The legal side of the POSS world

[sort of]

Richard Fontana
OSCON 2013
This talk has nothing to do with Marten Mickos.
Deconstruction of a tweet

monkchips
@monkchips

younger devs today are about POSS - Post open source software. fuck the license and governance, just commit to github.

11:34 PM - 16 Sep 12
Licensing

Stephen Walli @stephenrwalli
@monkchips Governance implies community but promiscuous sharing w/out a license leads to software transmitted diseases

Dave Neary @nearyd @stephenrwalli @monkchips The answer to "but how do they manage <problem X>" is typically "<problem X>? Get with the times, paps!"

Stephen Walli @stephenrwalli @nearyd @monkchips How about "#Github w/out licenses is like a 'free lovefest' w/out condoms." /c @mmilinkov
Déjà vu all over again

Jason Hibbets
@jhibbets

It's getting hot in the GitHub session, audience comments include "ban GitHub" and "Github is a threat to the open source movement" #osbc

4:38 PM - 22 May 12

Jason Hibbets
@jhibbets

For context of my last tweet, it's about the issue that there are no default license for projects on Github #osbc

4:46 PM - 22 May 12
The intelligentsia responds

- Legal-formalist alarmism; irresponsible GitHub, ignorant kids
- Bad for business → bad for open source
- Association with broader free culture legal trends

Common assumption: developers *intend* code to be open source
GitHub needs to take open source seriously

- "Under copyright law, code without a license cannot be legally shared"; danger of future © infringement litigation

- GitHub growing platform by appealing to younger devs' "licensing is for losers" sensibilities
Notices under US copyright law

- 1909 Act: publication without notice = automatic public domain
- 1976 Act: © attaches automatically on creation
- 1978-01-01 - 1989-03-01 (pre-Berne): publication without notice = public domain unless effort to correct within 5 years
- Berne (present): absence of notice has no effect on copyright protection
Law in books vs. law in action

- If "unlicensed code" is ARR, why treat it as an open source problem?
- If we assume developers want code to be open source but just don't know how to do it "properly", copyright infringement risk here must be low.
- No reason to think "unlicensed code" is most significant liability risk for users of GitHub-hosted code.
Implied copyright license?

- May arise from circumstances creating reasonable expectation that © holder intends work to be used for some purpose

- For GitHub-hosted code, decent argument for implied broad license to copy, prepare derivative works, and distribute *for development purposes*

- But with "no-declared-license" code, can't necessarily easily get to 'implied fully-OSD-conformant license' without special facts
GitHub's statements don't help here

- Doll: on GitHub, absence of explicit license rightly interpreted as ARR
- ToS: "By setting your repositories to be viewed publicly, you agree to allow others to view and fork your repositories"
Lack of license a "disturbing trend"

"ensures software will be removed from any product meant to be used by corporations", undercutting goal of broad code reuse
'Only benefits lawyers' corollary

Luis Villa, Licensing confusion is great! (for lawyers)
Shavianism taken to an extreme?

- "The last thing a startup wants to admit is that they don’t own their intellectual property"
- "No reason to give them unrestricted use of my software since they are only interested in turning my software into a hot IPO 2–5 years from now"

Why I (A/L)GPL (2009)
Pushing back against licensing and the permission culture

- MIT probably most common explicit GitHub license
- Recent PDD instruments: CC0 (2009), "The Unlicense" (2010)

- Villa: POSS as implicit critique of "open license ecosystem" assumption: sharing can't/shouldn't be done without explicit permission
Some firsthand experience

Lots of GitHub repositories are trivial, de minimis
Lawyer discussion

(Notice we're both concerned about problem of ambiguity)
Evidence?

• Williamson: via FOSSology, 16.6% of 28% oldest GitHub repos have licensing information at **top level** (widely misreported by sensationalists) *(Licensing of Software on GitHub: A Quantitative Analysis)*

• Berkholz analysis of Ohloh data: POSS is largest license trend in absolute market share; associated with smallest projects (which tend to be on GitHub) *(The size of open-source communities and its impact upon activity, licensing, and hosting)*
Why assume this is new/GitHub-specific?

- Palamida, Black Duck have claimed so
- Google Code, SourceForge licensing requirements?
- GitHub popularity exposed phenomenon of "no-explicit-license seemingly-FLOSS code" to the previously innocent
- Usenet newsgroups for "public domain sources" (1980s to mid-1990s) - is "unlicensed code" uninterrupted free software tradition?
Overlooked: GitHub facilitates transparent, collaborative legal improvements

- Many users submit "add a license" issues/PRs (sometimes corporate or Linux distro concerns)
- Project creator often apologetic and seeks education/consensus
- In many cases a license *is* present but non-obviously presented!
- License selection is simply being "delayed"?
- Not just "ignorant kids"
Typical example

Users empowered to participate in license selection
Consistent with Berkholz conjecture:

"as projects grow, they tend to sort out any licensing issues, likely because they get corporate users, professional developers, etc."
What GitHub did last week
choosealicense.com

Which of the following best describes your situation?

I want it simple and permissive.
The MIT License is a permissive license that is short and to the point. It lets people do anything they want with your code as long as they provide attribution back to you and don’t hold you liable.

jQuery and Rails use the MIT License.

I'm concerned about patents.
The Apache License is a permissive license similar to the MIT License, but also provides an express grant of patent rights from contributors to users.

Apache, SVN, and NuGet use the Apache License.

I care about sharing improvements.
The GPL (V2 or V3) is a copyleft license that requires others who modify your code to disclose their changes if they redistribute it in source or binary form. V3 is similar to V2, but further restricts use in hardware that forbids software alterations.

Linux, Git, and WordPress use the GPL.

What if none of these work for me or I want to retain all of my rights? More licenses are available.

What if I don’t want to choose a license? You don’t have to.
choosealicense.com/licenses/

- "Featured": Apache 2.0, GPLv2, MIT
- "Other licenses": MPL 2.0, LGPLv2.1, 3-clause BSD, Artistic 2.0, GPLv3, LGPLv3, AGPLv3, "Public Domain (Unlicense)", "No License", EPL, 2-clause BSD
- CC-deed-reminiscent descriptions containing annoying errors
- Website is MIT-licensed; can fork and make pull requests
No License

Copyright [year] [fullname]

How to apply this license
Simply do nothing, though including a copyright notice is recommended.

Note: This option may be subject to the Terms Of Use of the site where you publish your source code.

Required
- License and copyright notice

Permitted
- Private Use
- Commercial Use

Forbidden
- Modification
- Distribution
- Sublicensing
No License

You're under no obligation to choose a license and it's your right not to include one with your code or project. But please note that opting out of open source licenses doesn't mean you're opting out of copyright law.

You'll have to check with your own legal counsel regarding your particular project, but generally speaking, the absence of a license means that default copyright laws apply. This means that you retain all rights to your source code and that nobody else may reproduce, distribute, or create derivative works from your work. This might not be what you intend.

Even in the absence of a license file, if you publish your source code in a public repository on GitHub, you have accepted the Terms of Service which do allow other GitHub users some rights. Specifically, you allow others to view and fork your repository.

If you want to share your work with others, please consider choosing an open source license.
Governance

monkchips
@monkchips

younger devs today are about POSS - Post open source software. fuck the license and governance, just commit to github.

11:34 PM - 16 Sep 12
A decade of management research

- All open source projects have 'governance' (they all make decisions); licensing initially seen as key element

- Wasserman & Capra (2007): 35% of community projects sampled "not managed at all" (= issues are decided via mailing lists)
When Free Software Isn't Better (2010)

SourceForge & Savannah studies: most free software projects (even actively-released and frequently-downloaded) are single developer with little or no outside help

(Consistent with Berkholz in 2013 Ohloh study)
In open source, *normative* governance no older than ~2011

- Phipps, *Open By Rule* (Feb. 2011) (software freedom requires "community equality guaranteed by fair governance")

- JimJag, *Lessons Learned in Open Source at Apache* (Oct. 2011) (3 governance types cover all projects; implies ASF meritocracy is best)

Apache Considered Harmful (Nov. 2011)

- GitHub is anarchic, decentralized, unmediated developer-to-developer communication system, imposing no "set of governance or process"
- Fork possibility creates "social imperative to manage contributions in a satisfactory manner"

Probably what Governor meant, and what 'project governance value-add' foundations see as a threat
Some quick general conclusions

- I don't think there is any 'POSS'
- 'Unlicensed code' has been around forever (though may be increasing if shift away from copyleft); mostly not a big deal
- Main problem is ambiguity for commercial users, but GitHub provides means of addressing
- Apache/Eclipse-style 'good governance' is a new marketing angle, irrelevant to most projects
Thanks!

@richardfontana