Opportunities and Pitfalls of GPL License Agreements from the Perspective of the Software Developer

Tihomir Katulić
Faculty of Law
University of Zagreb
Trg maršala Tita 14, Zagreb, Croatia
tihomir@pravo.hr

Abstract. *Permanent information revolution implies continued development of sophisticated information systems. Software development continues to be one of the fastest evolving aspects of information technology. Developing ever more complex software presents not only technical and economical but also considerable legal difficulties.*

Through self-regulation industry specific contracts and standards have emerged with the goal to define legal aspects of developing complex software. The GNU GPL license agreement is a broadly accepted standard with important repercussions for both the users as well as software developers.

Keywords. Information revolution, software, copyright, GPL

1 Introduction

Legal regulation of software as a copyright-protected work has caused certain effects not encountered in the usual course of commercial exploitation of other cathegories of works.

Software development, either by an individual programmer or by a group of programmers is traditionally an undertaking in which the authors improve and add to the work until the moment of publishing. Often enough, software is developed by employees of a company [3]. In such circumstances, autorship and other issues concerning the legal state of software are often regulated by contracts [14, 43] as well as positive legal framework [8, 807].

Autorship attribution issues can arise when the authors (software developers) cooperating on the development of computer program neglect to formalize their participation. This often occurs when developers participate in a project in their own spare time, out of a hoby or a learning activity.

The participating developers need not share the same opinion regarding the use of the software, approve design choices for the program user interface or other technical issues concerning the software development. This informal, collaborative development effort is, in reality, a widespread activity that is facilitated and encouraged by the advancement of information technology as well as its popularization. Millions of computer programmers, user interface designers and other software developers develop software without legally formalizing their relations. Distinguishing their individual contribution is almost an impossible task. Different forms and modes of development are constantly interchanged in the fierce struggle and competition present in the IT industry [5, 16].

In that regard, software development as a creative process and software as an object of copyright, a work protected by copyright, significantly differs from other protected works. One could argue that the utilitarian nature of software diverts from the philosophical grounds on which the system of copyright is founded [4, 29].

While collaboration of authors is often encountered in the creation of the works of art, software development is not limited by physical presence of authors or the need to meet, communicate or apply a common design policy or development decision. Development of a computer program can be a concentrated effort of many hundreds or even thousands of computer programmers that may, or may not follow certain general guidelines communicated by the software project homepage or by other means. There are several representative examples of these sites, first and foremost SourceForge and Freshmeat. One could also argue that developer communities built around Linux kernel, various Linux distributions and FreeBSD represent a similar model. All these developers are able to work together on their projects without negotiating or even getting to know everyone involved. This is facilitated by the formal license agreements such as the GNU GPL.

In contrast to the more traditional cathegories of protected works authored by one or a handful of authors, software projects can have several hundred or

even several thousand developers. Researchers from the Massachussetts Institute of Technology have described Open Source projects exceeding over three thousand active developers [12, 116].

It is not uncommon that authors develop their own versions (*forks)* of software, and for the software projects to be divided into indenpendent developments that retain a degree of common code from earlier stages of development. For example, when the code of the Netscape Navigator browser was made available to the public through application of Mozilla Public License, this formerly commercial software became a basis for development of numerous open source projects including the popular Firefox browser as well as browsers like Flock, IceWeasel, Songbird etc.

This behaviour is not specific only to pure software industry. Considering the profound impact of information technology on all fields of technology, it is safe to conclude that in one form or the other software permeats all manners of devices and machines being used and developed today. From television sets, mobile phones and entertainment devices to cars, airplanes, ships, heavy machinery and machine tools software is an essential ingredient that makes these products smarter and more effective. Many of these products and tools have been equipped with software developed by a heterogenous group of developers collaborating over the Internet, often under the broad and unclear legal terms and conditions.

These machines may often use embedded software developed by tens or hundreds of software developers that have never met or signed a legal binding agreement. The regulation of rights and obligations between these contributors, all of them authors in their own right and in relation to the code contributed to the protected work, is a real challenge for the traditional approach to rights management. Even when it is possible to regulate the rights between these co-developers and towards users and anyone else through traditional copyright licensing agreements, it is never economical or practical.

Certainly the fact that these works are created under the conditions and via technology that could not be forseen or taken into account when current, globally accepted and applied legal framework concerning copyright was being considered does not mean these works are not elligible for copyright protection.

Indeed, the copyright protection of computer programs has been a legal standard since later twentieth century. In 1980. US Copyright Act was the first national copyright law that introduced copyright protection to computer software [9,619]. Earlier efforts by the WIPO considered a *sui generis* right similar to the one concerning the structure and organisation of databases. It is these new conditions and development of information technology that have facilitated the need to develop software industry specific lincensing agreements, a kind of formular licensing agreements that fit the needs of developers and users in the digital environment more than the traditional copyright licensing agreements ever could. These new licensing schemes and their broad acceptance in the software industry over the last thirty years require a more in-depth analysis into their structure, provisions and solutions and the benefits and caveats that may arise from their usage.

2 *Free software* and GNU public license

While the Creative Commons [2] set of licenses can also be used to manage rights over computer software, the fast development of software industry has produced a number of other initiatives meant to regulate the use of this specific cathegory of copyright-protected works. In this regard, we consider software to be both computer programs as well as instruction and function libraries and other programming objects.

The GPL, or the GNU Public License is arguably the most popular and most often used formular licensing agreement in the field of software rights management. The author and leading proponent of the GPL, American software developer and computer scientist *Richard M. Stallman* has envisioned the GPL, written in 1985. [1] and publicly introduced in 1989., as a means of fostering an alternative to traditional legal protection of software development, distribution and commercial and non-commercial exploitation.

This was achieved through provisions that stipulate rights holders waiver of commercial exploitation rights, most notably the right to prevent unautorized distribution and reproduction, as well as the authors commitment to distribute together both the source and the executive code of the software, as well as the full text of the GPL. These provisions would ensure the users right to adapt the software to their own particular needs as well as to be able to tell what they may or may not do with the GPL licensed work.

The later versions of the GPL were co-developed by Stallman and several distinguished legal scholars (such as prof. Moglen from Columbia University, the founder and the director of the Software Freedom Law Center) gathered around the *Free Software Foundation*, a non-profit organization committed to promotion of free software and further development of the GNU project. This paper is referring specifically to the current, third version of the GNU GPL that replaced the long-lived second version [11,130] and that has been made public relatively

recently (in 2007.) spurred on by new technologies and methodologies in software development (SaaS, software as a service) as well as regulatory progress primarily through widespread adoption of TRIPS, WIPO Internet Treaties and the EU Directives.

3 Impact of GPL on the position of the software developer

Basically, the GNU license is an unorthodox form of a copyright license agreement that allows and binds users and/or future co-developers of the object of license, that being software (as mentioned earlier, in the form of computer programs, instruction libraries or other digital objects). The users and/or co-developers can use (distribute, copy, modify) the software under certain conditions presented as software *freedoms* that the user of software should possess in order for the software to be considered free. Those freedoms are the freedom to use the software for any purpose, the freedom to adapt software according to users needs and desires, freedom of further non-commercial software distribution and the freedom of the user to distribute the versions of the software adapted from the original program.

From the user's perspective, these freedoms constitute a broad license regarding the use of the work.

However, from the perspective of the software developer, these freedoms are actually serious limitations regarding the exploitation of his exclusive rights. These limitations extend, as stipulated by the GPL Agreement, to all participating authors and follow the use of the work until copyright expires.

In the sense of commercial exploitation, software published under the GPL license (not to be confused with LGPL, or the *Lesser GPL*) can no longer be an object of certain by positive law regulated exclusive material rights, such as a right to authorize distribution and reproduction, as well as moral personal rights (such as the right to prevent modification of the protected work).

By accepting the terms of the GNU GPL the author has irreversably waived his or her exclusive rights guaranteed by the positive legal framework (international conventions such as the Berne Conventions, WIPO Treaties, WTO Trips, EU Directives as well as national positive law, such as Croatian Copyright and Neighbouring Rights Act - CNRA). Fundamentally, the same terms that allow the user to access, use and modify the GPL licensed software allow the same for the future users and co-developers and prevent (all) rightsholders from legal intervention.

The developer who wants to use the existing source code as a basis for further code development can do so, under the obligation to allow others to do the same with the programs or code his development of the existing GPL licensed code results in. The waiver of material, commercial exclusive rights does not mean that author(s) cannot charge for the service of distribution via data carrier mediums or an internet service, or by writing manuals, documentation or offering consulting services.

In comparison with the existing copyright model and terms and conditions of traditional copyright licensing agreements, and even with regard to the Creative Commons formular licensing agreements it is obvious that accepting the terms and conditions of the GNU GPL will have serious ramifications for the author and for the future use and development of the thus licensed work. The GPL license seriously hampers the author's ability to influence the future of his work, allowing only for two basic rights. The first is the most basic moral right concerning copyright – the paternity right (the right to be mentioned as an author of the work, as regulated by the Croatian CNRA, Article 15), as well as the right to file a civil suit against anyone who might use the work against the stipulations of the GNU GPL.

Acceptance of the terms and conditions of the GPL license also implies the author has waived remaining moral rights, such as those regulated by the articles 16. and 17. of the Croatian CNRA as well.

From the standpoint of the content creator, in this case the software developer, the frequently asked questions regarding the application of the GNU public license concern the reasoning behind waiving the exclusive rights provided by the positive regulation.

Indeed, why would a rightsholder relinquish any of his rights or accept the restrictions imposed by the GPL regarding the enforcement of rights already acknowledged by the statutory framework? What possible motive could an author have in giving up his moral rights? Finally, what is the legal status of the GNU GPL – is this licensing agreement even enforceable in court [10,1]?

4 Why would rightsholders accept the limitations imposed by the GNU GPL?

Since GNU GPL basically allows for free copying and distribution of software, the rightsholder who has accepted the terms and conditions of the GNU GPL has no legal recourse in obtaining a financial reward from users who accept the use of the protected work under the provisions of GNU GPL. In other words, the distribution and reproduction of protected software is free, and there is no legal ground on which

the author can base his petition against the user in this regard.

Considering the success of software industry and the financial rewards gained by the software developers this limitation seems paradoxical and unfair. What possible motive could a hard working software developer have in waiving his right to fair financial compensation for licensing his work that may have taken years to develop?

The answer is contained in the principles of the Free software movement, which by now are well known to both information technology professionals as well as ordinary computer users. The Free software proponents maintain that commercial software, that is to say proprietary, closed source software is dangerous to user privacy, innovation and competition. By allowing users to freely access the application source code free software guarantees insight into program funcionality, and reveals potential bugs, errors and security oversights. Users with the proper technical knowledge can spot *backdoors*, analyze data flow, detect provisions allowing remote access and surveillance of user's habits and information systems. Since commercial software is usually distributed only in the form of executable code, there is no practical way for users to check the source code of the program for hidden funcionality, bugs, backdoors and other errors and oversights. Modern software may containt hundreds of thousands of lines of code and, in the case of operating systems, even millions of lines of code effectively preventing the end user to be sure about what commercial programs really do

While the benefits for users are clear, the case for rightsholders is far less convincing. The model proposed by the GNU GPL allows for free distribution of software. The documents and opinions put forth by the *Free Software Foundation*, by *Stallman* himself and by many other free software activists and experts suggest that software developers have other effective means of monetizing their efforts such as offering custom adaptation services to individual clients, offering support and educational services or using the licensed works as a means of personal marketing which could lead to further business opportunities.

This marketing aspect of free software distribution has not been adequately assessed in literature and while further investigation of the matter is required before a sound scientific conclusion can be offered, it is safe to say that all of these activities require additional personal effort by the author to obtain rewards that may or may not be greater than the monetization based on the traditional economic exploitation of the statutory property rights contained in copyright.

From a legal perspective, the acceptance of the GPL has a profound negative impact on the position of the software developer. Those who choose the GPL license model to offer their work to the public, both users and future co-developers, should be well aware of the rights they relinquish by doing so.

The co-developers who approach a GPL licensed work with the intention to invest (their skill, time and money) in its improvement in order to use it in commercial sense need to be aware of the legal repercussions that follow from the use of GPL licensed work. While the GPL model is indeed immensely popular among software developers, there is a reasonable amount of court cases in comparative judicial practice that indicates a broad lack of understanding considering the ramifications of its use.

5 GPL Structure and analysis

The GPL v3 Agreement text consists of a preamble and eighteen articles numbered, somewhat eccentrically, from 0 to 17. The full text of the current version of the GNU GPL License Agreement is available at the GNU Foundation web page address and it is this text that is being considered here. The first two articles define the terms *source code*, *standard interface*, *system libraries*, *program.* The first Article also goes on to define terms *convey* and *propagate* where these terms basically encompass the activities of reproduction and distribution of work, as well as making it available to the public. In this regard, the term *convey* is defined as *„...any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying*“.

What this means is that to convey software is to distribute the software via data communication networks and protocols that would allow users to keep a copy of the software on their local information system or any other storage system of their own choosing. Accessing software over the communication network without this ability (i.e. using a remote desktop or similar feature to access a program that is run on another information system) does not constitute *conveying.* Both *propagate* and *convey* represent unorthodox legal constructs whose main purpose is to ensure the upkeep of the four cental freedoms GPL demands.

Article 2 of the Agreement defines the basic user rights concerning the licensed software. According to the Article 2 of the GPL, the users are allowed to compile, run, propagate and convey the licensed software.

Article 3. of the Agreement explicitely forbids the use of DRM technology on the licensed software, echoing the notions and positions from the preamble of the Agreement, and the philosophical standpoint of the GNU Foundation: „*Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users*."

The following Articles 4 through 7 regulate the obligation of the author as well as the user to, when distributing and copying the protected work, always include the source code, the full text of the GPL Agreement and notice whether the program has been modified.

Articles 8. through 10. consider the termination of the agreement as well as regulate the distribution of the protected works over peer-to-peer networks stipulating that such distribution need not require the explicit consent of the user holding that the user has already given consent to the terms of the licensing agreement by initiating data transfer.

6 GNU GPL and (software) patents

While the main function of the GPL Agreement is to regulate the use of protected works through innovative redefinition of rights from the user's perspective, the Article 11 of the GPL considers the effects of the GPL on patent rights. This is quite unusual.

Software patents, that is, patents concerning software technologies as well as particular programs themselves are far from a globally accepted intellectual property standard. They are specific to American and Asian legal systems, most notably the legal systems of United States, Japan, South Korea and other technologically advanced societies in Asia. The question of allowing patent protection to computer programs is one of the most controversial issues in the current field of intellectual property. The controversy is both theoretical (dichotomy between certain common law and European continental legal traditions) and practical (the fundamental political and economic reasons) [9,616]. Without taking into account the economic ramifications of granting software patents for computer programs, there can be no real insight into this matter. Since this issue is not the topic of this paper, suffice it to say that strong economic interests of big patent holders certainly exert an influence over the position of software patents in Europe [9,620].

With that in mind, the Article 11 of the GNU GPL binds the authors of the protected work to allow a free license of the relevant software patents to the users of the GNU licensed software (Article 3, paragraph 3 of the GPL).

Furthermore, according to the provisions of the Article 12 of the GNU GPL Agreement, the user of the licensed work is obliged to allow the subsequent users the same terms and conditions of using the licensed work. Should a user accept other conditions that may limit the aforementioned freedoms, those conditions that contradict the conditions of the GNU GPL License may not excuse the user from the conditions of the GPL.

Articles 15 and 16 of the GPL contain the clauses regarding the liability of the author concerning warranty (Article 16) and liability of the author for damages arising out of the use of the program, or the inability to use the program (Article 17). This waiver of liability covers the loss of data, data corruption, interoperability failure and so on.

The users of GPL licensed software therefore enjoy no warranty regarding the performance of the licensed computer programs and by themselves bear the risk of using these programs and suffering potential damages arising out of their use. The license Agreement limits the liability of the authors or rightsholders and all other potential actors (most notably subsequent users who became co-authors by contributing bits and pieces of code to the source code originally available to the public).

7 Conclusion

Free software and the defining license agreements like the GNU GPL have often been thought of as contrary to the well established system of intellectual property. In relation to the so called *proprietary software*, the software licensed by the rightsholders just like any other traditional cathegory of copyrighted-protected works, *free software* is often referred to as unreliable, *anti-copyright* oriented and of questionable legal safety.

The purpose of this article was to show that the provisions of the GPL, regardless of the manner in which they are presented, especially with the focus on user freedoms and the unorthodox nomenclature regarding the distribution, reproduction and publishing rights, are still valid copyright contracts [13].

These contracts continue to operate inside the scope of positive intellectual property legislation framework, both national and international, and have produced in theory well documented positive effects on the software industry in general.

The American and the European case law shows the courts on the both sides of the Atlantic have found the GNU GPL provisions and management of rights on the protected works to be in line with the positive legislative framework. Tens of thousands of programmers and software developers have used these contracts to develop and improve software applications used by the millions of users. *Free software* is not a threat to proprietary software. With it's lack of warranty and the scope of the limitation of liability of its numerous co-authors, it represents a different product aimed at different markets and different users. However, it certainly does represent competition considering its functionality and reliability and in this regard it affects the development of the proprietary software.

References:

[1] A. Guadamuz González: "*GNU General Public License v3: A Legal Analysis*", (2006) 3:2 SCRIPTed 154a;

[2] Ćuk, Ivan i Vukmir, Mladen: "*Novi modeli iskorištavanja autorskog djela*", Zbornik Hrvatskog društva za autorsko pravo, 7(2006), p.95-116;

[3] Dragičević, Dražen: „*Kompjutorski kriminalitet i informacijski sustavi*", Informator, Zagreb 2006.;

[4] Drahos, Peter: „*A philosophy of intellectual property*", Ashgate Darmouth 1996;

[5] Fitzgerald, Brian and Basset, Graham: „*Legal Issues Relating to Free and Open Source Software",*Queensland University of Technology School of Law, Essays in Technology Policy and Law, Volume 1, 2004;

[6] Gliha, Igor: "*Korištenje autorskih djela*", Zbornik Pravnog fakulteta Sveučilišta u Rijeci. Vol. 21, broj 1 (2000). p.111-126.

[7] Gliha, Igor: „*Raspolaganje autorskim pravom (i srodnim pravima)*", Zbornik hrvatskog društva za autorsko pravo, vol. 5 (2004), p. 97-98

[8] Gliha, Igor: "*Prava na autorskim djelima nastalim u radnom odnosu i po narudžbi*", Collected Papers of Zagreb Law Faculty, Zbornik Pravnog fakulteta vol.56, special issue, 2006.

[9] Hilty, Geiger: „*Patenting Software? A Judicial and Socio-Economic Analysis*" IIC 2005, 615;

[10] Höppner, Julian P: "*The GPL prevails: An analysis of the first-ever Court decision on the validity and effectivity of the GPL*", (2004) 1:4 SCRIPTed 628;

[11] Jaeger, Metzger: „*Die neue Version 3 der GNU General Public License*", IIC 2008, 130;

[12] Lerner, Pathak, Tirole: "*The Dynamics of Open-Source Contributors*", p.116, Roots of Innovation, vol.96 no.2, Massachussetts Institute of Technology;

[13] Moglen, Eben: „*Free Software Matters: Enforcing the GPL I and II*", http://emoglen.law.columbia.edu/publications/lu-12.html;

[14] Rosen, Lawrence: "*Open Source Licensing: Software Freedom and Intellectual Property Law*", Prentice Hall, 2005.