

# What is involved in open sourcing proprietary software? [closed]

Asked 15 years, 11 months ago   Modified 3 years ago   Viewed 398 times



7



**Closed.** This question needs to be more [focused](#). It is not currently accepting answers.



**Want to improve this question?** Update the question so it focuses on one problem only by [editing this post](#).

Closed 9 years ago.

[Improve this question](#)

My primary motivation for asking this question is [this uservice suggestion](#). Jeff declined the ticket to make the SO software open source saying that it will take more time.

I've seen this before in various other pieces of software that have gone from proprietary to open source. So, my question is: **why does it seem to take so long to make software open source?** To me, it seems pretty simple: put your code on sourceforge and google code and be done with it. But there's obviously something that I'm missing in the whole process.

(And before anyone gets the impression that I'm trying to be critical of Jeff or anyone else that delays open sourcing their software, I'm not. I just want to get an understanding of the process for open sourcing stuff and its costs.)

language-agnostic

open-source

Share

Improve this question

Follow

edited Nov 23, 2021 at 11:55



Glorfindel

22.6k ● 13 ● 89 ● 116

asked Jan 4, 2009 at 15:58



Jason Baker

198k ● 138 ● 382 ● 520

This question appears to be off-topic because it is about licences and the law. – [Raedwald](#) Dec 4, 2013 at 0:03

5 Answers

Sorted by:

Highest score (default)



5



The main answer to your question has been given by others - getting the legal permission to do so (see the SO blog entry on [Reverse Engineering the WMD Editor](#) for an SO-related problem child) is often extremely difficult, even impossible.





So, my question is: why does it seem to take so long to make software open source? To me, it seems pretty simple: put your code on Sourceforge and Google Code and be done with it. But there's obviously something that I'm missing in the whole process.

What you describe - dumping the source - is not really Open Source. It is more akin to either AbandonWare or perhaps 'Available Source'. An Open Source project needs to accept inputs from outside, and build a community. One of the criteria that the Apache Software Foundation uses for its incubator projects is "has it acquired a critical mass of contributors from outside the original authors?". This is a valid concern.

Note that neither AbandonWare nor 'Available Source' is necessarily bad; both make code available that would otherwise not be available (and provide some of the benefits of Open Source). But there is more to Open Source than that.

Also, there is management overhead in handling a truly Open Source project. That is not negligible.

And, finally, it is not unheard of for the code quality to be such that the authors would rather not make the source available for fear of ridicule. I doubt if that applies in this case, but it can in other areas of the software world.

Follow

edited Jan 18, 2021 at 12:38



Community Bot

1 ● 1

answered Jan 4, 2009 at 16:55



Jonathan Leffler

752k ● 145 ● 946 ● 1.3k

---

1 lol, the last bit is true. If you looked at our codebase you'd see the odd comment like: "this is f\*\*\*ed, needs a rewrite ASAP! - Jim, 10/04/92" :) – [gbjbaanb](#) Jan 5, 2009 at 15:49

---



5

Usually it requires all contributors to agree, for some projects that means you have to ask quite a few people if its ok to open their code too.



For other projects, it requires the libraries you've used to be either open source, or taken out of the project where you've used them. IIRC Sun said they couldn't [open Java](#) for so long because they used some 3rd party code that couldn't be opened.

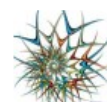


Share Improve this answer

edited Jan 4, 2009 at 16:11

Follow

answered Jan 4, 2009 at 16:03



[gbjbaanb](#)

52.6k ● 12 ● 110 ● 154

---



3



Opening the source could be good or bad. In the best case, they only benefit from contributions from the community, retaining creative control. In the worst case, the project could fork, and two differing codebases are not usually beneficial to the community.

In SO's case, they may be worried that if enough people are impatient, then opening the source too early is inviting a fork of the project as other developers implement their "pet" features more quickly (or differently) than SO. Maybe those people really want features that have been declined.

Forking also brings in the aspect of competition that Shawn mentioned. Perhaps much later when SO has a more well-established community, opening the source would be more acceptable as SO would have a head start on any potential competition.

The conservative thing to do is to wait a bit before making a decision, rather than opening now and hoping for the best.

**EDIT:** Note that even in the best case, they are taking on the added role of managing an open-source project and dealing with whomever wants to contribute. The "herding cats" methaphor comes to mind, they may want to focus on the codebase right now rather than administrative overhead.

Follow



Adam Bellaire

110k ● 19 ● 152 ● 165



3



The reasons are:

1. Are you sure you have the right to make software open-source?
2. And do you **really** want to?



Regarding the first question, note that you may be using third-party code, so you have to go through all your code making sure you have the right to make it all open-source.

In second place, it requires the project owner to want to open-source the project. In some cases, he/she may think it's not worth it.

Share Improve this answer

Follow

edited Jan 4, 2009 at 16:44



Jonathan Leffler

752k ● 145 ● 946 ● 1.3k

answered Jan 4, 2009 at 16:04



luiscubal

25.1k ● 9 ● 59 ● 84



1

Here are some factors to consider before open sourcing:

1. Are you legally able to make the code open source?  
Due to libraries, other contributors, etc.



2. Do you have a business model / Do you need one?  
In projects like SO or Digg, where the edge is mainly in the community, then it should be OK, but in other projects that gets deployed inside corporate firewalls making it OS reduces the ability to make money to cover your costs. Although some companies do that via support, however, the ability to get support costs depends on the nature of the product and how critical it is to the companies deploying it.
3. What is your feel of the community? There are tons of open source projects that have 1 or 2 contributors and not many users contributing patches. If that is the case, then the overheads may be larger than any benefits.
4. Are you ready to support others deploying your product in other verticals or even the same vertical? This means forums, mailing lists, bug trackers, easy to use upgrade scripts, documentation, etc.

For a project to be successful as open source, it has to be well managed. Putting the code out there is simply not enough. You will just end up with unhappy community and potentially a fork on which you have no control.

Share Improve this answer

answered Jan 4, 2009 at 17:25

Follow



vboctor

905 ● 5 ● 9