

Il software libero: un nuovo modello economico?

Amadeu Albós Raya

PID_00145045



Universitat Oberta
de Catalunya

www.uoc.edu

Indice

Introduzione.....	5
Obiettivi.....	6
1. Le basi del modello.....	7
1.1. La produzione sociale	8
1.2. Economia e cultura in rete	9
2. Le caratteristiche del modello del software libero.....	12
2.1. Lo sviluppo del software	12
2.2. Il paradigma cooperativo	15
3. L'efficacia e la validità del modello di software libero.....	18
Riepilogo.....	21
Bibliografia.....	23
Appendice.....	24

Introduzione

In questo modulo esamineremo il concetto di software libero considerandolo come modello economico. Studieremo, per meglio dire, la definizione e la sostenibilità del software libero in qualità di modello di funzionamento economico sostenibile nel lungo periodo.

Nello studio del software libero come modello economico saremo limitati dalla relativa novità del business basato sul software libero. Nonostante ciò, tenendo in considerazione che in generale le regole economiche del mercato non sono state modificate, ci baseremo, nello studio, sulla differenza che distingue il business del software libero dai mercati tradizionali. Questo punto di osservazione ci permetterà di ottenere una prima approssimazione realistica circa le potenzialità del software libero come modello economico.

Inizialmente passeremo in rassegna le basi sostegno del concetto di software libero, il suo funzionamento e le sue possibilità. Ci soffermeremo, cioè, su quelle caratteristiche concettuali legate alla filosofia operativa e sottostante il modello, così come, ad esempio, la produzione sociale.

In seguito analizzeremo le conseguenze del modello basato sul software libero da diverse angolazioni, tenendo conto delle differenze che presenta rispetto ai modelli tradizionali di produzione di software e a quelli di business. La presentazione di questi concetti ci sarà di aiuto per comprendere meglio come potrà inserirsi il modello del software libero nel mercato in un futuro prossimo.

Termineremo poi con uno studio sulla relazione tra il modello del software libero e solidità e forza delle imprese che vi fanno affidamento, dando rilievo all'importanza di coniugare strategia ed opportunità.

Obiettivi

Al termine di questo modulo lo studente dovrà essere in grado di:

1. Avere familiarità con gli aspetti economici del modello legato al software libero.
2. Conoscere i concetti cardine e le implicazioni del modello di software libero rispetto al modello tradizionale.
3. Comprendere la peculiare diversità del modello di software libero e saper valutare la sua capacità di creare valore per il mercato.
4. Saper approfondire la validità e la sostenibilità del modello di software libero ed i modelli di business utilizzabili.

1. Le basi del modello

Del software libero conosciamo molte delle caratteristiche tecnologiche che, in minor od in maggior misura, possono risultare simili a quelle di un qualsiasi software di proprietà. Ovverossia, qualora vi siano, le differenze fondamentali tra il software libero ed il software di proprietà non si riscontrano negli aspetti interni od esterni del prodotto.

A grandi linee, la tecnologia applicata ad un prodotto (ad esempio il disegno, l'architettura o la realizzazione particolare) non giustifica di per sé una differenza sostanziale tra modelli liberi e modelli di proprietà, al meno dal ristretto punto di vista del prodotto finito.

Le principali differenze tra il software libero ed altri modelli di produzione di software (su tutti, quello di proprietà) si rilevano nelle particolarità del progetto di sviluppo, nella comunità degli utenti e nella differenza di valore aggiunto del prodotto.

Queste differenze non si basano su aspetti tecnologici propri dell'applicazione o del software, ma nelle caratteristiche e nelle implicazioni sottostanti la sua produzione. Concentra, cioè, un orientamento particolare alla creazione di valore in prodotti e servizi che differiscono dal punto di vista tradizionale.

Come spiegato nei moduli precedenti, nel corso degli ultimi anni si sono perfezionati modelli di business che sfruttano queste caratteristiche differenziali in un mercato tradizionale. Ad ogni modo, il valore maggiore non ricade nel software in sé, bensì nel capitale che si acquisisce quando se ne adotta l'utilizzo.

Questo capitale caratterizza stabilmente i fondamenti del software libero. Questo significa che il software libero trova il suo fondamento nella produzione sociale e nella cultura in rete, che non solo lo rendono possibile, ma ne potenziano anche le capacità e gli effetti.

Nelle prossime sezioni svilupperemo brevemente entrambi i concetti. Prima esamineremo le principali caratteristiche della produzione sociale e poi ci occuperemo della cultura in rete e della sua influenza sull'economia che sostiene il software libero.

1.1. La produzione sociale

Probabilmente, tanto i progressi nella comunicazione globale quanto la diffusione della tecnologia degli ultimi decenni hanno influito, seppure in diversi modi, su quello che oggi chiamiamo software libero.

Basti osservare la facilità d'accesso all'informazione e la volontà di cooperazione, che non sono caratteristiche del software libero solamente, ma bensì creano la base per lo sviluppo di alternative valide e praticabili in una moltitudine di campi.

Anche se al giorno d'oggi esistono molteplici iniziative più o meno connesse alla produzione sociale, le organizzazioni imprenditoriali ritrovano in questo modello uno strumento per incentivare la creazione e la cattura di valore per i loro modelli di business.

Yochai Benkler, nel suo libro *The Wealth of Networks*, studia approfonditamente questa questione. Di seguito ci addentreremo in alcuni degli aspetti più rilevanti che caratterizzano la produzione sociale.

L'economia dell'informazione

L'informazione è un bene pubblico che ha implicazioni economiche a vari livelli grazie alle tecnologie dell'informazione.

L'innovazione, intesa come creazione di nuova informazione, può essere difficile in situazioni di limitazione o di controllo, mentre è spesso facilitata dall'apertura e dalla collaborazione nella produzione di informazione, conoscenza e cultura.

In questo senso, la produzione o innovazione verso reti di collaborazione o da pari a pari genera una spirale di opportunità caratterizzate da motivazione ed efficienza, con il supporto della tecnologia.

Lo sviluppo e la diffusione dell'informazione

Lo sviluppo e la diffusione dell'informazione possono seguire cammini diversi a seconda della distribuzione di libertà esistente tra produttori e consumatori. In generale, quanta più libertà viene affidata al produttore, tanta meno ne ottiene il consumatore.

I canali di diffusione dell'informazione influiscono sulla maniera di diffonderla. Anche il verso direzionale della trasmissione e gli obiettivi della stessa influiscono su come viene diffusa l'informazione.

Un esempio di produzione sociale

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

Lettura consigliata

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

Web consigliato

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

Reti di collaborazione

In inglese, *Peer-to-Peer*. In questo caso, la parola fa riferimento al funzionamento della comunità e non all'assistenza tecnologica che sostiene la comunicazione.

In ogni caso, le licenze ed i brevetti possono restringere il flusso dell'informazione, mentre non si vede perché la crescita della rete dovrebbe renderla frammentaria o limitarla.

Le implicazioni della produzione sociale

Benkler afferma che il nostro modo di percepire la struttura di funzionamento del mondo che ci sta attorno è in piena trasformazione, soprattutto quanto al modo di collaborare ed interagire tra tutti nel processo di condivisione di idee e di conoscenze per creare nuova conoscenza.

1.2. Economia e cultura in rete

Le implicazioni della produzione sociale ultimamente sono diventate più chiare in una moltitudine di campi, soprattutto in quello del software libero. La condivisione e l'interazione di conoscenze e il perfezionamento di idee sono, al giorno d'oggi, una buona maniera per approfondire lo sviluppo di un concetto.

Questa visione della produzione come collaborazione per il raggiungimento di un obiettivo concreto contrasta con la visione più tradizionale del mercato circa idee e conoscenze. In esso l'importanza risiede più nella realizzazione finale del prodotto che nel consenso, nel dialogo o nella qualità.

Nella pubblicazione di David Bollier *When Push Comes to Pull: The New Economy and Culture of Networking Technology* viene esaminato in dettaglio come l'evoluzione della tecnologia dell'informazione ha permesso di creare un nuovo punto di vista in aperto contrasto con la tendenza all'accentramento e la gerarchia del modello tradizionale.

Nei successivi paragrafi esamineremo brevemente le principali caratteristiche economiche e culturali della cultura in rete che considera Bollier.

Il modello *push* e il modello *pull*

Il modello *push* si basa sulla produzione di massa. Si anticipa la domanda dei consumatori e si gestiscono in maniera dinamica i tempi e la logistica delle risorse della produzione.

Il modello *pull* si basa sull'apertura e sulla flessibilità dei programmi di produzione che vengono utilizzati come risorse. Non anticipa la domanda dei consumatori, ma personalizza i prodotti in funzione della domanda mediante processi rapidi e dinamici.

Web consigliato

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>).

Reti di creazione di valore

Nei modelli *pull*, il fatto di condividere tanto la informazione quanto le buone procedure migliora in maniera sostanziale il corpus delle conoscenze di tutti i membri della rete.

Questa rete stimola e rende coesi modelli di business aperti, strutturati sulla creazione di valore e sulla personalizzazione o differenziazione dei prodotti.

In questo contesto, i programmi del modello *pull* plasmano, migliorano e danno flessibilità all'innovazione e all'evoluzione attraverso la comunità, senza dover sostenere i costi che, a condizioni simili, sarebbero inevitabili in un modello di tipo *push*.

Il mercato obiettivo

I modelli *push* hanno successo nei settori in cui i consumatori non hanno molto chiaro quello che cercano e preferiscono affidarsi a tipologie predefinite.

Per contro, nei modelli *pull*, i consumatori vogliono avere voce nel processo di elaborazione e selezione, nel senso che magari non sanno esattamente cosa vogliono, però sono certi di voler partecipare nel processo.

La produzione

I modelli *push* tendono a cercare alternative di produzione che possano risultare economicamente più competitive (ad esempio, minimizzando i costi di produzione), mentre i modelli *pull* tendono piuttosto alla ricerca delle soluzioni più atte ad apportare valore alla rete di produzione.

Questo specifico orientamento dei modelli *pull* favorisce la possibilità di risalire nella scala della rete di produzione, così come anche il confronto tra i migliori partecipanti, per ottenere la migliore specializzazione della produzione.

La cooperazione

Nei modelli *pull* si favorisce la creazione di una rete di relazioni basate sulla fiducia, sulla condivisione delle conoscenze e sulla collaborazione tra i membri della rete per il beneficio di tutti.

In svariate occasioni, questa particolare filosofia sfocia in un regime di governo collettivo, allo scopo di gestire con giustizia ed allo stesso tempo efficacia le risorse comuni.

Per questo, le imprese che operano attraverso modelli *pull* devono garantire il riconoscimento di tutti i membri della rete, dato che il modello si fonda su fiducia e creazione di valore.

L'educazione

I modelli *push* permettono agli studenti di concentrare i loro sforzi sulla costruzione di conoscenze statiche, che servano ad abituarsi alla successiva società gerarchizzata.

I modelli *pull* promuovono forme di educazione alternative, in quanto le tecnologie dell'informazione rendono possibile che gli studenti entrino in un vortice di attività dinamiche e possano avere accesso ad un massiccio insieme di risorse indipendenti tale da potersi creare il proprio corpus di conoscenze (e di metterlo a loro volta a disposizione della condivisione).

2. Le caratteristiche del modello del software libero

I fondamenti sui quali poggia il software libero danno forma ad una struttura nella quale la collaborazione e la condivisione di conoscenza tra i suoi membri permettono di innovare, modificare e far progredire la conoscenza globale.

Senza alcun dubbio, la creazione di valore è un obiettivo importante per tutti i membri della comunità (che siano utenti, sviluppatori, ecc.) e per il modello in se stesso. Per questo, la decentralizzazione, la libertà e l'indipendenza che reggono la comunità offrono solide garanzie per consolidare e compattare tanto la produzione quanto il capitale sociale.

Il modello del software libero trova il suo fondamento nella differenza con i valori che reggono il mercato tradizionale, dal punto di vista dello sviluppo del software così come da quello della valutazione del valore creato.

Se pure è vero che da un punto di vista classico alcune caratteristiche del modello del software libero sono applicabili anche ad altri schemi di sviluppo e di creazione di valore, il modello del software libero introduce delle novità sostanziali nella percezione e nella valutazione dei valori connessi al mercato tradizionale.

In questa parte del modulo esamineremo le caratteristiche del modello del software libero mettendole a confronto con quelle di un modello tradizionale, allo scopo di dare risalto alla piena novità che propone il modello nel suo svolgimento quotidiano.

Nel primo paragrafo esamineremo il modello nell'ottica dello sviluppo del software, per poi passare, nel secondo, all'analisi delle implicazioni della sua novità, intesa come schema basato sulla produzione sociale.

2.1. Lo sviluppo del software

La metodologia utilizzata nel lavoro di sviluppo del software libero è probabilmente uno dei fattori universalmente ritenuti caratterizzanti la differenza con gli altri schemi di sviluppo di software, ad esempio con il modello di proprietà. Ma è davvero così?

Dal punto di vista di produzione di software, lo sviluppo di software libero ha importanti punti di contatto con altri modelli di sviluppo, ad esempio con il modello di proprietà, dato che i metodi di produzione hanno una certa indipendenza relativamente alle singole realizzazioni.

Il fatto però che la produzione di software possa risultare più o meno coincidente con altri modelli, o che alcuni dei requisiti di libertà rispetto al codice siano più o meno necessari nella pratica, non implica che non possa esistere una differenza significativa sotto altri aspetti, la quale permetta di attribuire all'insieme un carattere di novità.

A proposito di questo contesto, l'articolo di Fuggetta intitolato *Software libre y de código abierto: ¿un nuevo modelo para el desarrollo de software?* analizza in dettaglio questo ed altri aspetti che differenziano il modello di sviluppo del software libero dal modello di sviluppo del software di proprietà. Nei seguenti paragrafi si espongono brevemente alcune delle sue conclusioni.

Il contesto

Il successo del software libero può essere attribuito ad una complessa varietà di aspetti tecnologici ed economici connessi all'innovazione ed alla produzione del modello stesso.

Le caratteristiche di decentralizzazione, collaborazione e libertà di uso e di sfruttamento trasformano il software libero nel cavallo di battaglia di una nuova filosofia che si avvicina a problematiche di diversa natura e tenta di risolverle.

Secondo Fuggetta, molte delle convinzioni relative al software libero possono essere attribuite anche al software di proprietà, ed è pertanto conveniente realizzare uno studio esaustivo della tematica.

Il processo di sviluppo

Sotto il profilo tecnologico, lo sviluppo del software libero non implica un nuovo schema, dato che la maggioranza dei progetti sono maneggiati da un numero limitato di collaboratori, mentre le metodologie di sviluppo incrementale e di sviluppo evolutivo non sono esclusive del software libero.

Per contro, il software libero è riuscito a motivare tanto gli sviluppatori quanto gli utenti ad inserirsi nel progetto, riuscendo a far coincidere lo sviluppo e l'evoluzione del software con le necessità della comunità.

La protezione dei diritti dei clienti

Web consigliato

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>)

I problemi relativi alla protezione dei clienti compaiono principalmente quando si ha a che fare con un pacchetto di software, dato che negli sviluppi fatti su misura il cliente è già proprietario del codice.

Nel caso di pacchetti di software potrebbe essere sufficiente avere accesso al codice fonte senza poterlo modificare o passarlo ad altri. D'altra parte, il supporto all'utente da parte dell'impresa dovrebbe reggersi su norme che facilitino la consegna del codice nel caso in cui l'impresa non si possa fare carico del suo mantenimento.

La diffusione della conoscenza

La diffusione della conoscenza mediante l'accesso al codice fonte è insufficiente, se è vero, come è vero, che le materie relazionate all'ingegneria del software dimostrano che è necessario disporre di documenti che descrivano l'architettura del software.

D'altro canto, se fosse possibile diffondere questa conoscenza, sarebbe sufficiente la pubblicazione del suo codice fonte (tolto il diritto di copiare e ridistribuire il software).

Il costo

Il fatto che il software sia divulgato con una licenza libera non implica che non possa essere commercializzato o che il suo sviluppo non abbia dei costi associati (anche se non ne conosciamo l'entità).

D'altra parte, che non si possano quantificare i costi e che non si possano assegnare ad un unico centro di spesa, questo non significa che gli stessi non siano distribuiti tra i collaboratori, tenendo in considerazione anche l'ipotesi di distribuzione per via indiretta ad imprese che poco o niente hanno a che fare con il mondo del software.

La validità del modello di business

I principali modelli di business che sfruttano realmente il software libero sono quelli che si dedicano allo sviluppo ed alla distribuzione di pacchetti puri con codice aperto, così come quelli che sfruttano le piattaforme di software libero e le piattaforme di proprietà. Altri approcci al business si combinano più o meno bene tanto al software libero come a quello di proprietà.

D'altra parte, fino ad oggi non abbiamo nulla che indichi che un'impresa basata unicamente sui servizi possa essere redditizia in maniera duratura nel tempo.

L'industria del software

L'Europa non dispone di una strategia industriale che permetta l'azione coordinata delle diverse imprese coinvolte. In questo senso, sostenere il software libero non è una strategia, rispetto alla creazione di prodotti innovativi.

2.2. Il paradigma cooperativo

Anche se alcune delle caratteristiche del modello di software libero non sono innovative secondo la prospettiva classica, come abbiamo visto, lo sono invece quelle che causano esattamente un cambiamento nella prospettiva del mercato.

Per assegnare il giusto valore all'originalità del modello di software libero rispetto agli altri modelli tradizionali, bisogna prendere in considerazione gli aspetti della produzione e della creazione di valore, oltre che quelli delle conoscenze sulle quali si fonda il modello.

Nell'articolo *Open Source Paradigm Shift*, Tim O'Reilly analizza queste ed altre caratteristiche del software libero, considerandole come innovatrici e capaci di creare un vantaggio competitivo che può essere sfruttato a scopo di lucro. Qui di seguito passeremo in rassegna alcune delle sue conclusioni.

Il cambiamento

Il software libero suppone un cambiamento profondo nella struttura del mercato di riferimento, che, in svariate occasioni, ha implicazioni e conseguenze che vanno oltre quelle immaginate dai suoi creatori.

I cambiamenti si manifestano nella qualità del prodotto, nell'abbassamento dei costi di produzione e nello sfruttamento dei modelli standard, così come nelle novità nelle aree marketing, distribuzione e logistica.

Il software come *commodity*

Se consideriamo un contesto in cui la comunicazione permanente è standard, come quello in cui viviamo, tutte le applicazioni che veicolano la comunicazione sono intercambiabili tra loro (ad esempio, un navigatore web). Questo per dire che lo sfruttamento di modelli standard fa sì che il software può essere considerato come una *commodity*.

Sulla redditività

Nel libro *The Business of Software*, Michael Cusumano afferma che le compagnie di software dipenderanno sempre più dalla combinazione delle entrate tra licenze e servizi.

Web consigliato

T. O'Reilly (2004). *Open Source Paradigm Shift*.
(http://www.oreillynet.com/pub/a/oreilly/tim/articles/paradigmshift_0504.html).

Per questo, quando la capacità di un'applicazione di generare profitti si esaurisce a causa del processo di *comoditization*, in quel momento compare un nuovo mercato per i prodotti di proprietà, soprattutto quando questi utilizzano la rete di comunicazione globale.

Nonostante questo, il software libero continua ad essere un modello affidabile per le imprese di servizi, anche se non ci si può aspettare un margine di guadagni pari a quello delle grandi compagnie di software.

Collaborazione in rete

La cultura dello scambio del software è cresciuta fin dai suoi inizi allo stesso ritmo di Internet, che sperimenta ogni giorno di più una maggiore partecipazione in praticamente tutte le sue funzioni.

Il software libero costituisce il linguaggio naturale della comunità in rete, dando luogo ad uno stile di collaborazione e di partecipazione da parte di ognuno dei suoi membri. Questa collaborazione è fondamentale per il successo e per gli aggiornamenti delle applicazioni leader di Internet, dato che ha dato risalto all'importanza del trattare gli utenti come co-sviluppatori del software.

Personalizzazione e *software-as-service*

Al giorno d'oggi siamo ormai abituati a considerare un'applicazione non come un processo, ma piuttosto come un dato immobile. I programmi hanno bisogno di ingegneria per essere creati, ma i linguaggi dinamici che permettono la coesione delle componenti (ad esempio, degli *scripts* di gestione dei dati) forniscono la prospettiva di un processo dinamico ed in evoluzione per l'applicazione.

Buona parte dei servizi che vengono offerti su Internet (ad esempio, un motore di ricerca) ha bisogno di revisioni ed aggiornamenti costanti per poter assolvere correttamente il suo compito. Questa situazione genera un nuovo modello di business legato al mondo dei computer e della tecnologia dell'informazione, in generale, ed in particolar modo all'utilizzo del software come servizio.

Il sistema operativo Internet

Possiamo trattare Internet come se fosse un grande ed unico computer virtuale che costituisce il sistema operativo partendo dall'unione di tanti piccoli pezzi e che consente la partecipazione di chiunque nella creazione di valore.

I valori della comunità degli utenti del software libero sono importanti per il modello, in quanto promotori di uno spirito di ricerca e di condivisione delle scoperte.

Il processo di *commodization* della tecnologia è parte del processo che permette all'industria di progredire nella direzione di creare un maggior valore per tutti. Per l'industria è essenziale trovare l'equilibrio che le permetta di creare un valore maggiore di quello che si otterrebbe se la partecipazione fosse più semplicemente individuale.

3. L'efficacia e la validità del modello di software libero

Nei sottocapitoli precedenti abbiamo esaminato tanto i fondamenti che stanno alla base del modello del software libero, quanto le caratteristiche che lo rendono diverso dai modelli più tradizionali.

Per assegnare un valore alla sostenibilità nel lungo periodo del modello del software libero ci servono molti più dati di quelli che sono ad oggi disponibili; ci manca, cioè, la possibilità di osservare uno spazio temporale molto più ampio, che ci dia maggiori strumenti per poter confrontare il modello con quelli più tradizionali in modo più preciso.

Il passare del tempo sarà il discrimine che determinerà se il software libero costituisce realmente un nuovo modello economico, e quali saranno le caratteristiche e le condizioni del modello che lo renderanno possibile.

Facciamo in merito delle considerazioni. A dispetto del fatto che il business basato sul software libero sia al momento della stesura di questo testo relativamente recente, abbiamo cercato di dare risalto alle differenze che permettono di avere una nuova prospettiva del business basata principalmente sul potenziamento della produzione in forma di cooperazione della conoscenza.

L'applicazione basata sul software libero

La produzione su base allargata di una applicazione o di una soluzione concreta favorisce la creazione di valore senza alcun impatto negativo per i costi di produzione, cosa che assegna un vantaggio competitivo rispetto ad altre alternative del mercato.

Le applicazioni basate sul software libero, assieme con l'apertura dei modelli standard, possono limitare alcuni degli effetti economici che rafforzano i prodotti basati sul modello tradizionale. In questo modo, oltre ad indurre una differenza sostanziale rispetto alle applicazioni tradizionali, rendono possibili strategie e politiche di concorrenza tra imprese secondo un profilo vincitore-vincitore.

Il mercato

La produzione sociale ha riempito Internet di iniziative alternative ai modelli tradizionali. Il capitale sociale è diventato, con il passare del tempo, un valore importante per l'innovazione e lo sviluppo in ambienti aperti. Oggi esistono modelli di business lucrativi che remunerano la produzione di conoscenza.

Il business della conoscenza

Innocentive (<http://www.innocentive.com/>), tra gli altri, è un portale web dedicato a ricompensare le idee che diano una soluzione a problemi concreti. Per meglio dire, vi partecipano utenti che propongono problemi (*seekers*) ed altri che li risolvono (*solvers*) in cambio di un riconoscimento economico

Questo ed altri esempi hanno dato avvio alla creazione di una nuova logica di mercato, chiamata in alcuni contesti *wikinomia* e *crowdsourcing*. Questa logica si basa sul modello *pull* che abbiamo precedentemente spiegato, ovvero, sulla attrazione di idee e di sforzi messa a confronto con il tradizionale modello *push*.

Col tempo scopriremo se questa prospettiva di mercato consentirà l'evoluzione dei protettori del mercato tradizionale verso una nuova situazione, nel mondo della tecnologia.

Il business

La nuova prospettiva di mercato può schiudere nuove opportunità di business legate all'utilizzo a scopo di lucro di idee, concetti e conoscenze, senza che si debba esserne i proprietari. Questo significa che il valore dell'applicazione basata sul software libero non risiede nella soluzione in se stessa, ma bensì nel capitale che si acquisisce ed in quello che si può generare attraverso l'utilizzo del software libero.

Nonostante questo, l'efficacia e l'affidabilità del software libero come modello traggono sostentamento anche dalla particolare concezione dell'impresa che lo utilizza. Sembra fondamentale concepire e vivere l'impresa come una opportunità di business solida e duratura.

I rischi

Senza dubbio, i principali rischi per il modello basato sul software libero sono: ottenere una massa di utenti che renda sostenibile il progetto, e gettare le fondamenta di un modello solido ed affidabile nel tempo. Bisogna poi tener conto del rapporto tra investimento iniziale e benefici attesi.

Lo studio di sostenibilità dell'impresa

L'analizzare, disegnare e dare forma all'impresa in maniera completa ed esaustiva ci permette di accrescere le garanzie di successo del business basato sul software libero. Per massimizzare queste garanzie, la sostenibilità dell'impresa va progettata e studiata prima della sua costituzione, e va precisata in un piano di azione dell'impresa.

Nell'impresa basata sul software libero dobbiamo completare gli aspetti precedenti con le caratteristiche dei modelli di business basati sul software libero che vedremo nel quarto modulo, di modo che la loro combinazione dia luogo alla possibilità di realizzare una base solida sulla quale edificare un business sostenibile.

L'impresa di software libero

Come in qualunque altro modello aziendale, anche l'impresa basata sul software libero richiede un disegno ed una progettazione dettagliata prima dell'avvio della sua attività. Nei paragrafi precedenti abbiamo sottolineato l'importanza dell'analizzare con cura i fondamenti commerciali dell'impresa come condizione per valutare la sua validità e la sua sostenibilità.

Tanto i fondamenti del software libero come le implicazioni che abbiamo visto in dettaglio in questo modulo possono esercitare forze differenti in funzione della tipologia dell'opportunità di business e del contesto nel quale si vuole lavorare.

Di conseguenza, la strategia dell'impresa basata sul software libero può e deve modificare le sue azioni considerando l'originalità del business, gli effetti economici sul suo ambiente, il capitale e la produzione sociale, e la concorrenza.

Guarda anche

Nel terzo modulo di questo corso si è già fatta una prima approssimazione delle principali caratteristiche legate alla sostenibilità imprenditoriale del business classico del software, ad esempio gli aspetti di commercializzazione e di marketing, così come i prodotti ed i servizi oggetto del business imprenditoriale.

Riepilogo

In questo modulo abbiamo esaminato le caratteristiche del software libero come modello economico, sempre tenendo in considerazione la scarsità di dati disponibili, dovuta alla relativa novità dei modelli di business nati sullo schema del software libero.

Da un lato, i fondamenti del capitale sociale e la produzione collettiva di idee e conoscenze non sono esclusivi del software libero. Ai giorni nostri esistono diverse iniziative che dimostrano quanto possano risultare fattibili la cooperazione e la collaborazione nell'innovazione e nella produzione di conoscenza.

Questi fondamenti fanno notare l'importanza della rete di collaboratori, la sua implicazione e la sua spinta verso un progresso globale ed individuale dei membri della comunità, e rappresentano un'alternativa credibile ai modelli tradizionali di produzione.

Per altro verso, le implicazioni della filosofia della produzione sociale possono essere posti sotto diverse lenti di ingrandimento. Se anche è vero che certe caratteristiche del software libero non rappresentano una differenza significativa rispetto ad altri modelli, esistono altre caratteristiche che sono fortemente innovative.

Se si considera il business del software libero, risulta di primaria importanza rinforzare e servirsi degli aspetti originali del software libero per offrire alternative valide ed affidabili rispetto ai modelli tradizionali. Queste azioni devono essere necessariamente accompagnate dallo studio e dalla progettazione dettagliata del business, perché ne venga garantita la sostenibilità presente e futura.

Bibliografia

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> [Consultazione: marzo 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>> [Consultazione: marzo 2009]

Fuggetta, A. (set.-ott., 2004): *Open Source and Free Software: A New Model for the Software Development Process?* (num. 171). Novática – Upgrade: Monografia del processo del software, inglese <<http://www.upgrade-cepis.org/issues/2004/5/up5-5Fuggetta.pdf>> | spagnolo: (<http://alarcos.inf-cr.uclm.es/doc/ig1/doc/temas/4/IG1-t4slibreabierto.pdf>)> [Consultazione: febbraio 2009]

Goldhaber, M. (giugno, 2006). *The Value of Openness in an Attention Economy* (vol. 11, num. 6). <<http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/1334/1254>> [Consultazione: marzo 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>> [Consultazione: marzo 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, pag. 229-246). Boston: Springer.

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

Appendice

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>>.

