Free software, a new economic model?

Amadeu Albós Raya

PID_00145045

UOC
Universitat Oberta
de Catalunya
www.uoc.edu

Index

Introduction

This module will examine the paradigm of free software from the point of view of an economic model. In other words, we will study the fit and viability of free software as a model of economic operation that can be sustained in the long term.

In the study of free software as an economic model, we are limited by the relative youth of businesses based on free software. However, considering that the economic rules of the market are generally the same, we will base our study on the differentiation introduced by the free software business with respect to traditional markets. This view will give us a realistic initial approach to the qualities of free software as an economic model.

First of all, we will explore the foundations of the paradigm of free software and, hence, its operation and possibilities. More specifically, we will describe the conceptual features of the underlying operational philosophy of the model, such as its social production.

We will then explore the consequences of the model based on free software from different points of view, taking into account its differences with traditional models of software production and business. The projection of these concepts should give us a better understanding of how the free software model could fit into the market in the near future.

Lastly, we will study how the free software model relates to the validity and viability of companies based on it, explaining the importance of combining strategy with opportunity.

Objectives

After completing this module, students should have achieved the following aims:

1. To be familiar with the economics of the model based on free software.

2. To understand the fundamentals and implications of the free software model vis-à-vis the traditional model.

3. To understand the differentiation introduced by the free software model and evaluate its suitability for the creation of value for the market.

4. To explore the validity and viability of the free software model and exploitable business models.

1. Basis of the model

We are familiar with many of the technological features of free software that are similar to those of proprietary software to varying degrees. In other words, the fundamental differences – if any – between free and proprietary software are not based on the internal and external aspects of the product.

Broadly speaking, the technology applied to a product (for example, the design, architecture or specific implementation) does not in itself create a substantial differentiation between free and proprietary models, at least from the strict point of view of the finished product.

The differences between free software and other paradigms of software production (especially proprietary) mainly concern the specific features of the model of development, the user community and the differentiation of the product's value added.

These differences do not lie in technological aspects unique to the development of the application or software, but on the characteristics and implications underlying their production. In other words, free software sums up a particular orientation to create value in products and services that differ from the traditional point of view.

As explained in previous modules, the business models that exploit these distinctive features in a traditional market have been perfected in recent years. In all events, the chief value lies not in the software itself but in the capital acquired when it is adopted.

This capital formalises the foundations of free software. In other words, free software is based on the social production and network culture that not only allow but also promote its possibilities and effects.

The following sections will succinctly develop these two concepts. We will first of all examine the main features of social production before moving on to characterise network culture and its impact on the economics on which free software is based.

1.1. Social production

Advances in global communications and the democratisation of technology in recent decades could have influenced what we now consider free software in different ways.

That is, the ease of access to information and willingness to cooperate are not unique features of free software; they form a basis for the development of valid and viable alternatives in many fields.

While there are now many initiatives associated to varying degrees with social production, in this model, business organisations discover a way to encourage creation and attract value for their business models.

An example of social production

Wikipedia (http://www.wikipedia.org/).

In *The Wealth of Networks*, Yochai Benkler explores this issue in detail. Below, we will discuss some of the most relevant aspects characterising social production.

Recommended reading

L. Morgan; P. Finnegan(2008). *Deciding on open innovation: an exploration of how firms create and capture value with open source software*(Vol. 287, pp. 229-246). IFIP.

Economics of information

Information is a public good with economic implications at different levels as a result of the use of information technologies.

Innovation, as the creation of new information, may be adversely affected by situations with restriction or control, and facilitated by openness and collaboration on the production of information, knowledge and culture.

Recommended website

Y. Benkler(2006). *The Wealth of Networks: How social production transforms markets and freedom.* (http://www.benkler.org/Benkler_Wealth_Of_Networks.pdf).

Hence, production or innovation in peer-to-peer networks or generates a spiral of opportunities characterised by motivation and efficiency with technological support.

Peer-to-peer networks

In this case, the term refers to the operation of the community, rather than the architectural or technological basis of communication.

Development and distribution of information

The development and distribution of information can follow a variety of patterns, depending on how freedom is distributed between producers and consumers. In general, the more freedom given to producers, the less obtained by consumers.

Distribution channels for information influence how the latter is shared. The direction of the transfer and its aims also influence how information is shared.

In all events, licensing and patents can restrict the flow of information, while the quantitative growth of the network need not fragment or restrict it.

Implications of social production

Benkler maintains that the way we see the operational structure of the world around us is changing, especially in terms of how we all collaborate and interact with the integration of ideas and knowledge to create new knowledge.

1.2. Networked economy and culture

The implications of social production have become apparent in many fields in recent times, particularly in free software. The interaction of knowledge and the refinement of ideas is now a good way to encourage and further develop a concept.

> This view of production as a collaboration to qualitatively achieve a given aim contrasts with the more traditional view of the market of ideas and knowledge, where the importance lies more with the final adoption of the product than with consensus, fit or quality.

David Bollier's *When Push Comes to Pull: The New Economy and Culture of Networking Technology* explores how the evolution of information technology has allowed a new point of view to emerge that contrasts with the centralisation and hierarchy of the traditional model.

Recommended website

D. Bollier(2006). *When Push Comes to Pull: The New Economy and Culture of Networking Technology.* (http://www.aspeninstitute.org/atf/cf/%7bDEB6F227-659B-4EC8-8F84-8DF23CA704F5%7d/2005InfoTechText.pdf).

The following sections will now briefly examine the main economic and cultural features of networked culture considered by Bollier.

The push and pull models

The push model is based on mass production, anticipating consumer demand and dynamically managing time and the location of production resources.

The pull model is based on the openness and flexibility of the production platforms used as resources. This model does not anticipate consumer demand, but rather customises products according to demand using fast and dynamic processes.

Value creation networks

In pull models, the sharing of information and best practices substantially improves the corpus of knowledge of all members of the network.

This network promotes and integrates open business models based on the creation of value and product customisation or differentiation.

Hence, pull model platforms formalise, improve and increase the flexibility of innovation and evolution through the community, without incurring the costs of a similar implementation in a push model.

Target market

Push models are successful in areas where consumers are not very clear on what they want and prefer to make their selection based on predefined typologies.

By contrast, in pull models, consumers want to form part of the production and selection process, in the sense that they may not know exactly what they want, but they are sure that they want to participate and form part of the process.

Production

Push models tend to seek alternative forms of production that may be more economically competitive (for example, lower production costs), while pull models tend mainly to seek the best ways to add value to the production network.

This special orientation of pull models favours the scalability of the production network and the union of the best participants for production specialisation.

Cooperation

Pull models favour the creation of relationships based on trust, the sharing of knowledge and cooperation among members of the network, to everybody's benefit.

This ethos is often transformed into a system of collective government for the sustainable and fair management of shared resources.

In this sense, companies based on pull models should provide guarantees for the recognition of network members, since the model is based on trust and the creation of value.

Education

With push models, the activity of students is focused on the construction of static knowledge as prior training for a subsequent hierarchical society.

Pull models promote alternative forms of education in that information technologies allow students to enter a dynamic flow of activity with access to many independent resources for creating their own corpus of knowledge (and sharing it).

2. Characteristics of the free software model

The foundations of free software formalise a structure in which cooperation and the sharing of knowledge among members allow for the innovation, production and evolution of global knowledge.

The creation of value is undoubtedly an important goal for all members of the community (be they users, developers, etc.) and for the model itself. Hence, the decentralisation, freedom and independence that are the mark of the community offer guarantees for the consolidation and cohesion of production and social capital.

> The free software model is based on differentiation in relation to the values that govern the traditional market, both from the point of view of software development and of appreciation of the value created.

While it is true that, from a traditional point of view, some of the features of the free software model are also applicable to other paradigms of development and value creation, the free software model introduces new features to the perception and appreciation of the values associated with the traditional market.

In this section of the module, we will determine the features of the free software model by comparing them with those of a traditional model, with the aim of assessing the real differentiation introduced by the model in daily practice.

First of all, we will discuss the model from the point of view of software development, before moving on to analyse the implications of differentiation as a paradigm based on social production.

2.1. Software development

The methodology of free software development is possibly one of the factors popularly considered as a differentiation compared to other software development paradigms, such as the proprietary model. But is this really the case?

From the point of view of software production, there are points in free software development that clearly overlap with other development models, such as proprietary, since the production methodologies have a certain independence from specific implementations.

However, the fact that software production may be more or less consistent with other models or that some of the requirements for code freedom are more or less necessary in practice, this does not mean that there cannot be significant differences in other aspects leading us to evaluate the whole as innovative.

Fuggetta's article *Software libre y de código abierto: ¿un nuevo modelo para el desarrollo de software?* explores these and other aspects of the differences between the development model of free software and the development model of proprietary software. The following sections will briefly outline some of its findings.

Recommended website

A. Fuggetta(2004). *Software libre y de código abierto: ¿un nuevo modelo para el desarrollo de software?* (http://alarcos.inf-cr.uclm.es/doc/ig1/doc/temas/4/IG1-t4slibreabierto.pdf)

Context

The success of free software can be attributed to a range of technological and economic aspects affecting its innovation and production.

Its decentralisation, cooperation and freedom of use and exploitation have made free software the standard-bearer of a new philosophy for addressing and solving a variety of problems.

According to Fuggetta, many beliefs on free software can also be applied to proprietary software, so it is a good idea to explore the topic thoroughly.

The development process

From a technological standpoint, the development of free software is not a new paradigm, since most projects have a limited number of collaborators. Moreover, incremental and evolutionary development methodologies are not unique to free software.

Nonetheless, free software has managed to motivate both developers and users to get involved in the project, sharing and associating the development and evolution of the software with the needs of the community.

Defence of customer rights

Problems related to customer protection arise mainly in reference to software packages, since the customer already owns the code in custom developments.

For software packages, it may be enough to be able to access the source code without subsequently modifying or redistributing it. The company's user support should also abide by rules that facilitate the handing over of the code in the event that the company cannot maintain it.

Dissemination of knowledge

The spread of knowledge through access to the source code is insufficient, since the subjects on software engineering reveal that documents describing the software architecture are also needed.

Moreover, in the event that this knowledge could be disseminated, it would only be necessary to publish its source code (without the right to copy and redistribute the software).

Cost

The fact that the software is released under a free license does not mean it cannot be commercialised or that its development does not have an associated cost (although we do not know the extent of this).

In addition, just because we cannot quantify or centralise its cost, this does not mean that it is not distributed among the collaborators, even indirectly by companies with little or nothing in common with the world of software.

Effectiveness of the business model

The main business models that actually exploit free software engage in the development and distribution of pure open source packages or free and proprietary software platforms. Other forms of business can be set up to a greater or lesser extent with both free and proprietary software.

Moreover, there is currently no evidence to suggest that a company based solely on services will be profitable over time.

On profitability

In *The Business of Software*, Michael Cusumano argues that software companies will increasingly depend on the combination of revenue from licences and services.

The software industry

Europe does not have an industrial strategy to unify the actions of the various companies involved. Hence, offering support to free software is not a strategy in comparison to the creation of innovative products.

2.2. The cooperative paradigm

While some of the features of the free software model are not innovative from a classical perspective, as we saw above, those that motivate a change in market perspective are.

To analyse in detail the differentiation introduced by the free software model compared to other traditional models, we will need to assess the aspects of production and the creation of value and knowledge underpinning the model.

In his article *Open Source Paradigm Shift*, Tim O'Reilly identifies these and other features of free software that are differentiating and which create a competitive advantage that can be exploited for profit. The following paragraphs will briefly outline some of his findings.

Recommended website

T. O'Reilly(2004). *Open Source Paradigm Shift.*

(http://www.oreillynet.com/pub/a/oreilly/tim/articles/paradigmshift_0504.html).

Change

Free software has deeply transformed the structure of the benchmark market, often with implications extending beyond those imagined by its creators.

These changes are based on product quality, lower production costs and the use of standards, in addition to differentiation in marketing, distribution and logistics.

Software as a commodity

In a context of permanent standardised communications such as the current one, all communication applications are interchangeable (a web browser, for example). In other words, the use of standards means that software can be considered a commodity.

Hence, when the revenue-generating potential of an application is diminished because of the commoditisation process, a new market will emerge for proprietary products, especially if they exploit the global communications network.

Moreover, free software remains a viable model for companies providing services, although we cannot expect similar profit margins to those of the modern software giants.

Network collaboration

The culture of software sharing has grown since its origins at the same pace as the Internet, whose participatory architecture is present in virtually all of its functionalities.

Free software is the natural language of the networked community, resulting in a style of collaboration and participation unique to its members. This collaboration is critical to the success and differentiation of leading Internet applications, since it has highlighted the importance of treating users as co-developers of the software.

Customisation and software as a service

Nowadays, we are used to considering applications as artefacts rather than static processes. Programs require engineering for their creation but the dynamic languages that allow for the cohesion of components (such as data management scripts) offer the perspective of a dynamic and evolving process of the application.

Many of the services offered on-line (such as search engines) require constant revisions and updates in order to perform their functions properly. This generates a new business paradigm for computers and information technology in general, and for the exploitation of software as a service in particular.

The Internet operating system

We can consider the Internet as a single virtual computer that builds an operating system from the connection of several small pieces and allows anybody to participate in the creation of value.

The values of the free software user community are important to the paradigm as they promote the spirit of seeking out and sharing knowledge.

The commoditising of technology is part of the process that allows the industry to move forward to create more value for everybody. For industry, it is essential to strike a balance that will create more value than that obtained with individual participants.

3. Validity and feasibility of the free software model

In previous sections, we looked at the foundations of the free software model and the features that distinguish it from the more traditional models.

To evaluate the long-term sustainability of the free software model, we need much more data than we currently possess, i.e. a much wider time slot for a more precise comparison with traditional models.

Time will tell whether free software is a new economic model and what features and conditions will allow it to be so.

We will now offer some conclusions. Although, at the time of writing, business based on free software is still relatively new, we have highlighted the differences allowing the adoption of a new business perspective based primarily on promoting the cooperative production of knowledge.

Applications based on free software

The social production of a specific application or solution encourages the creation of value above and beyond its cost of production, affording it a competitive advantage over other market alternatives.

Free software-based applications, together with open standards, can offset some of the economic effects that strengthen products based on the traditional model. Thus, besides inducing a substantial differentiation with traditional applications, they allow for strategies and policies of coopetition between companies in a win-win paradigm.

The market

Social production has plagued the Internet with alternative initiatives to traditional models. Over time, social capital has become a significant value for innovation and development in open environments. We now have profitable business models that pay for the production of knowledge.

The business of knowledge

Innocentive (http://www.innocentive.com/) is just one website that rewards ideas that solve specific problems. On it, there are users who pose questions (seekers) and others that solve them (solvers) in exchange for a financial reward.

This and other examples have led to the creation of a new market logic, referred to in some contexts as Wikinomics and crowdsourcing. This logic is based on the pull model we saw above, i.e. the attraction of ideas and effort in contrast to the traditional push model.

In time, we will discover whether this market perspective allows the patterns of technology adoption typical of the traditional market to evolve towards a new situation.

The business

The new market perspective can offer new business opportunities associated with the exploitation of ideas, concepts and knowledge for profit without owning the latter. In other words, the value of an application based on free software does not lie in the solution itself but in the capital acquired and generated with it.

Nonetheless, the validity and viability of free software as a model also depends on the particular design features of the company that exploits it. That is, it is essential to design the company around a solid and lasting business opportunity.

Risks

Undoubtedly, the main risks for the model based on free software are obtaining a critical mass of users to ensure the project's viability and laying the foundations for a business model that will prove stable over time. We must also take into account the relationship between the initial investment and the expected benefits.

Business viability study

Comprehensively analysing, designing and formalising the company will increase the guarantees of success of our free software-based business. To maximise these guarantees, the company's viability must be studied prior to its launch and formalised in a business plan.

See also

In the third module of this subject, we took an initial approach to the main features affecting the business viability of the traditional software business, namely aspects of sales and marketing, along with the products and services covered by the business.

Companies based on free software must complement the above aspects with the features of business models based on free software seen in the fourth module, creating a combination to formalise a sound basis on which to set up a sustainable business .

The free software company

As with any business model, a company based on free software will also require detailed planning and design prior to start-up. In the previous sections, we emphasised the importance of carefully analysing the business fundamentals of the company as a condition for evaluating its validity and feasibility.

Both the basic features of free software and the implications that we have described throughout this module can exert different influences depending on the typology of the business opportunity and the context we seek to exploit.

Thus, the strategy of a company based on free software can and should characterise its actions in the differentiation of its business and the economic effects of its environment, as well as in social capital and production, in addition to coopetition.

Summary

Throughout this module, we have explored the features of free software as an economic model, even considering the constraints of data limitations given the fact that business models based on free software are relatively new.

Firstly, the basics of social capital and the collective production of knowledge and ideas are not unique to free software. There are currently several initiatives demonstrating how cooperation and collaboration can be feasible in the innovation and production of knowledge.

These basics reveal the importance of the network of collaborators and their involvement and motivation in the global and individual progress of the members of the community. They also offer a viable alternative to traditional production models.

The implications of the philosophy of social production can also be explored from different points of view. While certain features of free software do not reveal major differences with other models, there are some features that can lead to important distinctions.

In the free software business, it is essential to strengthen and exploit the distinguishing fundamentals of free software in order to provide valid and viable alternatives to traditional models. These actions must inevitably be complemented by the detailed study and planning of the business opportunity to ensure the viability and future of the free software company.

Bibliography

Benkler, Y.(2006). *The wealth of networks: How social production reforms markets and freedom*. New Haven: Yale University Press. <http://www.benkler.org/Benkler_Wealth_Of_Networks.pdf; WIKI:http://cyber.law.harvard.edu/wealth_of_networks/Main_Page> [Consulted in March 2009]

Bollier, D.(2006). *When Push Comes to Pull: The New Economy and Culture of Networking Technology*. <http://www.aspeninstitute.org/atf/cf/%7bDEB6F227-659B-4EC8-8F84-8DF23CA704F5%7d/2005InfoTechText.pdf>

Fogel, K.(2004). *The Promise of the Post-Copyright World*<http://www.questioncopyright.org/promise> [Consulted in March 2009]

Fuggetta, A.(Sept – Oct 2004): *Open Source and Free Software: A New Model for the Software Development Process?* (No. 171). Novática – Upgrade: Monografía del proceso de software, English <http://www.upgrade-cepis.org/issues/2004/5/up5-5Fuggetta.pdf) | Spanish: (http://alarcos.inf-cr.uclm.es/doc/ig1/doc/temas/4/IG1-t4slibreabierto.pdf> [Consulted in February 2009]

Goldhaber, M.(June 2006). *The Value of Openness in an Attention Economy*(Vol. 11, no. 6). <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1334/1254> [Consulted in March 2009]

Moglen, E.(1999). *Anarchism Triumphant and the Death of Copyright*. <http://www.uic.edu/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/684/594> [Consulted in March 2009]

Morgan, L.; Finnegan, P. (2008). *Deciding on open innovation: an exploration of how firms create and capture value with open source software*. In: G. León; A. Bernardos; J. Casar; K. Kautz; J. DeGross (ed.). International Federation for Information Processing. Open IT-Based Innovation: Moving Towards Cooperative IT Transfer and Knowledge Diffusion (Vol. 287, pp. 229-246). Boston: Springer.

O'Reilly, T.(2004). *Open Source Paradigm Shift*.<http://tim.oreilly.com/articles/paradigmshift_0504.html> [Consulted in February 2009]

Appendix

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

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.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means 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.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>
    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.