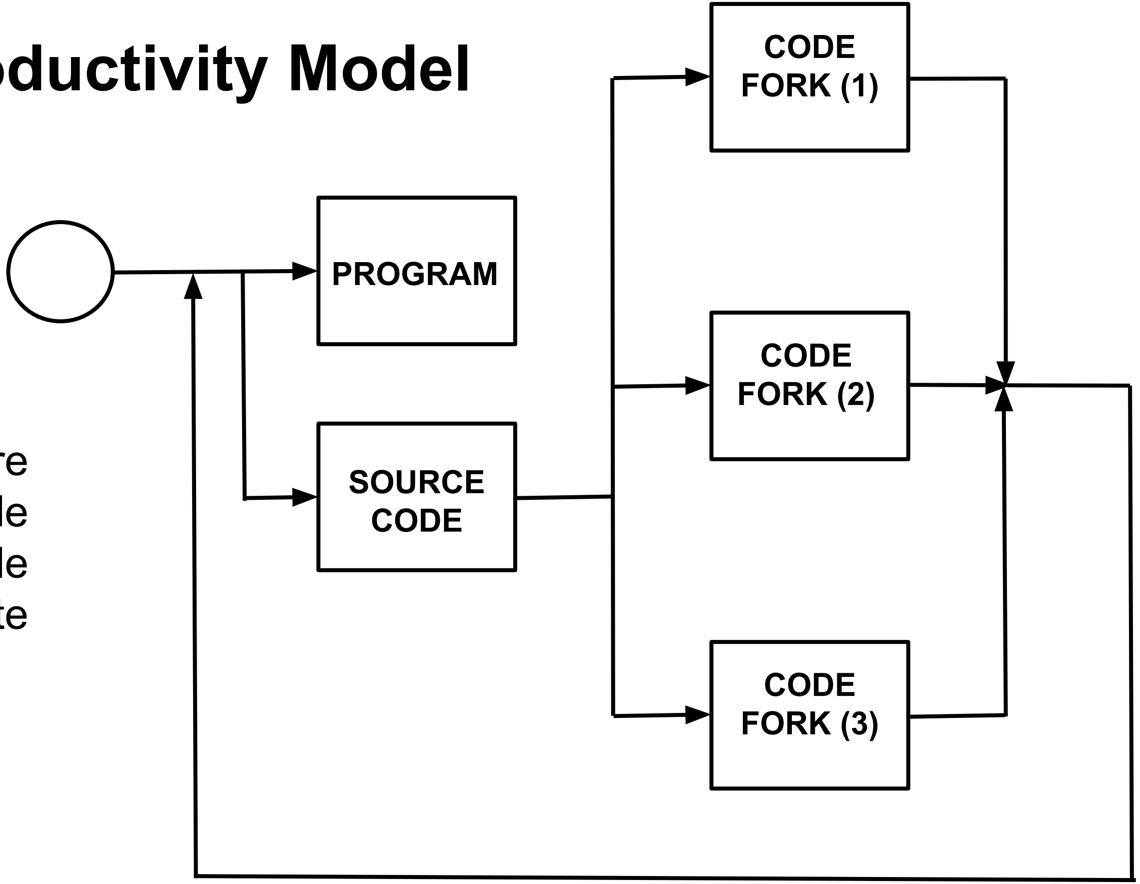


Open-source Productivity Model



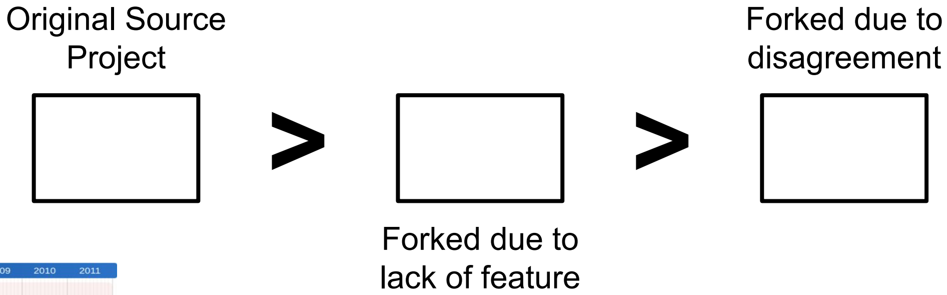
Closed-source software provides a proprietary “program” with source code that is not freely-available.

Open-source Productivity Model



Open-source software provides freely-modifiable source code where people can make forks (alternate copies).

Open-source Genealogies



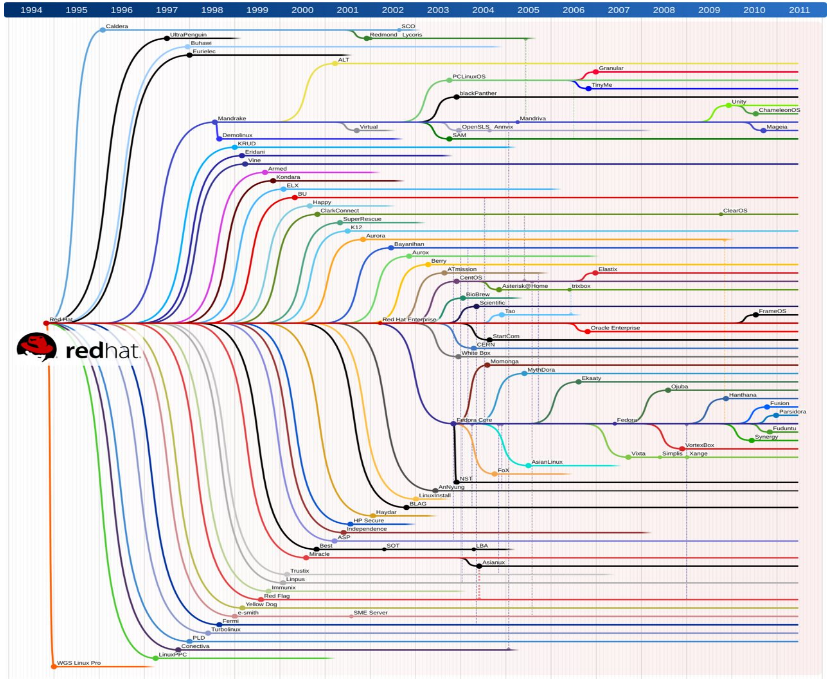
Original Source Project

>

Forked due to lack of feature

>

Forked due to disagreement



The Hidden Benefit of Giving Back to Open Source Software

Harvard Business Review, September 5 (2018)

Open-source projects can be thought of as a public good.

Companies can benefit from hiring workers with experience in open-source organizations.

- open-source contributions outside firm benefits individuals (skill building, access to technological systems).

- companies can benefit from hiring open-source contributors (can build upon, utilize value of open-source tools).

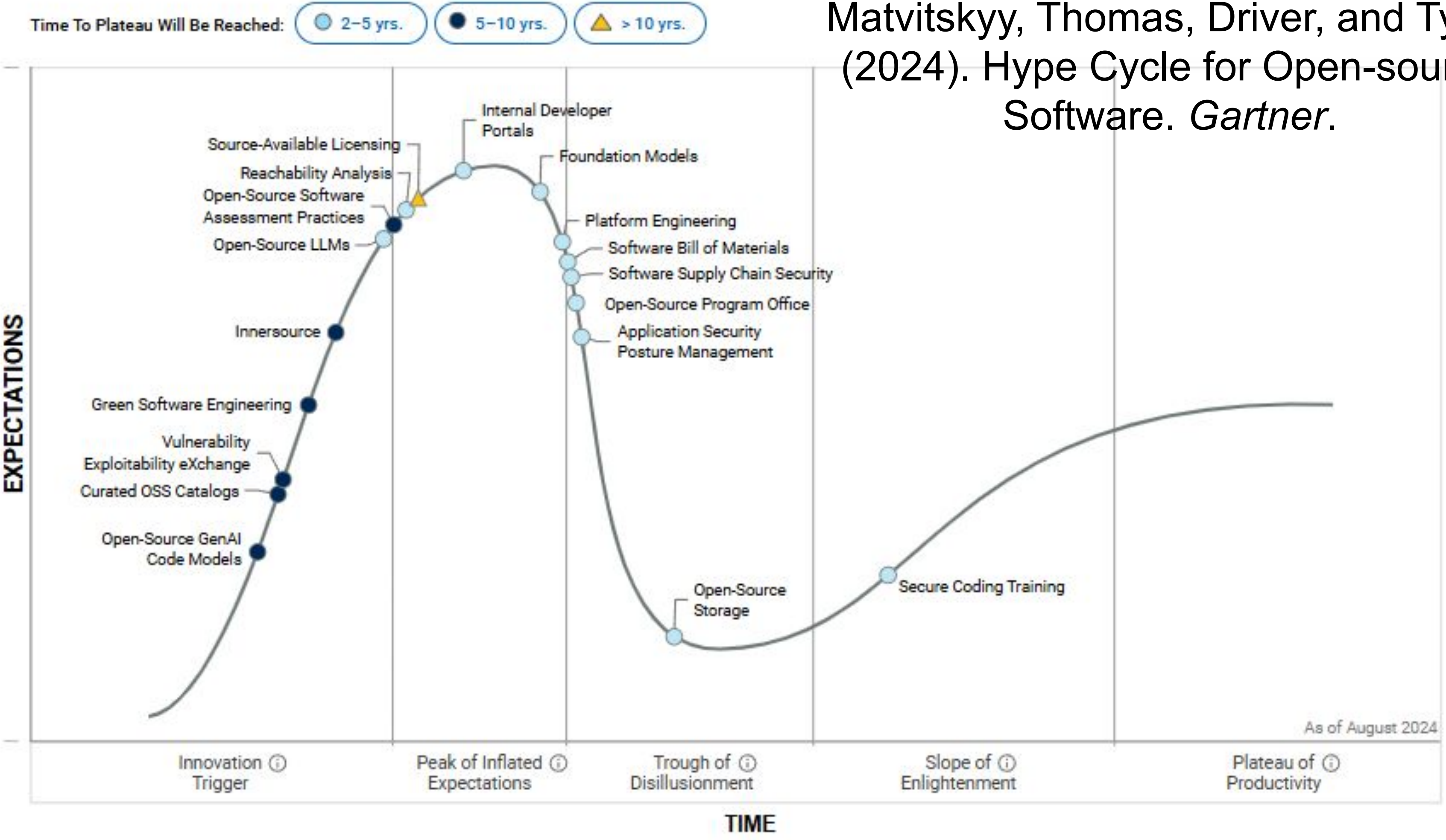
Debian Social Contract (1997)

Ensure that software remains open and free as it gets forked, scales up:

- gift improvements to the community that made the operating system possible.
- transparency in methods, software, and organization.
- staying focused on the users and the software that started the phenomenon.
- enable the use of free software with non-free software.

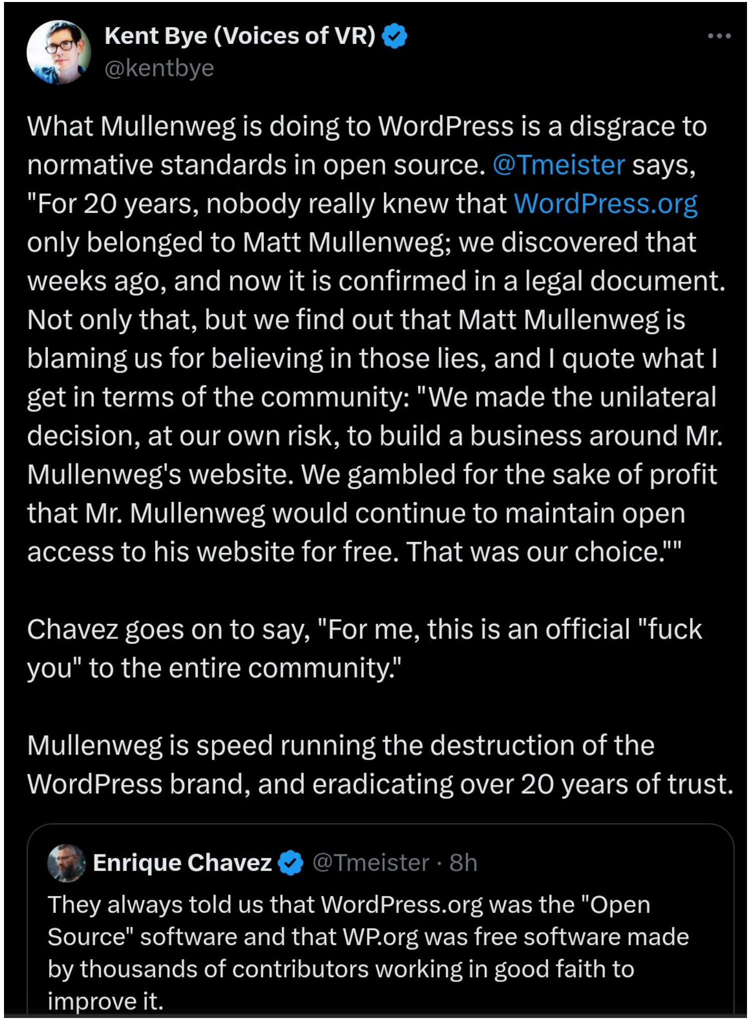
Forms the basis for the Open Source Definition (https://en.wikipedia.org/wiki/The_Open_Source_Definition)

Matvitskyy, Thomas, Driver, and Tyagi (2024). Hype Cycle for Open-source Software. *Gartner*.



Breaking the norms of open-source

WordPress fork controversy.



Kent Bye (Voices of VR)
@kentbye

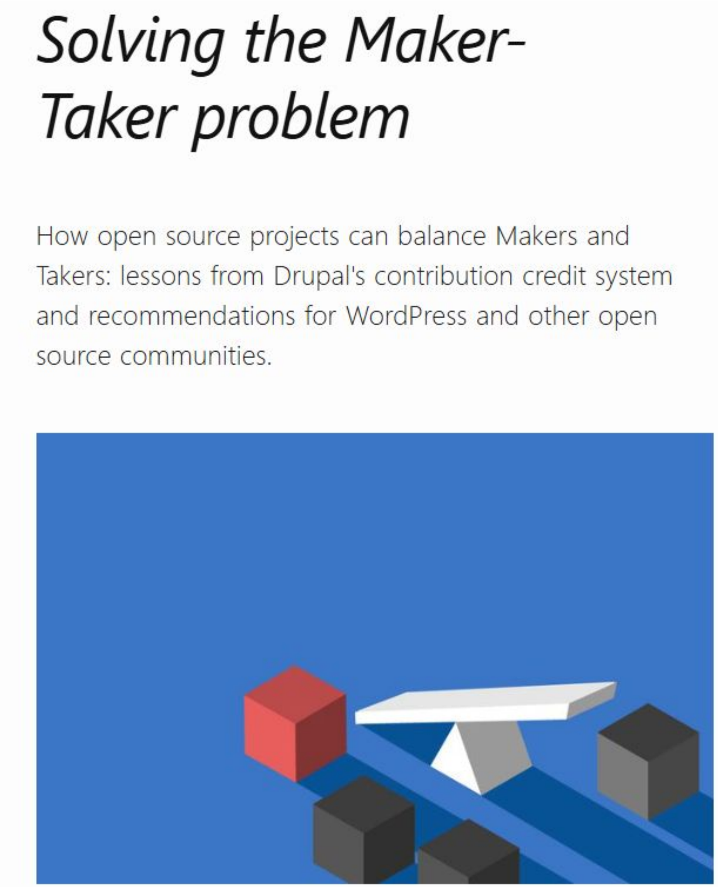
What Mullenweg is doing to WordPress is a disgrace to normative standards in open source. @Tmeister says, "For 20 years, nobody really knew that WordPress.org only belonged to Matt Mullenweg; we discovered that weeks ago, and now it is confirmed in a legal document. Not only that, but we find out that Matt Mullenweg is blaming us for believing in those lies, and I quote what I get in terms of the community: "We made the unilateral decision, at our own risk, to build a business around Mr. Mullenweg's website. We gambled for the sake of profit that Mr. Mullenweg would continue to maintain open access to his website for free. That was our choice.""

Chavez goes on to say, "For me, this is an official "fuck you" to the entire community."

Mullenweg is speed running the destruction of the WordPress brand, and eradicating over 20 years of trust.

Enrique Chavez @Tmeister · 8h

They always told us that WordPress.org was the "Open Source" software and that WP.org was free software made by thousands of contributors working in good faith to improve it.



Solving the Maker-Taker problem

How open source projects can balance Makers and Takers: lessons from Drupal's contribution credit system and recommendations for WordPress and other open source communities.

https://dri.es/solving-the-maker-taker-problem

Wordpress (organization) v. WP Engine (fork)

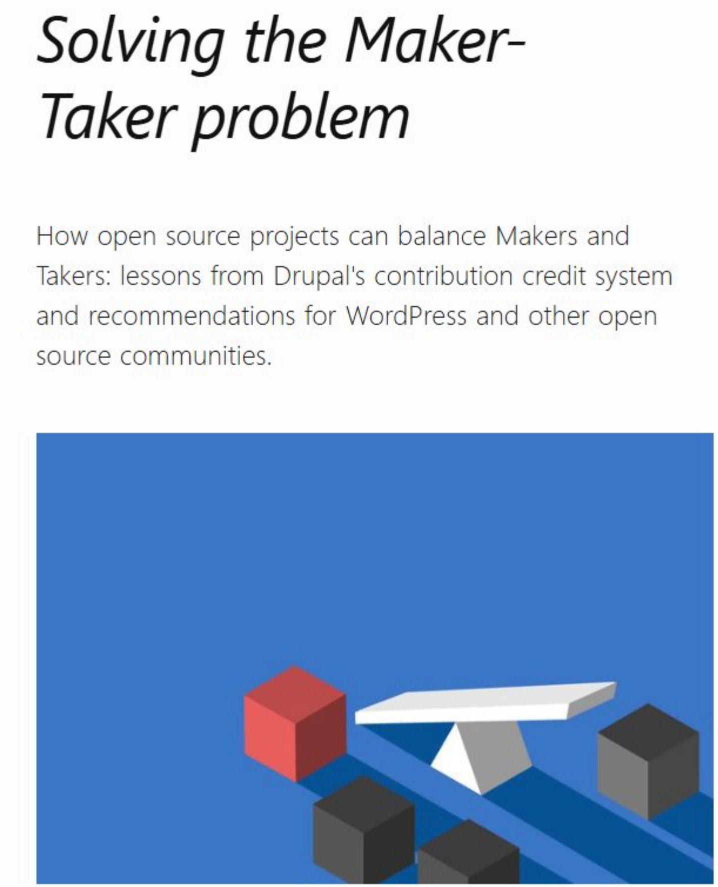
https://techcrunch.com/2024/10/20/wordpress-vs-wp-engine-drama-explained

Wordpress (founder Matt Mullenweg) and WP Engine (solution provider for Wordpress fork).

- WP Engine started disabling core features (revision history) as a cost-saving measure.
- WP Engine (Auttomatic) does not contribute to upstream organization.
- In retaliation, Wordpress has restricted fork access for all users. Actions have widespread effect.

Should developers rely on commercial open source products related to WordPress, especially when their access is capricious?

- Single person control (benevolent dictator governance) + commercial conflict = unraveling of open-source functionality.



Solving the Maker-Taker problem

How open source projects can balance Makers and Takers: lessons from Drupal's contribution credit system and recommendations for WordPress and other open source communities.

https://dri.es/solving-the-maker-taker-problem

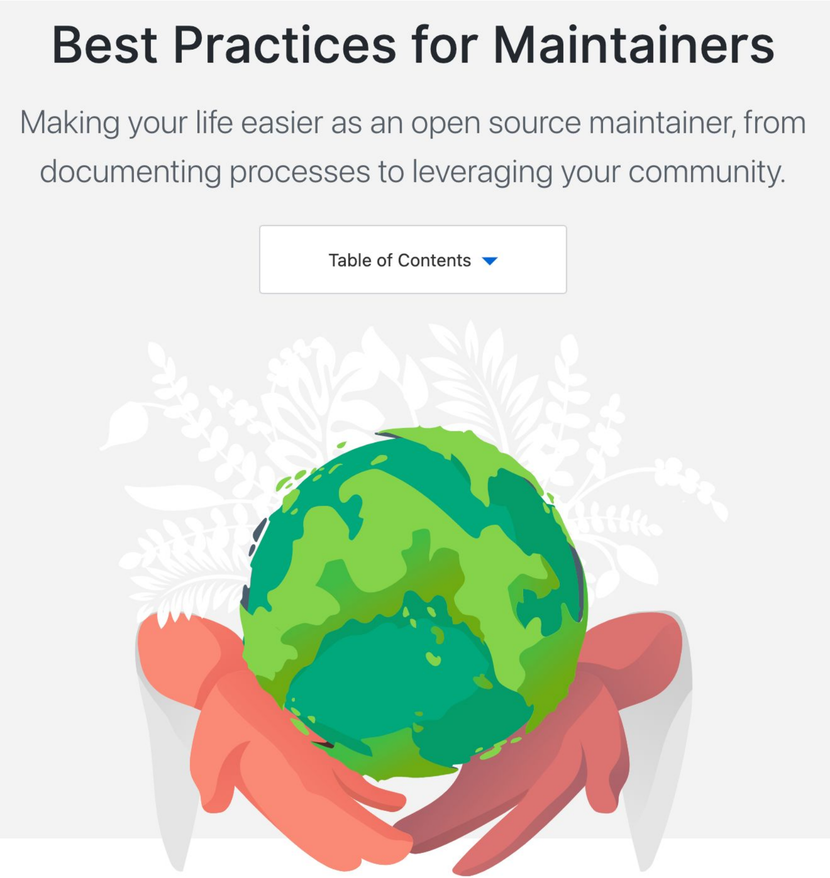
WP Engine forks from Wordpress, WP engine benefits from open-source platform and branding.

- WP Engine profits without giving back (financially, maintenance) to Wordpress.

Two challenges:

1) Imbalance → major contributors and minimal contributors.

1) Lack of supportive environment (users not reciprocating).



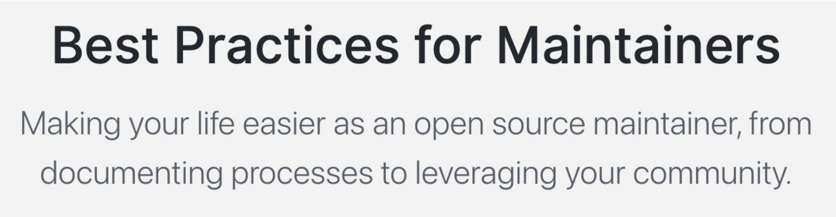
Best Practices for Maintainers

Making your life easier as an open source maintainer, from documenting processes to leveraging your community.

https://opensource.guide/best-practices/

Documenting process (develop community standards)

- getting it out of your head.



Best Practices for Maintainers

Making your life easier as an open source maintainer, from documenting processes to leveraging your community.

Communicate expectations (avoid wasted work)

- keep this public.

Learning how to say no (avoiding burnout)

- enforce rules, maintain project scope, maintain personal boundaries.

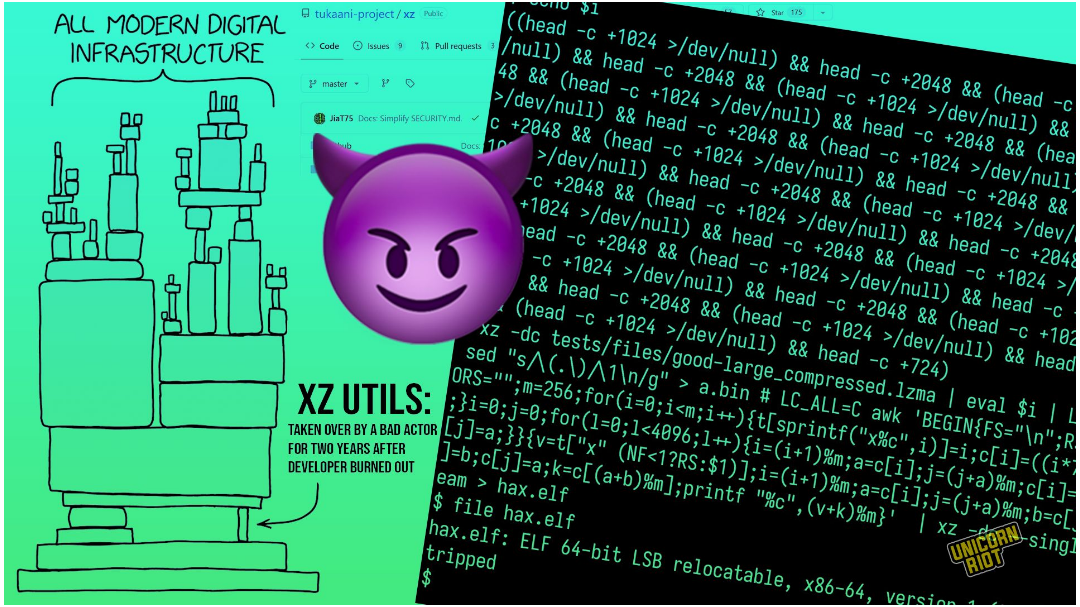
Leverage the power of community (avoid overreliance on key people)

- share the workload and fractional participation.

XZ backdoor and Autotools

‘xz utils’ Software Backdoor Uncovered in Years-Long Hacking Plot

https://unicornriot.ninja/2024/xz-utils-software-backdoor-uncovered-in-years-long-hacking-plot/


ALL MODERN DIGITAL INFRASTRUCTURE
XZ UTILS:
TAKEN OVER BY A BAD ACTOR FOR TWO YEARS AFTER DEVELOPER BURNED OUT
tukaani-project / xz
Public
Code
Issues 9
Pull requests 3
master
JiaT75 Docs: Simplify SECURITY.md.
Docs:
Star 175
UNICORN RIOT

```
echo $i
((head -c +1024 >/dev/null) && head -c +2048 && (head -c
/null) && head -c +2048 && (head -c +1024 >/dev/null) &&
48 && (head -c +1024 >/dev/null) && head -c +2048 && (hea
>/dev/null) && head -c +2048 && (head -c +1024 >/dev/null
c +2048 && (head -c +1024 >/dev/null) && head -c +2048 &&
>/dev/null) && head -c +2048 && (head -c +1024 >/dev/
-c +2048 && (head -c +1024 >/dev/null) && head -c +2048
+1024 >/dev/null) && head -c +2048 && (head -c +1024 >/
ead -c +2048 && (head -c +1024 >/dev/null) && head -c
-c +1024 >/dev/null) && head -c +2048 && (head -c +102
&& head -c +2048 && (head -c +1024 >/dev/null) && head
(head -c +1024 >/dev/null) && head -c +724)
xz -dc tests/files/good-large_compressed.lzma | eval $i | L
sed "s/\(.\)/\1\n/g" > a.bin # LC_ALL=C awk 'BEGIN{FS="\n";R
ORS="";m=256;for(i=0;i<m;i++){t[sprintf("x%c",i)]=i;c[i]=((i*
;}i=0;j=0;for(l=0;l<4096;l++){i=(i+1)%m;a=c[i];j=(j+a)%m;c[i]=
[j]=a;}}{v=t["x" (NF<1?RS:$1)];i=(i+1)%m;a=c[i];j=(j+a)%m;b=c[
]=b;c[j]=a;k=c[(a+b)%m];printf "%c",(v+k)%m}' | xz -dc --singl
eam > hax.elf
$ file hax.elf
hax.elf: ELF 64-bit LSB relocatable, x86-64, version
tripped
$
```

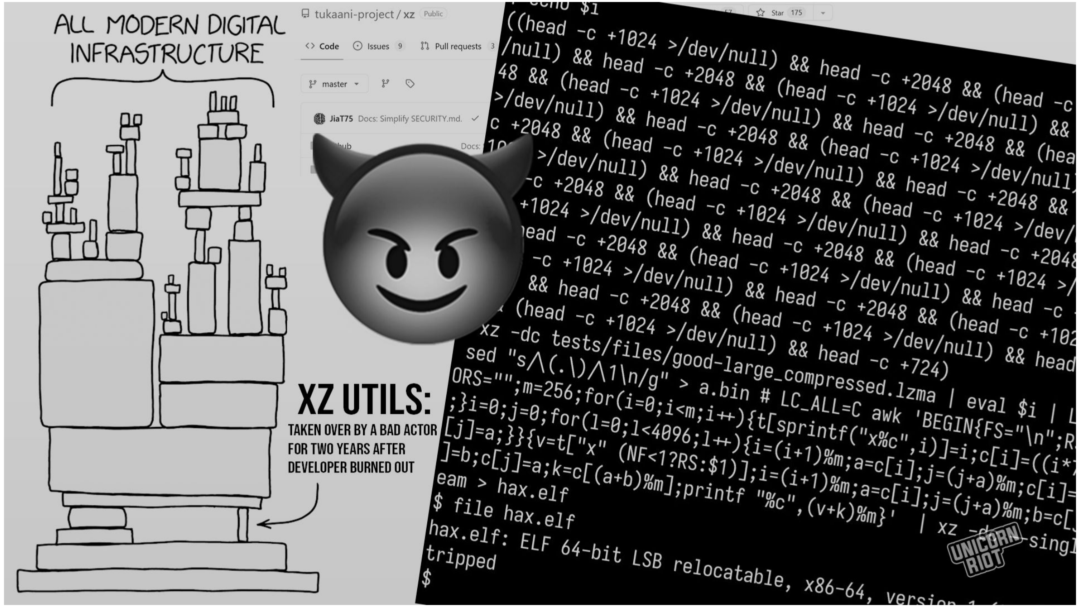
Maintainer burnout and package security

https://haacked.com/archive/2019/05/28/maintainer-burnout/

XZ backdoor and Autotools

‘xz utils’ Software Backdoor Uncovered in Years-Long Hacking Plot

https://unicornriot.ninja/2024/xz-utils-software-backdoor-uncovered-in-years-long-hacking-plot/


ALL MODERN DIGITAL INFRASTRUCTURE
XZ UTILS:
TAKEN OVER BY A BAD ACTOR FOR TWO YEARS AFTER DEVELOPER BURNED OUT
tukaani-project / xz
Public
Code
Issues 9
Pull requests 3
master
JiaT75 Docs: Simplify SECURITY.md.
Docs:
Star 175
UNICORN RIOT

```
echo $i
((head -c +1024 >/dev/null) && head -c +2048 && (head -c
/null) && head -c +2048 && (head -c +1024 >/dev/null) &&
48 && (head -c +1024 >/dev/null) && head -c +2048 && (hea
>/dev/null) && head -c +2048 && (head -c +1024 >/dev/null
c +2048 && (head -c +1024 >/dev/null) && head -c +2048 &&
>/dev/null) && head -c +2048 && (head -c +1024 >/dev/
-c +2048 && (head -c +1024 >/dev/null) && head -c +2048
+1024 >/dev/null) && head -c +2048 && (head -c +1024 >/
ead -c +2048 && (head -c +1024 >/dev/null) && head -c
-c +1024 >/dev/null) && head -c +2048 && (head -c +102
&& head -c +2048 && (head -c +1024 >/dev/null) && head
(head -c +1024 >/dev/null) && head -c +724)
xz -dc tests/files/good-large_compressed.lzma | eval $i | L
sed "s/\(.\)/\1\n/g" > a.bin # LC_ALL=C awk 'BEGIN{FS="\n";R
ORS="";m=256;for(i=0;i<m;i++){t[sprintf("x%c",i)]=i;c[i]=((i*
;}i=0;j=0;for(l=0;l<4096;l++){i=(i+1)%m;a=c[i];j=(j+a)%m;c[i]=
[j]=a;}}{v=t["x" (NF<1?RS:$1)];i=(i+1)%m;a=c[i];j=(j+a)%m;b=c[
]=b;c[j]=a;k=c[(a+b)%m];printf "%c",(v+k)%m}' | xz -dc --singl
eam > hax.elf
$ file hax.elf
hax.elf: ELF 64-bit LSB relocatable, x86-64, version 1 (
tripped
$
```

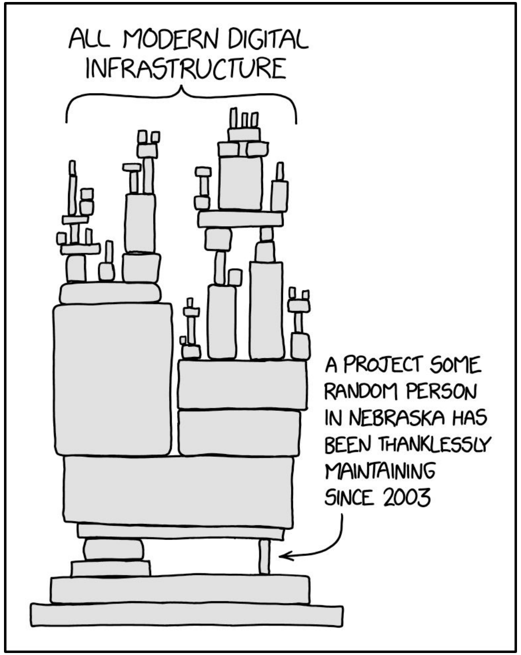
Pay the maintainers!!

Only depend on modules that you know are definitely maintained!

When you depend on something, you should take part in maintaining it.

Maintainer burnout and package security

https://haacked.com/archive/2019/05/28/maintainer-burnout/


ALL MODERN DIGITAL INFRASTRUCTURE
A PROJECT SOME RANDOM PERSON IN NEBRASKA HAS BEEN THANKLESSLY MAINTAINING SINCE 2003

“I haven’t lost interest but my ability to care has been fairly limited mostly due to longterm mental health issues but also due to some other things. Recently I’ve worked off-list a bit with Jia Tan on XZ Utils and perhaps he will have a bigger role in the future, we’ll see”.

“Xz-utils versions 5.6.0 and 5.6.1 were found to have the malicious programming added by “Jia Tan” (jiaT75) on GitHub. It got into the widely used Debian testing (pre-release) branch but not the stable branch. It also made it into the widely used OS X third-party package manager homebrew. Some Fedora Linux 40 systems (as well as Rawhide, the pre-release for Fedora 41) may have gotten the malicious software”.