

El software libre, ¿un nuevo modelo económico?

Amadeu Albós Raya

PID_00145045



Universitat Oberta
de Catalunya

www.uoc.edu

Índice

Introducción	5
Objetivos	6
1. Las bases del modelo	7
1.1. La producción social	8
1.2. Economía y cultura en red	9
2. Las características del modelo del software libre	12
2.1. El desarrollo de software	12
2.2. El paradigma cooperativo	15
3. La validez y la viabilidad del modelo del software libre	18
Resumen	21
Bibliografía	23
GNU GENERAL PUBLIC LICENSE.....	24

Introducción

En el presente módulo examinaremos el paradigma del software libre desde el punto de vista de modelo económico. Es decir, estudiaremos el ajuste y la viabilidad del software libre como modelo de funcionamiento económico sostenible a largo plazo.

En el estudio del software libre como modelo económico estaremos limitados por la relativa juventud del negocio basado en software libre. Aun así, considerando que las reglas económicas del mercado por lo general no han sido modificadas, basaremos el estudio en la diferenciación que supone el negocio del software libre con respecto a los mercados tradicionales. Este punto de vista nos permitirá obtener una primera aproximación realista sobre las cualidades del software libre como modelo económico.

En un primer momento revisaremos las bases que sustentan el paradigma del software libre y, por ende, su funcionamiento y posibilidades. Es decir, veremos aquellas características conceptuales ligadas a la filosofía subyacente y operativa del modelo, como por ejemplo la producción social.

A continuación analizaremos las consecuencias del modelo basado en software libre desde diferentes puntos de vista, teniendo en cuenta las diferencias que supone con respecto a los modelos tradicionales de producción de software y modelos de negocio. La proyección de estos conceptos nos debe ayudar a comprender mejor cuál puede ser el encaje del modelo del software libre en el mercado en un futuro próximo.

Por último estudiamos cómo se relaciona el modelo del software libre con la validez y la viabilidad de las empresas que se basan en él, denotando la importancia de conjugar tanto estrategia como oportunidad.

Objetivos

Los objetivos que se deben alcanzar al finalizar este módulo son los siguientes:

- 1.** Familiarizarse con los aspectos económicos del modelo ligado al software libre.
- 2.** Entender los fundamentos y las implicaciones del modelo de software libre con respecto al modelo tradicional.
- 3.** Comprender la diferenciación que supone el modelo de software libre y evaluar su adecuación a la creación de valor para el mercado.
- 4.** Profundizar en la validez y viabilidad del modelo de software libre y en los modelos de negocio explotables.

1. Las bases del modelo

Del software libre conocemos muchas de las características tecnológicas que, en mayor o menor medida, pueden resultar similares a las de cualquier software propietario. Es decir, las diferencias fundamentales (de haberlas) entre el software libre y el propietario no están basadas en los aspectos internos o externos del producto.

A grandes rasgos, la tecnología aplicada a un producto (por ejemplo, el diseño, la arquitectura o la implementación particular), no justifican por sí solas una diferenciación substancial entre modelos libres y propietarios, al menos desde el punto de vista estricto del producto acabado.

Las principales diferencias entre el software libre y otros paradigmas de producción de software (especialmente el propietario) se centran en las particularidades del modelo de desarrollo, la comunidad de usuarios y en la diferenciación del valor añadido del producto.

Estas diferencias no se basan en aspectos tecnológicos propios de la aplicación o del software para desarrollarlo, sino en las características e implicaciones subyacentes a su producción. Es decir, condensa una orientación particular para crear valor en productos y servicios que difieren del punto de vista tradicional.

Como se explica en los módulos precedentes, a lo largo de los últimos años se han perfeccionado modelos de negocio que explotan estas características diferenciales en un mercado tradicional. En cualquier caso, el valor principal no recae en el software en sí, sino en el capital que se adquiere cuando se adopta.

Este capital formaliza los fundamentos del software libre. Es decir, el software libre se fundamenta en la producción social y la cultura en red que no sólo lo hacen posible sino que también potencian sus capacidades y efectos.

En los siguientes apartados desarrollaremos ambos conceptos de forma breve. En un primer momento examinaremos las principales características de la producción social y después caracterizaremos la cultura en red y sus efectos en la economía que sustenta el software libre.

1.1. La producción social

Posiblemente, tanto los avances en las comunicaciones globales como la democratización de la tecnología a lo largo de las últimas décadas han influido de diferentes formas en lo que hoy en día consideramos como software libre.

Es decir, la facilidad de acceso a la información y la voluntad de cooperación no son características únicas del software libre, sino que estructuran una base para el desarrollo de alternativas válidas y viables en multitud de campos.

Si bien en la actualidad existen múltiples iniciativas ligadas en mayor o menor medida a la producción social, las organizaciones empresariales encuentran en este modelo una manera de incentivar la creación y captura de valor para sus modelos de negocio.

Yochai Benkler, en su libro *The Wealth of Networks*, estudia en profundidad esta cuestión. A continuación revisaremos algunos de los aspectos más relevantes que caracterizan la producción social.

La economía de la información

La información es un bien público que tiene implicaciones económicas en diferentes niveles gracias a las tecnologías de la información.

La innovación, como creación de nueva información, puede verse negativamente afectada por situaciones de restricción o control, mientras que puede ser facilitada por la apertura y colaboración en la producción de la información, el conocimiento y la cultura.

En este sentido, la producción o innovación en redes colaborativas o de igual a igual genera una espiral de oportunidades caracterizada por la motivación y la eficiencia con el soporte de la tecnología.

El desarrollo y distribución de la información

El desarrollo y distribución de la información puede seguir diversidad de patrones específicos en función de la distribución de libertad entre productores y consumidores. En general, cuanta más libertad se otorga al productor, menos libertad obtiene el consumidor.

Los canales de distribución de la información influyen en la forma de compartirla. El sentido direccional de la transmisión y los objetivos de la misma influyen también en cómo se comparte la información.

Una muestra de producción social

Por ejemplo, la Wikipedia (<http://www.wikipedia.org/>).

Lectura recomendada

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

Web recomendada

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

Redes colaborativas

En inglés, *Peer-to-Peer*. En este caso, el término hace referencia al funcionamiento de la comunidad y no al soporte arquitectural o tecnológico que sustenta la comunicación.

En cualquier caso, las licencias y las patentes pueden restringir el flujo de información, mientras que el crecimiento cuantitativo de la red no tiene por qué fragmentarla o limitarla.

Las implicaciones de la producción social

Benkler afirma que nuestra forma de percibir la estructura de funcionamiento del mundo que nos rodea está en plena transformación, especialmente en cuanto a la forma de colaborar e interactuar entre todos en la integración de ideas y conocimientos para crear nuevo conocimiento.

1.2. Economía y cultura en red

Las implicaciones de la producción social se han hecho patentes en multitud de campos en los últimos tiempos, especialmente en el software libre. La interacción de conocimientos y el refinamiento de ideas es, en la actualidad, una buena forma de motivar y profundizar en el desarrollo de un concepto.

Esta visión de la producción como colaboración para alcanzar cualitativamente un objetivo concreto contrasta con la visión más tradicional del mercado de ideas y conocimientos, donde la importancia recae más en la adopción final del producto que en el consenso, la adecuación o la calidad.

En la publicación de David Bollier *When Push Comes to Pull: The New Economy and Culture of Networking Technology* se examina con detalle cómo la evolución de la tecnología de la información ha permitido crear un nuevo punto de vista que contrasta con la centralización y jerarquía del modelo tradicional.

En los siguientes apartados examinaremos brevemente las principales características económicas y culturales de la cultura en red que considera Bollier.

El modelo *push* y el modelo *pull*

El modelo *push* se basa en la producción en masa, anticipando la demanda de los consumidores y gestionando de forma dinámica el tiempo y la ubicación de los recursos de producción.

El modelo *pull* se basa en la apertura y la flexibilidad de las plataformas de producción que se utilizan como recursos. Este modelo no anticipa la demanda de los consumidores, sino que personaliza los productos en función de la demanda mediante procesos rápidos y dinámicos.

Redes de creación de valor

Web recomendada

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>).

En los modelos *pull*, el hecho de compartir tanto la información como las buenas prácticas mejora sustancialmente el corpus de conocimientos de todos los miembros de la red.

Esta red fomenta y cohesiona modelos de negocio abiertos basados en la creación de valor y personalización o diferenciación de los productos.

En este sentido, las plataformas del modelo *pull* formalizan, mejoran y flexibilizan la innovación y evolución a través de la comunidad, sin tener que asumir los costes que supondría su implementación similar en un modelo *push*.

El mercado objetivo

Los modelos *push* tienen éxito en las áreas donde los consumidores no tienen muy claro lo que quieren y prefieren realizar la selección sobre tipologías predefinidas.

En cambio, en los modelos *pull*, los consumidores quieren formar parte del proceso de elaboración y selección, en el sentido de que quizás no saben qué es lo que quieren, pero están seguros de querer participar y formar parte del proceso.

La producción

Los modelos *push* tienden a buscar alternativas de producción que puedan resultar más competitivas económicamente (por ejemplo, menor coste de producción), mientras que los modelos *pull* tienden a buscar sobre todo las mejores formas de aportar valor a la red de producción.

Esta particular orientación de los modelos *pull* favorece la escalabilidad de la red de producción, así como el reencuentro de los mejores participantes para la especialización de la producción.

La cooperación

En los modelos *pull* se favorece la creación de relaciones basadas en la confianza, el hecho de compartir conocimientos y la cooperación entre los miembros que forman parte de la red para el beneficio de todos.

En muchas ocasiones, esta particular filosofía se transforma en un régimen de gobierno colectivo para gestionar de forma equitativa y sostenible los recursos compartidos.

En este sentido, las empresas basadas en los modelos *pull* deben ofrecer garantías para el reconocimiento de los miembros de la red, puesto que el modelo se basa en la confianza y en la creación de valor.

La educación

Los modelos *push* permiten centrar la actividad de los estudiantes en construir conocimiento estático, a modo de entrenamiento para la posterior sociedad jerarquizada.

Los modelos *pull* promueven formas de educación alternativas, en el sentido de que las tecnologías de la información hacen posible que los estudiantes entren en un flujo de actividad dinámica y puedan acceder a multitud de recursos independientes para crear su propio corpus de conocimiento (y de compartirlo a su vez).

2. Las características del modelo del software libre

Los fundamentos en los que se basa el software libre formalizan una estructura donde la cooperación y el hecho de compartir conocimientos entre sus miembros permite innovar, producir y hacer evolucionar el conocimiento global.

Sin lugar a dudas, la creación de valor es un objetivo importante para todos los miembros de la comunidad (ya sean usuarios, desarrolladores, etc.) y para el modelo en sí. En este sentido, la descentralización, la libertad y la independencia que rigen la comunidad ofrecen garantías para consolidar y cohesionar tanto la producción como el capital social.

El modelo del software libre se fundamenta en la diferenciación de los valores que rigen el mercado tradicional, tanto desde el punto de vista de desarrollo de software como de la apreciación del valor creado.

Si bien es cierto que desde un punto de vista clásico algunas de las características del modelo del software libre son aplicables también a otros paradigmas de desarrollo y de creación de valor, el modelo del software libre introduce novedades en la percepción y apreciación de los valores ligados al mercado tradicional.

En este apartado del módulo examinaremos cuáles son las características del modelo del software libre en comparación con las de un modelo tradicional, con el objetivo de valorar la diferenciación real que propone el modelo en la práctica diaria.

En un primer momento examinaremos el modelo desde el punto de vista del desarrollo del software, para luego analizar las implicaciones de la diferenciación como paradigma basado en la producción social.

2.1. El desarrollo de software

La metodología de desarrollo de software libre es posiblemente uno de los factores que popularmente se consideran diferenciadores con respecto a otros paradigmas de desarrollo de software, por ejemplo el modelo propietario. ¿Es realmente así?

Desde el punto de vista de producción de software, el desarrollo de software libre tiene importantes puntos de coincidencia con otros modelos de desarrollo, por ejemplo el propietario, dado que las metodologías de producción tienen una cierta independencia con respecto a las implementaciones particulares.

Pero que la producción de software pueda resultar más o menos coincidente con respecto a otros modelos o que algunos de los requisitos de libertad sobre el código sean más o menos necesarios en la práctica, no implica que no pueda existir una diferencia significativa en otros aspectos que permita valorar el conjunto como novedoso.

En este sentido, el artículo de Fuggetta titulado *Software libre y de código abierto: ¿un nuevo modelo para el desarrollo de software?* analiza en profundidad este y otros aspectos de las diferencias entre el modelo de desarrollo del software libre y el modelo de desarrollo del software propietario. En los siguientes apartados se revisan de forma breve algunas de sus conclusiones.

El contexto

El éxito del software libre se puede atribuir a una variedad de aspectos tecnológicos y económicos ligados a la innovación y a la producción del mismo.

Las características de descentralización, cooperación y libertad de uso y explotación convierten al software libre en el caballo de batalla de una nueva filosofía para abordar y solucionar problemáticas de diversa índole.

Según Fuggetta, muchas de las creencias sobre el software libre pueden ser aplicadas también al software propietario, por lo que es conveniente realizar un análisis exhaustivo de la temática.

El proceso de desarrollo

Desde el punto de vista tecnológico, el desarrollo del software libre no supone un nuevo paradigma, puesto que la mayoría de proyectos se sustentan en un número limitado de colaboradores, mientras que las metodologías de desarrollo incrementales y evolutivas no son exclusivas del software libre.

En cambio, el software libre ha conseguido motivar tanto a desarrolladores como a usuarios a involucrarse en el proyecto, compartiendo y conectando el desarrollo y la evolución del software con las necesidades de la comunidad.

La protección de los derechos de los clientes

Web recomendada

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>)

Los problemas relacionados con la protección de los clientes aparecen principalmente cuando se trata de un paquete de software, puesto que en los desarrollos a medida el cliente ya es propietario del código.

En el caso de paquetes de software podría resultar suficiente tener acceso al código fuente sin poder modificarlo y redistribuirlo a su vez. Por otro lado, el soporte al usuario por parte de la empresa debería regirse por normas que faciliten la entrega del código en el caso de que la empresa no pueda asumir su mantenimiento.

La difusión del conocimiento

La propagación del conocimiento mediante el acceso al código fuente es insuficiente, puesto que las materias relacionadas con la ingeniería del software demuestran que es necesario disponer de documentos que describan la arquitectura del software.

Por otra parte, en el caso de que se pudiera propagar este conocimiento, sólo sería necesaria la publicación de su código fuente (sin derecho a copiar y redistribuir el software).

El coste

El hecho de que el software esté publicado bajo una licencia libre no significa que no pueda ser comercializado o que su desarrollo no tenga un coste asociado (aunque no sepamos qué magnitud tiene).

Por otra parte, que no se pueda cuantificar o centralizar su coste no significa que este no sea asumido de forma distribuida por los colaboradores, incluso de forma indirecta por empresas que poco o nada pueden relacionarse con el mundo del software.

La efectividad del modelo de negocio

Las principales modelos de negocio que explotan realmente el software libre son los que se dedican al desarrollo y distribución de paquetes de código abierto puros o a plataformas de software libre y propietarias. Otras formas de negocio pueden aplicarse en mayor o menor medida tanto al software libre como al propietario.

Por otra parte, por el momento nada indica que una empresa basada únicamente en servicios pueda ser rentable en el tiempo.

La industria del software

Europa no dispone de una estrategia industrial que permita cohesionar las actuaciones de las diferentes empresas implicadas. En este sentido, apoyar el software libre no es una estrategia en comparación con la creación de productos innovadores.

2.2. El paradigma cooperativo

Si bien algunas de las características del modelo del software libre no son innovadoras desde una perspectiva clásica, tal como hemos examinado anteriormente, sí lo son aquellas que motivan precisamente un cambio en la perspectiva del mercado.

Para valorar en profundidad la diferenciación que supone el modelo de software libre con respecto a otros modelos tradicionales hay que valorar los aspectos de producción y creación de valor y de conocimiento en los que se basa el modelo.

En el artículo *Open Source Paradigm Shift*, Tim O'Reilly desgrana estas y otras características del software libre como diferenciadoras y creadoras de una ventaja competitiva explotable de forma lucrativa. En los próximos apartados realizaremos un breve repaso a algunas de sus conclusiones.

El cambio

El software libre supone un cambio profundo en la estructura del mercado de referencia que, en muchas ocasiones, tiene implicaciones que van más allá de las consideradas por sus creadores.

Los cambios se sustentan en la calidad del producto, la disminución del coste de producción y la explotación de estándares, así como en la diferenciación en las áreas de marketing, distribución y logística.

El software como *commodity*

Considerando un contexto de comunicaciones permanentes estandarizadas como el actual, todas las aplicaciones de comunicación son intercambiables entre sí (por ejemplo, un navegador web). Es decir, la explotación de estándares provoca que el software pueda ser considerado como *commodity*.

Sobre la rentabilidad

En el libro *The Business of Software*, Michael Cusumano argumenta que las compañías de software dependerán cada vez más de la combinación de ingresos entre licencias y servicios.

Web recomendada

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

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

En este sentido, cuando el potencial de generar beneficios de una aplicación se diluye a causa del proceso de *comoditization*, aparece un nuevo mercado para productos propietarios, especialmente si éstos explotan la red global de comunicaciones.

Por otra parte, el software libre sigue siendo un modelo viable para empresas prestadoras de servicios, aunque no se puede esperar un margen de beneficios similar al de las grandes compañías de software actuales.

Colaboración en red

La cultura del intercambio de software ha crecido desde sus inicios al mismo ritmo que Internet, cuya arquitectura participativa se implementa en prácticamente todas sus funcionalidades.

El software libre constituye el lenguaje natural de la comunidad en red, dando lugar a un estilo de colaboración y participación particular entre sus miembros. Esta colaboración es fundamental para el éxito y la diferenciación de las aplicaciones líderes en Internet, dado que ha puesto de relieve la importancia de tratar a los usuarios como co-desarrolladores del software.

Personalización y *software-as-service*

En la actualidad, estamos acostumbrados a considerar una aplicación como un artefacto estático y no como un proceso. Los programas requieren ingeniería para ser creados, pero los lenguajes dinámicos que permiten la cohesión de los componentes (por ejemplo, los *scripts* de gestión de datos), ofrecen una perspectiva de proceso dinámico y evolutivo de la aplicación.

Buena parte de los servicios que se ofrecen por Internet (por ejemplo, un buscador) requieren revisiones y actualizaciones constantes para realizar su función correctamente. Esta situación genera un nuevo paradigma de negocio relacionado con el mundo de los ordenadores y de la tecnología de la información, en general, especialmente la explotación del software como servicio.

El sistema operativo Internet

Podemos tratar Internet como un único ordenador virtual que construye un sistema operativo a partir de la unión de diferentes piezas pequeñas y permite la participación de cualquiera en la creación de valor.

Los valores de la comunidad de usuarios del software libre son importantes para el paradigma, puesto que promueven un espíritu de buscar y compartir el conocimiento.

El proceso de *commoditization* de la tecnología es parte del proceso que permite que la industria avance para crear más valor para todos. Para la industria es esencial encontrar el equilibrio que permita crear más valor que el que se captura con participantes individuales.

3. La validez y la viabilidad del modelo del software libre

En los anteriores apartados hemos examinado tanto los fundamentos sobre los que se sostiene el modelo del software libre como las características que lo diferencian de los modelos más tradicionales.

Para valorar la sostenibilidad del modelo del software libre a largo plazo necesitamos muchos más datos que los que se poseen en la actualidad, es decir un corte temporal mucho más amplio que permita una equiparación mucho más precisa con los modelos tradicionales.

El paso del tiempo determinará si el software libre constituye un nuevo modelo económico y cuáles serán las características y condiciones que lo permitirán.

Veamos a continuación algunas conclusiones. A pesar de que en el momento de desarrollar los presentes materiales el negocio basado en software libre es relativamente reciente, se han puesto de relieve las diferencias que permiten una nueva perspectiva del negocio basada principalmente en la potenciación de la producción cooperativa del conocimiento.

La aplicación basada en software libre

La producción social de una aplicación o solución concreta favorece la creación de valor por encima de su coste de producción, lo que confiere una ventaja competitiva respecto a otras alternativas del mercado.

Las aplicaciones basadas en software libre, conjuntamente con los estándares abiertos, pueden contrarrestar algunos de los efectos económicos que fortalecen los productos basados en el modelo tradicional. En este sentido, además de inducir una diferenciación sustancial respecto las aplicaciones tradicionales, permiten estrategias y políticas de coopectencia entre empresas bajo un paradigma ganador-ganador.

El mercado

La producción social ha plagado Internet de iniciativas alternativas a los modelos tradicionales. El capital social ha devenido, con el paso del tiempo, un importante valor para la innovación y desarrollo en entornos abiertos. En la actualidad existen modelos de negocio lucrativos que remuneran la producción de conocimiento.

El negocio del conocimiento

Innocentive (<http://www.innocentive.com/>), entre otros, es un portal web que se dedica a recompensar las ideas que solucionan problemas concretos. Es decir, hay usuarios que plantean problemas (*seekers*) y otros que los solucionan (*solvers*) a cambio de una compensación económica.

Este y otros ejemplos han dado pie a la creación de una nueva lógica de mercado, llamada en algunos contextos *wikinomía* y *crowdsourcing*. Esta lógica se basa en el modelo *pull* que hemos visto anteriormente, es decir, la atracción de ideas y esfuerzos frente al tradicional modelo *push*.

Con el paso del tiempo veremos si esta perspectiva de mercado permite la evolución de los patrones de adopción tecnológica propios del mercado tradicional hacia una nueva situación.

El negocio

La nueva perspectiva de mercado puede ofrecer nuevas oportunidades de negocio ligadas a la explotación lucrativa de ideas, conceptos y conocimientos sin tener que ser propietarios de los mismos. Es decir, el valor de la aplicación basada en software libre no está en la solución en sí, sino en el capital que se adquiere y en el que se puede generar con él.

Aun así, la validez y la viabilidad del software libre como modelo también se sustentan en las particularidades de concepción de la empresa que lo explota. Es decir, resulta fundamental concebir la empresa sobre una oportunidad de negocio sólida y duradera.

Los riesgos

Sin duda, los principales riesgos para el modelo basado en software libre son obtener una masa crítica de usuarios que permita la viabilidad del proyecto y fundamentar un modelo de negocio estable a lo largo del tiempo. También hay que tener en cuenta la relación entre inversión inicial y beneficios esperados.

El estudio de viabilidad empresarial

Analizar, diseñar y formalizar exhaustivamente la empresa nos permite aumentar las garantías de éxito del negocio basado en software libre. Para maximizar estas garantías, la viabilidad de la empresa debe estudiarse previamente a su puesta en marcha y formalizarse en un plan de empresa.

En la empresa basada en software libre debemos complementar los aspectos anteriores con las características de los modelos de negocio basados en software libre vistos en el cuarto módulo, de forma que la combinación permita formalizar una base sólida sobre la que fundamentar un negocio sostenible.

La empresa de software libre

Como en cualquier otro modelo empresarial, la empresa basada en el software libre también requiere un diseño y planificación detallados antes de su puesta en marcha. En los anteriores apartados hemos puesto de relieve la importancia de analizar detenidamente los fundamentos comerciales de la empresa como condición para evaluar su validez y viabilidad.

Tanto los fundamentos del software libre como las implicaciones que hemos ido detallando a lo largo del presente módulo pueden ejercer fuerzas diferentes en función de la tipología de la oportunidad de negocio y el contexto donde se pretende explotar.

De este modo, la estrategia de la empresa basada en software libre puede y debe caracterizar sus actuaciones en la diferenciación del negocio y en los efectos económicos de su entorno, así como en el capital y producción social, además de la cooperación.

Ved también

En el tercer módulo de esta asignatura ya se ha realizado una primera aproximación a las principales características ligadas a la viabilidad empresarial del negocio clásico del software, por ejemplo los aspectos de comercialización y de marketing, así como a los productos y servicios objeto del negocio empresarial.

Resumen

A lo largo de este módulo hemos examinado las características del software libre como modelo económico, aun considerando las limitaciones propias de la escasez de datos debido a que los modelos de negocio basados en software libre son relativamente recientes.

Por una parte, los fundamentos del capital social y la producción colectiva de ideas y conocimientos no son exclusivos del software libre. En la actualidad existen diferentes iniciativas que demuestran cuán factibles pueden resultar la cooperación y colaboración en la innovación y producción de conocimientos.

Estos fundamentos denotan la importancia de la red de colaboradores, su implicación y motivación en el avance global e individual de los miembros de la comunidad, y suponen una alternativa viable a los modelos tradicionales de producción.

Por otra parte, las implicaciones de la filosofía de producción social pueden ser examinadas desde diferentes puntos de vista. Si bien es cierto que ciertas características del software libre no suponen una diferencia significativa con respecto a otros modelos, existen otras características que sí promueven una diferenciación importante.

Considerando el negocio del software libre, resulta primordial fortalecer y explotar los fundamentos diferenciadores del software libre para ofrecer alternativas válidas y viables a los modelos tradicionales. Estas actuaciones deben complementarse necesariamente con el estudio y planificación detallados de la oportunidad de negocio para garantizar la viabilidad presente y futura de la empresa del software libre.

Bibliografía

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> [Fecha de consulta: marzo del 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>> [Fecha de consulta: marzo del 2009]

Fuggetta, A. (sept.-oct., 2004): *Open Source and Free Software: A New Model for the Software Development Process?* (núm. 171). Novática – Upgrade: Monografía del proceso de software, Inglés <<http://www.upgrade-cepis.org/issues/2004/5/up5-5Fuggetta.pdf>> | Español: (<http://alarcos.inf-cr.uclm.es/doc/ig1/doc/temas/4/IG1-t4slibreabierto.pdf>) [Fecha de consulta: febrero del 2009]

Goldhaber, M. (junio, 2006). *The Value of Openness in an Attention Economy* (vol. 11, núm. 6). <<http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1334/1254>> [Fecha de consulta: marzo del 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>> [Fecha de consulta: marzo del 2009]

Morgan, L.; Finnegan, P. (2008). *Deciding on open innovation: an exploration of how firms create and capture value with open source software*. En: 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, pág. 229-246). Boston: Springer.

O'Reilly, T. (2004). *Open Source Paradigm Shift*. <http://tim.oreilly.com/articles/paradigmshift_0504.html> [Fecha de consulta: febrero del 2009]

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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 per-

forming 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>>.

