

Seminari ALSI 2007

Licenze Software e Open Source

Marino Miculan
DiMI, Università di Udine



SOME RIGHTS RESERVED

Questa presentazione è distribuita sotto la licenza Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (<http://creativecommons.org/licenses/by-nc-sa/3.0/>)

Marino Miculan <miculan@dimi.uniud.it>

“Open source”?

- Il termine “open source” viene applicato a molte *metodologie* di sviluppo software
- Due aspetti comunemente intesi
 - Comunità distribuita di sviluppatori
 - Codice sorgente disponibile a tutti
- Come si conciliano questi aspetti con il concetto di “licenza” e “copyright”?
- Molte licenze “open source” (GPL, LGPL, BSD, Qt, ...). Quali sono le differenze?

Outline

- Modello di sviluppo Open Source
 - Definizione di Open Source
 - Caratteristiche comuni a tutte le licenze O.S.
 - Caratteristiche variabili
- Esempi di Licenze O.S.
- Conclusioni

Il modello di sviluppo a “codice aperto” (Open Source)

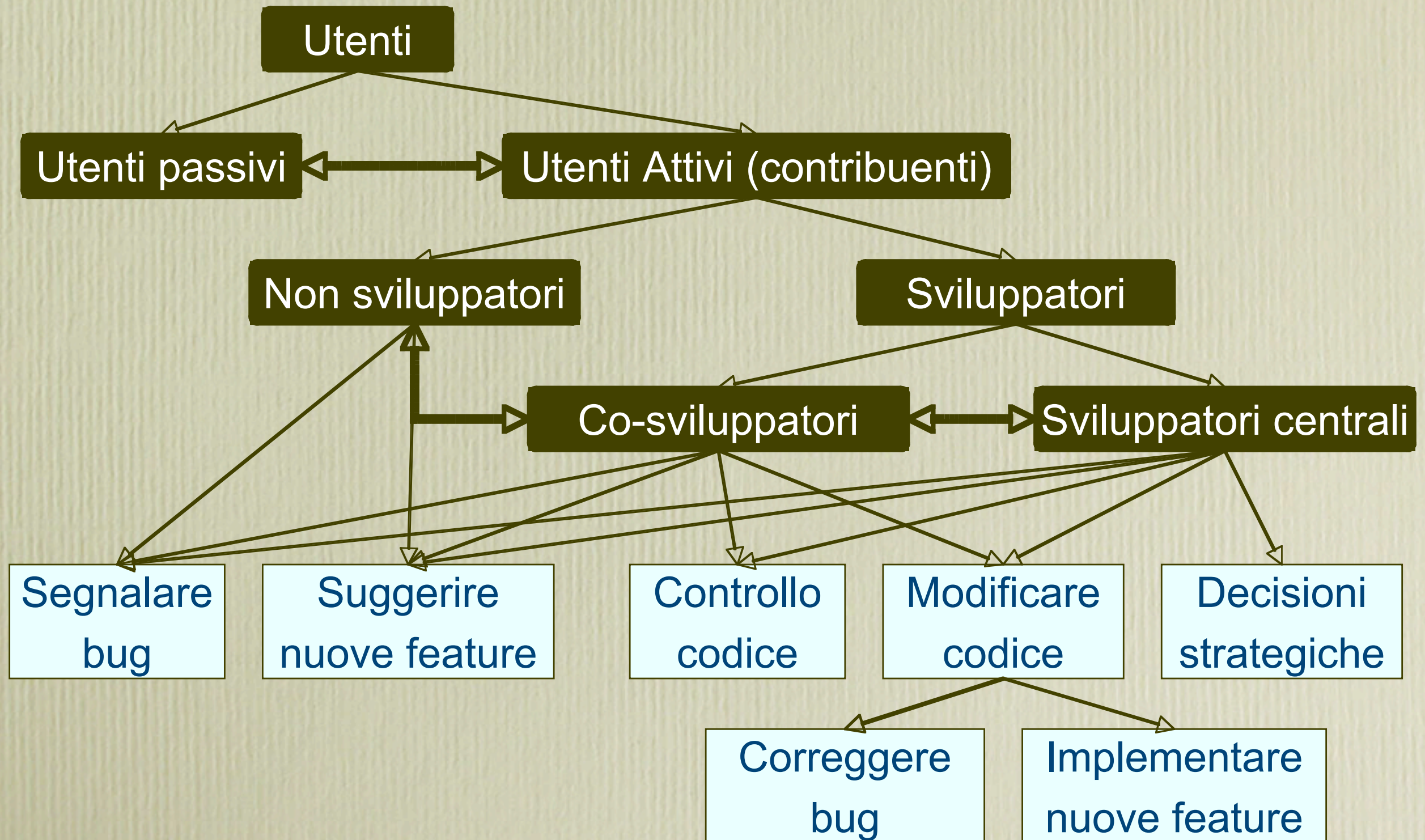
Idea di base:

- Tanto più facilmente l'informazione circola, tanta più gente ci può lavorare sopra, e quindi tanto più rapidamente:
 - si produce codice
 - si scoprono errori,
 - si ha il feedback dagli utenti
 - si individuano nuove necessità
- (È così che funziona la ricerca)

Il modello di sviluppo a “codice aperto” (Open Source)

- Basato sullo scambio reciproco e paritario di informazioni e conoscenze tra sviluppatori
- I codici sorgenti **devono essere** disponibili a chiunque voglia accedervi
- I contribuenti sono sempre anche utenti
- I contribuenti devono avere un riconoscimento (spesso non economico) per i loro sforzi

Utenti, sviluppatori e attività



Le licenze Open Source

- Una licenza Open Source è **necessaria** per garantire il modello di sviluppo O.S.

viene **negata** a **chiunque** la possibilità di **sfruttamento esclusivo** del lavoro
(proprio o altrui)

- Una licenza Open Source è un contratto a tutti gli effetti di legge, tra sviluppatori e utenti...
- ...ma soprattutto **tra gli sviluppatori**



Open Source Initiative

- OSI (1998): associazione no profit che promuove gli aspetti positivi del software libero, sul piano pratico e business-oriented, prescindendo da posizioni ideologiche
- Invece di specificare una o alcune licenze “ufficiali” (come fa la FSF con GPL/LGPL), la OSI ha dei **criteri generali** per **riconoscere** le licenze e le distribuzioni “open source”
- Questo permette di scegliere (o proporre) la licenza OS più adeguata ad ogni situazione

Open Source Definition

- 10 condizioni che disciplinano aspetti informatici, economici e giuridici della metodologia O.S.
 - Le prime 3 garantiscono le libertà fondamentali
 - Le rimanenti 7 garantiscono l'assenza di discriminazioni e i diritti e dignità degli autori
- Attualmente (luglio 2007) la OSI riconosce **59** licenze conformi a OSD
 - Corrispondono a diversi modelli di business, con differenti gradi di *controllo generativo*

I tre principi fondamentali

1. Di distribuzione Aperta (Libera e Gratuita)

- La Licenza non può impedire di vendere o donare i programmi, anche aggregati ad altri di altra origine. Non si può chiedere royalties.

2. Di Accesso Aperto (al Codice Sorgente)

- Il codice sorgente deve essere disponibile a costo di distribuzione (~0), e non obnubilato

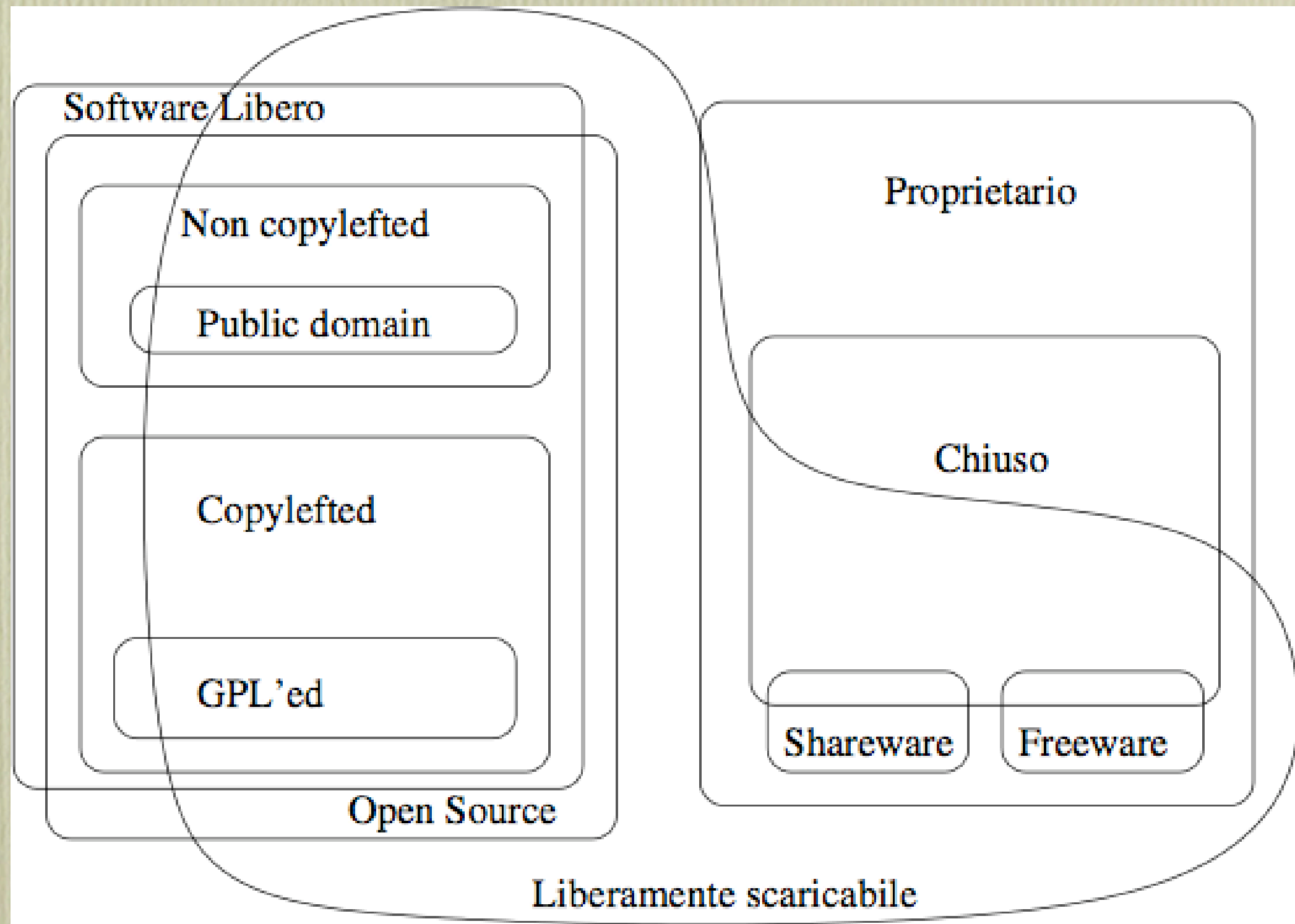
3. Di Modificabilità Aperta

- Deve essere permesso modificare il codice e la realizzazione di lavori derivati, che possono essere distribuiti alle stesse condizioni del software originale

Le altre 7 condizioni

1. Integrità del Codice Sorgente dell'Autore
2. Nessuna discriminazione contro persone o gruppi
3. Nessuna discriminazione contro campi applicativi
4. I diritti offerti dalla Licenza si applicano automaticamente a tutti i destinatari
5. La Licenza non deve essere specifica ad un Prodotto
6. La Licenza non deve porre vincoli su altro software (event. con altre licenze) allegato a quello licenziato
7. La Licenza non deve richiedere particolari tecnologie di accesso

OS e altre categorie di software



Adattato da <http://www.gnu.org/philosophy/categories.html>

Differenze tra le varie categorie

	Open Source o Free		Proprietario		
	Non copyleft	Copyleft	Chiuso	Shareware	Freeware
Codice sorgente disponibile	Si	Si	No	No	No
Libertà di redistribuzione ... di modifica ... di restrizione	Si	Si	No	Si	Si
	Si	Si	-	-	-
	Si	No	No	No	No
Versioni modificate sono sempre libere	No	Si	-	-	-
Liberamente scaricabili	Si	Si	No	Si	Si
Limitazioni d'uso temporali	No	No	No	Si	No
Possibilità di ritorno economico	Si	Si	Si	Si	No

Alcune licenze OSI

Licenza	Si estende al lavoro derivato?	Si estende alle modifiche?	Può essere chiusa?
BSD, MIT (X)	No	No	Sì
GPL	Sì	Sì	No
LGPL	No	Sì	No
Mozilla	No	Sì	Sì
QPL (Qt)	Sì	No	No

BSD, MIT (X)

- Molto semplici e generiche (risalgono agli '80)
- Codice licenziato con BSD, può essere successivamente “chiuso” sotto altre licenze
- Stesso dicasi per le modifiche fatte su codice
- Esempi: server X, codice Unix FreeBSD
- Licenze simili: Apache, IBM Public, Nokia...

Gnu Public License (GPL)

- Si estende automaticamente ai programmi contenente codice GPL (anche in minima parte)
- Codice GPL modificato si deve rilasciare sotto GPL (“copyleft”)
- Non può dare luogo a codice o pacchetti distribuiti solo con altre licenze
- Esempio: kernel Linux (GPLv2, non v3)

Free Software (Software Libero)

- Secondo la filosofia della FSF, il sw deve consentire ai suoi utenti 4 libertà fondamentali:
 - di eseguire il programma, per qualsiasi scopo
 - di studiare come funziona il programma e adattarlo alle proprie necessità
 - di ridistribuire copie senza oneri
 - di migliorare il programma e distribuirne pubblicamente i miglioramenti
- Implementate nella GNU Public License (GPL)

Lesser General Public License

- Pensata per le librerie (funzioni linkabili)
- Come GPL, ma convive con altre licenze: il software principale può essere rilasciato sotto qualsiasi licenza (anche proprietaria)
- Modifiche al codice LGPL sono comunque soggette a LGPL
- Esempio: librerie (zlib)

GPL v3 (29/06/07)

- Chiude alcune “falle” lasciate dalla GPLv2, aumentando le restrizioni sulla distribuzione
 - Rende praticamente impossibile l’uso di DRM
 - p.e., *tivoization* e *treacherous computing* sono permessi da GPLv2 ma non da GPLv3
 - Impedisce che si paghi per mettersi al riparo da *patent infringements* (v. Novell-Microsoft)
- Semplifica alcuni aspetti riguardo la (in)compatibilità con altre licenze

GPL v3 (29/06/07)

- Ritenuta da diverse parti troppo restrittiva
 - pare inapplicabile in molte situazioni “sensate” (p.e., videogames, software di controllo di dispositivi)
 - Rischia di ridurre il bacino di sviluppatori
- p.e. il kernel Linux stesso rimarrà sotto GPLv2.

“I refuse to play politics with Linux, and I think you can use Linux for whatever you want to--which very much includes things I don't necessarily personally approve of.” (Torvalds riferendosi a DRM)

Mozilla Public License

- Ibrido tra GPL e BSD
- “Covered Code”: il codice soggetto alla licenza, che è simile a GPL
- “Larger Work”: codice MPL può essere combinato con altro codice (anche proprietario)
- Le modifiche al Covered Code vanno sotto MPL
- Licenze derivate: Apple PL, Nokia OSL, Sun PL

Q Public License

- Distingue tra il Software e le Modifiche, e come queste sono distribuite
- Se le modifiche sono incorporate nel codice, allora si applica la QPL anche ad esse
- Se le modifiche sono distribuite sotto forma di Patches, possono essere soggette a qualsiasi licenza (anche proprietaria)
- Esempio: librerie Qt (che sono anche GPL)

(Non) Assicurazioni

- Spesso le licenze (open e closed) contengono dichiarazioni su assicurazioni sul prodotto
- Non necessario; non considerate dalla OSD
- Tipici esempi di (non) assicurazioni
 - Commerciabilità: il prodotto funziona come ci si aspetta implicitamente
 - Adeguatezza: ... a precise necessità
 - sulla proprietà intellettuale
 - sui danni diretti ed indiretti

Modelli di business con Open Source

- Support Seller
 - Si vende il servizio sul software (distribuzione, branding, assistenza post-vendita, formazione...) (RedHat)
- Loss-leader
 - L'azienda cerca di recuperare spazio in un mercato dominato da concorrenti proprietari (Netscape, Borland)
- Sviluppatori commerciali
 - OS come piattaforma per altri prodotti (Apple, SAP)
- Abilitatore di servizi
 - La piattaforma Open Source serve per accedere ad un servizio on-line a pagamento

Modelli di business con Open Source

- Produttori di hardware
 - Il software abilitante (driver) è spesso un costo, non è il core business
- Accessoristi
 - Macchine preinstallate, libri, gadget (O'Reilly, Lindows, VA Linux)
- “Vendilo, e poi liberalo”
 - Un prodotto inizialmente chiuso viene rilasciato aperto quando appropriato (es., end-of-life)

Un controesempio: Licenze MS Shared Source

- Microsoft propone 8 licenze Shared Source
- Tutte **vietano** di ridistribuire il codice, o di condividere il sorgente con terze parti
- Le licenze per enti commerciali o governativi sono “read only”, e non consentono la modifica neanche per uso interno (ma si può proporre le patch a MS)
- Il ricevente deve trattare il sorgente come dati proprietari confidenziali di Microsoft
 - Questo significa che uno sviluppatore che entra in contatto con il sorgente Microsoft è *tainted* (infetto): se successivamente sviluppa programmi competitivi può essere citato per violazione del copyright

Quale licenza scegliere?

- Spesso non è una scelta: la licenza viene ereditata dal software di partenza (p.e. GPL)
- Può essere conveniente rimanere O.S. anche se si ha la possibilità di “chiudere” il software (vedi IBM con Apache e Apple con Darwin)
- Se lo scopo è attirare sviluppatori DI un dato software, conviene una GPL/LGPL
- Se lo scopo è attirare sviluppatori SU un dato software, allora Apache/Mozilla/Qt vanno bene.

Conclusioni

- Open Source è un modello di sviluppo innovativo, mirante alla qualità e l'affidabilità del software attraverso la revisione indipendente tra pari e l'evoluzione rapida del codice sorgente.
- Per essere “open source”, il software deve essere distribuito sotto una licenza che garantisca il diritto di leggerlo, ridistribuirlo ed eseguirlo liberamente.

Conclusioni

- Esistono molte licenze conformi a OSI
 - Molti aspetti in comune (e.g., le libertà fondamentali)
 - Variano aspetti manageriali, organizzativi, e di controllo generativo
- Molti modelli di business possono essere implementati con vantaggio adottando il modello Open Source

Riferimenti

- <http://www.opensource.org/>
- <http://www.gnu.org/philosophy/philosophy.html>
- <http://www.mozilla.org/docs>
- C. Dibona, M. Stone, S. Ockman, Open Sources: Voices from the Open Source Revolution, O'Reilly, 1999
- C. Gacek, T. Lawrie and B. Arief, Interdisciplinary Insights on Open Source, Proc. of the Open Source Software Development Workshop, pp. 68-82, University of Newcastle, UK, 2002.
- G. Moody, Rebel code: Linux and the Open Source Revolution, The Penguin Press, 2001
- E. S. Raymond, The Cathedral and the Bazaar, O'Reilly, 1999
- A. M St. Laurent. Understanding Open Source & Free Software Licensing. O'Reilly, 2004