Group Project

Professor Mauger

CST-341

August 23, 2020

Open Source Licenses- Milestone 1 Research

1. **What are three different open source licenses? How do the licenses differ? How are the licenses the same?**

The Apache License, The GNU "General Public License," and the MIT License are all examples of popular open source licenses.

The Apache License "...works well for organizations or projects that are larger and managing more contributors, but don't care about others commercializing the work" (Morris, 2016). It is a great option if you are concerned about outsiders "encumbering" your software project, as the specificity in its licensing agreement ensures that there is a patent license attached to the project as well.

The GPL License emphasizes that you must "...provide the source code under the license" (Morris, 2016), if distributing a derivative of a specific project. Therefore you must be careful when you contribute to projects and then distribute them as you must ensure that you are "citing your sources."

With the MIT license, a big advantage is that "...more people may end up using your software" (Morris, 2016). It's more of a free-for-all open ended license than some of the other ones. A disadvantage that some people see with the MIT License is that you do have to be okay with people taking your work, expanding on it, and distributing it without explicit permission.

All of these licenses allow others to work on open-source projects somewhat freely, but there are differing levels of restrictions that come with each one, so it is important to keep that in mind.

1. **What happens and who owns the code you contribute to an open source project?**

"Under copyright law, the author of a line of software code is the owner of the copyright in that code" (Pelton, 2008). This means that when you are developing software, any line of code you type is legally your own. However, open source software complicates things, as licenses become involved, but the basis remains the same. When you create open source work using a license, you still own the code, but sometimes, others who work on the project can also distribute the software that includes your code.

There are not always clear answers as to who can distribute what part of a project and when. Often, all members of a party who worked on a project may need to get explicit permission from everyone else in the party in order to distribute the project. That being said, it is indisputable that the lines of code you write are your own.

1. **Research an open source project from the Apache Foundation and provide a detailed write up for how you can volunteer and contribute to one of their projects.**

The Apache Foundation has several project communities that a programmer can join that will allow them to contribute to the Apache Software Foundation. There are “…over 140 different projects, with many new project submissions…coming into the Apache Incubator every year” (Apache Software Foundation, 2020). The best way to figure out how to contribute to the Apache Foundation is to join one of their mailing lists. One should “…use mailing lists to coordinate development of the software and administration of the organization. Mailing lists also serve as a primary support channel where users can help each other learn to use the software” (Apache Software Foundation, 2020).

Once a contributor “…has has shown sufficient sustained commitment to a project by helping out and contributing work to the project (and the ASF) [they] may be voted in by the project as a committer” (Apache Software Foundation, 2020). Becoming a committer will open the gateway for a programmer to be able to shape the future of a project.

An example of one of these open source projects that has been worked on by many people over a long period of time is a project called Apache Ant. The Apache Ant’s mission is “…to drive processes described in build files as targets and extension points dependent upon each other” (The Apache Ant Project, 2020). It is mainly used to build Java applications, but what makes it unique is that “users of Ant can develop their own ‘antlibs’ containing Ant tasks and types, and are offered a large number of ready-made commercial or open-source ‘antlibs’ (The Apache Ant Project, 2020).

Apache Ant is a project that has been in the works since 2006, and the programmers who are working on it just released a new version of it in May of this year. There are probably still several bugs and errors that could potentially be fixed by new contributors to the project, and there’s still always room for improvement and innovation within Apache Ant. In order to be able to contribute to this project, a programmer should join the Apache Ant mailing list, just like with all other Apache projects.

There are many different projects within the Apache Foundation, and as long as a programmer is committed to helping out by identifying and working out bugs, they will become a valuable member of the project.

References

The Apache Foundation. (n.d.). Community-Led Development "The Apache Way". Retrieved August 24, 2020, from https://www.apache.org/foundation/getinvolved.html

MacNeill, C., & Bodewig, S. (n.d.). The Apache Ant Project. Retrieved August 24, 2020, from https://ant.apache.org/

Morris, J. (2019, October 11). Which License Should I Use? MIT vs. Apache vs. GPL. Retrieved August 18, 2020, from https://exygy.com/blog/which-license-should-i-use-mit-vs-apache-vs-gpl/

Open Source Initiative. (n.d.). Licenses & Standards. Retrieved August 18, 2020, from https://opensource.org/licenses

Pelton. (2008). Who Owns the Code? Retrieved August 21, 2020, from https://asp-software.org/www/misv\_resources/business-articles/who-owns-the-code/