Free and Open Source Software Licensing - June 18th, 2020

Groups exercises

Groups 1, 4 – Exercise A

Person(s) involved: 3 software developers (colleagues)

One of them creates a basic software and also collaborates with his colleagues in developing a very particular program.

The program is based on a GPL licensed software.
They want to adapt that program for the company where they work.

After several years, one of them doesn't work in that company anymore and he is really against the FOSS licenses. He wants to sell that program under a proprietary license. Is he allowed to do that?

*You have **2'** to briefly explain us:*

- *Did you create a new software or a derivative work?*
- *If derivative work, under what license was the earlier work?*
- ***What license** would you choose for your version of software? **Why**?*

Solutions

"Basic software" in the sense of a software lacking individuality: it is not considered a "work" as defined by art. 2 of the Swiss Copyright Act. Thus, there is no copyright protection on this software.

"Very particular program" can be considered a work protected by copyright. The colleagues that develop this program are coauthors of the new software.

GPL licenses are strong Copyleft licenses. The adapted version is a "derivative work" under the copyright meaning. The colleagues are allowed to use and adapt the program for the company they work for. As the adapted version is not distributed, there are no license problems to consider in this case.
As the adapted version is based on a GPL license software, it is not possible to distribute it under a proprietary (non FOSS) license.

Group 3 – Exercise B

Person(s) involved: A friend and a software developer

A guy asks his friend programmer to develop a software and he tells him all the details of how the software should be. Unfortunately, he's not an informatics engineer himself.

The program has been distributed under the GPL v2 license.
Several years later, the software developer finds a modified version of the program he developed. He's afraid nobody knows what he wrote. Does he get credit for his work? Can he modify it again?
Is he allowed to require that anyone who receives the copy of the software must pay him a fee or notify him?

*You have **2'** to briefly explain us:*

- *Did you create a new software or a derivative work?*
- *If derivative work, under what license was the earlier work?*
- ***What license** would you choose for your version of software? **Why**?*

Solutions

If the software developer merely executes the client's orders and it is not possible to recognize his individual, original, contribution in creating the software, the developer is not the author as defined by the Copyright law. On the opposite, if the software developer puts his individual contribution in developing his friend's idea, he is the author and he should expressly assign the economics rights (right to use, ect.) to his client (it is preferred with a written contract). If he *wants this software to be free and open source, he must grant a FOSS license (which must be attached to the program) so to authorize users to use, study, modify and distribute it.*

The software developer created a new work. The modified version he found later on is a derivative work based on his work (or on the earlier work). As the earlier work is under a GPLv2 (strong copyleft) license, the next works based on that one must be distributed under a GPL license as well (GPLv2 or GPLv3).
He doesn't need to worry about his credit: when redistributing a Free and Open Source software, it is required to add in the copyright and license notice attribution (authorship) as well as the statement that the modified version is actually a modified version.
He can modify again the program and redistribute it under a GPL license.
He can sell copies of the source code for he is the author and he should expressly assign the economics rights (right to use, ect.) to his client (it is preferred with a written contract).
money, but he's not allowed to claim money for the royalties.
Receivers of the licensed software don't need to notify him that they received and are using / redistributing it.

Groups 2, 5 – Exercise C

Person(s) involved: Freelancer software developer and his client

The freelancer software developer creates a program for his client.
The client just gives him an idea of how it should be.

The client wants this program to be open and he thinks it will be useful in a commercial field.
But he forgot to include a copy of the license with the software.
Later on, another software developer receives the copy of the object code (and the copy of the license), and wants to combine it with a code of a library software. Has the whole new version to be under the same license?

A user of the program is not happy with it and wants to sue the freelancer developer. Can he do that?

*You have **2'** to briefly explain us:*

- *Did you create a new software or a derivative work?*

- *If derivative work, under what license was the earlier work?*
- ***What license** would you choose for your version of software? **Why**?*

Solutions

The freelancer software developer is the author and he should expressly assign the economics rights (right to use, ect.) to his client (it is preferred with a written contract). When he and his client want this software to be free and open source, the software developer must grant a FOSS license (which must be attached to the program) so to authorize users to use, study, modify and distribute it.
If a software is distributed without any FOSS license attached, it can't be considered a free and open source software. A software can be used, studied, modified and redistributed under FOSS license's terms only if the appropriate license is attached with its whole text, so that users/receivers know what they can or can't do with that software.
Later on, considered that the original software has a copy of the FOSS license attached, the software developer needs to ask the licensor for the source code (the object code is not modifiable), and the licensor must provide it.
When combining code from different licenses, often the most restrictive of the licenses has to be used for the final product. It depends on the licenses used: there are licenses that are incompatible, other are compatible. For example, combining a software library with a larger proprietary program can be done with a LGPL (weak copyleft) license, which is a compromise between permissive and strongly protective licenses.
When programs are licensed under a FOSS license, there is no warranty for the program, therefore a user can't sue the program developer in this regard.