

Open Source e Riuso

Lezione 4bis



Termini del problema (i)

Il mercato della produzione del software si classifica in base a tre fattori:

Proprietà intellettuale del software:

software proprietario – il titolare detiene i diritti di sfruttamento economico

software non proprietario – il titolare condivide i diritti di IP

Strategia commerciale:

software commerciale – software a pagamento

software non commerciale – software gratuito

Prodotto venduto:

codice aperto – codice sorgente e codice oggetto

codice chiuso – solo codice oggetto

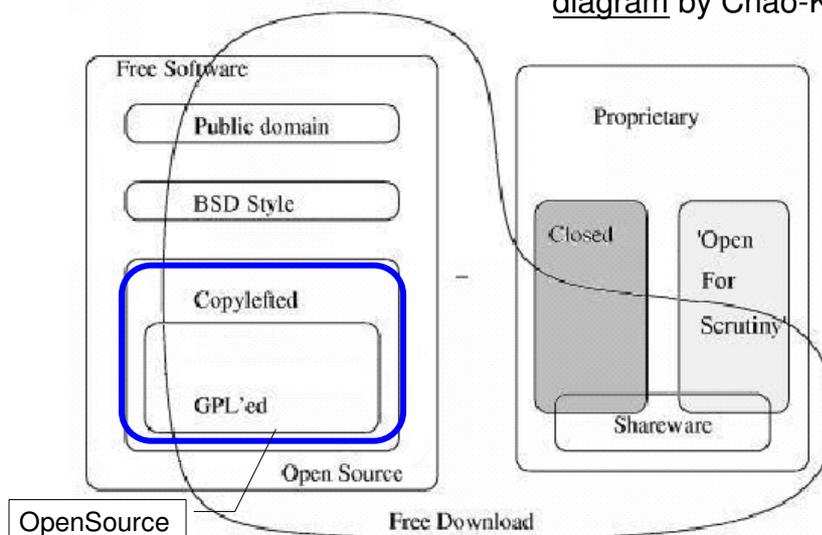
Termini del problema (ii)

Lo scenario è quanto mai complesso per diversi motivi:

- si intrecciano definizioni “ideologiche” con definizioni giuridiche o tecnologiche
- sul modello “astratto-teorico” si innesca il modello concreto-fattuale (i casi concreti)
- il confine fra le definizioni spesso non è così chiaramente demarcato
- la pratica spesso crea modelli “ibridi” non previsti dalla categorizzazione teorica

Scenario

diagram by Chao-Kuei



Definizioni: software proprietario

Software proprietario

- software sviluppato da una o più *soggetti* (imprese, enti o individui) i quali detengono i diritti di proprietà intellettuale, di sfruttamento commerciale, di utilizzo del software
- gli utenti/clienti possono utilizzare tale software mediante:
 - licenza d'uso
 - apposito contratto ad oggetto informatico
 - concessione d'uso gratuita da parte del titolare dei diritti
- il software proprietario è rilasciato in formato eseguibile, raramente in codice sorgente
- il software proprietario non può essere copiato, modificato, distribuito senza l'autorizzazione dal titolare dei diritti
- ogni utilizzo ulteriore del software da parte di terze parti deve essere regolato tramite **contratti di distribuzione o di utilizzo** fra il proprietario e le parti

Definizioni: software proprietario

Software proprietario secondo i tre parametri introdotti si colloca come segue:

- ad esso sono connessi diritti di proprietà intellettuale e di sfruttamento economico in capo ad uno o più *soggetti*
- può essere rilasciato sia a pagamento sia gratuitamente
- può essere rilasciato a codice aperto (codice sorgente)
- oppure rilasciato a codice chiuso (codice oggetto) per tutelare l'oggetto commerciale ossia la forma rappresentativa del codice sorgente

Definizioni: shareware/freeware

Shareware /Freeware

- appartiene alla categoria del software proprietario
- software distribuito con licenza gratuita
- disponibile solitamente solo l'eseguibile e non il codice sorgente
- è fornito in modalità gratuita con usi di prassi limitati
 - nelle funzioni (es:tutte tranne il salvataggio, tutte tranne la stampa)
 - negli usi (es: scopi didattici, dimostrativi, di ricerca)
 - nel tempo (es: solo per 30 giorni)

Scopo di questo software:

- motivi di pubblicità: demo, trial
- motivi divulgativi o di ricerca
- motivi di essere presenti sul mercato: entrare nelle analisi di mercato

Definizioni: public domain

Software public domain

- senza licenza - gli autori non sono tutelati
- codice accessibile a tutti
- tutti possono modificarlo, copiarlo, usarlo
- tutti possono distribuirlo in modalità diverse rispetto alle volontà dell'autore
e.g. aggiungere e distribuire modifiche proprietarie non più disponibili in codice sorgente, limitando quindi la redistribuzione
- viene costruito senza seguire particolari regole etiche o codici metodologici

Padri fondatori: free-software-fondation

- Anni '80 **Richard Stallman** - lancia il progetto GNU per la creazione di un sistema operativo Unix libero e circolabile- fonda la *Free software fondation*
- **free software** è più una filosofia legata alla libera circolazione del codice sorgente
- GNU="Gnu's Not Unix"
<http://www.gnu.org/>
- *Free as in freedom*
"Free software is a matter of freedom: people should be free to use software in all the ways that are socially useful."
- "The Free Software Foundation (FSF), founded in 1985, is dedicated to promoting computer users' right to use, study, copy, modify, and redistribute computer programs."

Definizioni: free software

- **Free software**
 - Nel 1985 **Richard Stallman**, ex ricercatore del MIT, fondò la **Free Software Foundation (FSF)**
 - codice sorgente disponibile
 - GPL base: possibilità di distribuirlo, copiarlo, modificarlo, usarlo GNU-*General Public Licence*
 - LGPL – *Lesser General Public Licence*

I quattro gradi di libertà del movimento free software

- (freedom 0)
 - Libertà di eseguire il programma per ogni scopo
- (freedom 1)
 - Libertà di studiare come il programma lavora e di adattarlo alle proprie esigenze. L'accesso al codice sorgente è un prerequisito per questo.
- (freedom 2)
 - Libertà di ri-distribuire copie in modo da aiutare il tuo "vicino"
- (freedom 3)
 - Libertà di migliorare il programma e di rilasciare i miglioramenti alla comunità per il beneficio pubblico. L'accesso al codice sorgente è un prerequisito per questo

Definizione: open source

Open Source

- **fondato da Bruce Perens**
- **codice sorgente disponibile**
- **modifica della licenza GPL di base per consentire la commercializzazione**
- **possibilità di commercializzare le distribuzioni derivate dal codice sorgente specificando l'autore e sgravando così l'autore precedente di eventuali responsabilità**
- www.opensource.org

Caso LINUX

- 1991 **Linus Benedict Torvalds** - lancia il progetto di creare la prima versione di un sistema UNIX per il processore 80386 ossia per PC

Annuncio fatto al newsgroup UseNet

**"Hello everybody out there using minix -
I'm doing a (free) operating system (just a
hobby, won't be big and professional like
gnu) for 386(486) AT clones."**

- 1994 esce la prima versione definitiva di LINUX 1.0
- 1997 Transmeta
- 2003 Open Source Development Lab

Open Source Definition: modifica della GPL di base

1. **Libera distribuzione**
2. **Codice sorgente**
3. **Prodotti derivati** - permettere di derivare altri prodotti sempre sottostanti alle stesse modalità di licenza GPL
4. **Integrità del codice sorgente dell'autore** - consentite solo modalità patch che modificano il codice dell'autore durante la compilazione
5. **Nessuna discriminazione** per persone e gruppi
6. **Nessuna discriminazione** per tipi di applicazioni
7. **Distribuzione della licenza**
8. La licenza **non** deve essere **specificata** per un prodotto
9. La licenza **non** deve essere **restrittiva** di altri software
10. Deve essere **neutra** rispetto a tutte le scelte tecnologiche

Copyleft

Copylefted sw: è *free software* le cui clausole di distribuzione e uso non permettono di aggiungere restrizioni nelle modifiche successive del software. Questo per garantire che ogni copia, modifica, distribuzione uscita da questo software mantenga ancora lo stesso grado di **circolabilità**.

Nessun copyright è permesso neppure sulla parte nuova. (OSS) – Il copyleft non significa “senza diritti” ma significa che i diritti permettono le 4 libertà.

Non-copyleft

Non-copylefted sw: è il software free le cui clausole di distribuzione permettono a chi modifica, copia, ridistribuisce il software di non mantenere lo stesso grado di libertà ossia di riservarsi dei diritti di tutela di quanto svolto.

Permette un grado di reintroduzione del copyright sulla parte nuova svolta o modificata.

(Public Domain, semi-free)

Perché delle Distribuzioni

- Spesso le imprese distribuiscono software open source in formato amichevole o corredando il software di base (kernel) di programmi aggiuntivi (sempre open source) che garantiscono alcune migliorie:
 - stabilità del codice
 - performance ottimizzate sulla singola piattaforma hardware
 - installazione veloce ed amichevole
 - ottimizzazione di tutti i moduli
 - assistenza
 - manutenibilità degli aggiornamenti e degli errori

Il modello di business cambia

- L'OOS impone quindi di modificare il modello di business dell'industria del software.
- Il modello classico di business è basato sulla licenza: l'impresa di software investe con risorse interne per produrre il prodotto finale e mediante il prezzo della licenza rientra dell'investimento in un lasso di tempo di 3-5 anni.
- Il modello basato sull'OOS non può più basarsi sulle *revenue* delle licenze. Quindi deve basare la sua politica di profitto **sui servizi, sulle consulenze, sulle personalizzazioni.**
- L'industria del software è pronta a questa trasformazione?
- La normativa sì---
- Codice della PA digitale – art. 69 sul riuso
- Direttiva Stanca 19/12/2003

Pros

- Circolabilità della conoscenza
 - Pari opportunità di accesso al software per i paesi/aziende/cittadini/utenti svantaggiati
 - Circolabilità delle idee nella ricerca
- Maggiore qualità del software
 - community di alto valore tecnologico
- Maggiore trasparenza
 - privacy
 - sicurezza
 - virus &c.
- Riduzione dei costi di acquisto del software (?)
 - Minimizzare il potere delle multinazionali di software
 - Minimizzare l'utilizzo di brevetti relativi al software
 - Smantellamento del modello "licenza d'uso"

Cons

- Struttura non gerarchica ma a rete distribuita in tutto il mondo: mancanza di efficienza, efficacia, tempestività specie per produrre nuove funzionalità legate al *mercato*
- "Non vi è una "mente" strategica e tattica
- Non vi è un "referente" chiaro per le modifiche apportate e per l'assistenza/manutenzione
- Incerti tempi di risposta della *community*
- Alti costi di produzione delle *personalizzazioni* software
- Alti costi di personale specializzato in queste tecnologie
- Il modello di business sposta il costo di acquisto sul costo di manutenzione ossia dalla licenza al costo del personale specializzato per la gestione di tale software

Attenzione

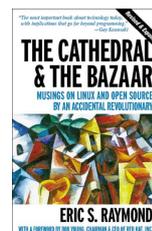
- **Codice aperto** non vuole dire **open source**
- **Standard aperto** non vuol dire **open source**

Non è detto che le due definizioni coincidano

- Codice aperto → può significare la facoltà di distribuire il codice sorgente e nulla imporre al fornitore in termini di cessione della titolarità (*licenza classica ma con rilascio del codice sorgente*)
- Codice aperto → può significare codice aperto + licenza che consenta le 4 libertà fondamentali, a fini specifici, ma non la cessione della titolarità (*simil-GPL*)
- Codice aperto → può indicare la mera traduzione in italiano della dicitura inglese *open source* ossia con le 4 libertà e quindi di fatto spogliarsi della titolarità (*GPL*)
- La normativa italiana non è chiara a questo riguardo: occorre interpretare dal contesto.

Sintesi

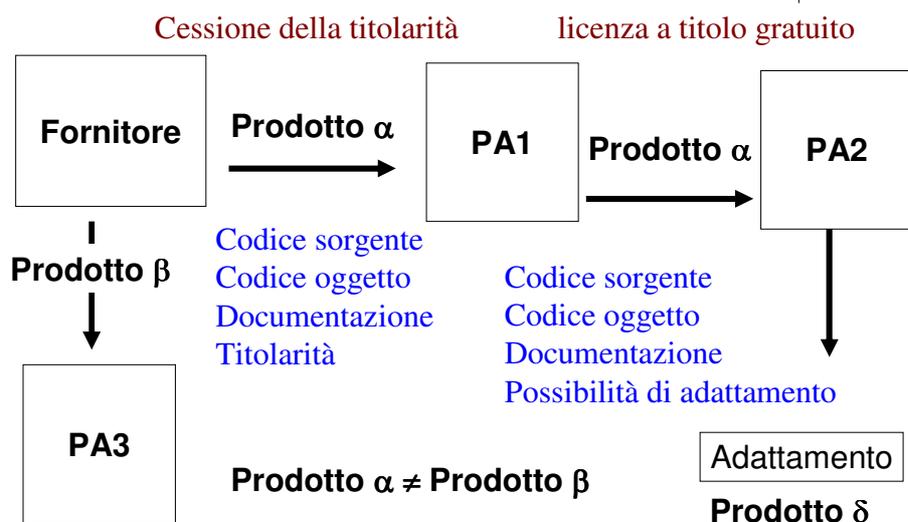
- Creare un *modello di business* sostenibile per tutelare l'idea di base dell'open source e creare un modo nuovo di fare mercato del software senza ritornare all'era dell'artigianato
- Sostenere l'industria del software nel passaggio senza uccidere l'industria stessa
- Fare profitto senza uccidere i principi fondamentali dell'open source
- *Eric Raymond "The cathedral and bazaar"*



RIUSO: Obblighi per la PA in materia di sviluppo software

- Per la PA vi sono norme che OBBLIGANO le amministrazioni a richiedere ed ottenere la titolarità dei software “ad hoc”
- ex Art. 5, Direttiva del 19 dicembre 2003 “Sviluppo ed utilizzazione dei programmi informatici da parte delle pubbliche amministrazioni” – Circolare Stanca – (Gazzetta Ufficiale n. 31 del 7-2-2004)
- e a rilasciare in “uso gratuito” tale software alle altre PA (ex Art. 69, D.lgs. 82/2005 – CAD – Codice dell’Amministrazione Digitale)
- Introduzione del concetto di RIUSO

Possibile scenario di riuso



Il riuso e la crisi del modello di business (1/2)

- Il riuso del software nella PA è stato introdotto per:
 - razionalizzare la spesa pubblica in tema di servizi informatici
 - incentivare il riuso piuttosto che duplicare gli acquisti
 - rendere autonome le PA di poter modificare, integrare, aggiornare i software senza un legame vincolante con il fornitore
- Riuso è spesso assimilabile una licenza proprietaria a titolo gratuito con permessi di rilascio del codice aperto e la garanzia delle libertà fondamentali (semi-GPL)

Il riuso e la crisi del modello di business (2/2)

- Il riuso accompagnato al fenomeno dell'open source ha fortemente modificato l'industria del software che da un modello basato principalmente sulla licenza d'uso a pagamento (es. Microsoft) è passata a due nuovi modelli di business:
 - open source – basato principalmente sulla vendita di personalizzazioni e di giornate uomo di assistenza
 - servizi ASP – *application service provider* vendita di servizi via rete (es. GoogleDocs, hosting di software, etc.)
- Sulla base di questi due nuovi modelli di business sono nati nuovi strumenti giuridici di tutela del software
 - GPL e LGPL – licenze dell'open source
 - Contratti/Appalti di servizi ASP

Creative Commons

<http://creativecommons.org/about/licenses/meet-the-licenses>



- Nuova forma di tutela il diritto d'autore delle opere dell'intelletto favorendo nel contempo la circolabilità della conoscenza
- Vi sono 6 forme di tutela per attivare delle restrizioni sull'utilizzo del prodotto dell'ingegno
- non utilizzo commerciale NC
- no opere derivate ND
- stesso grado di tutela SA – Share-Alike



Choose by-nc-nd license



Choose by-nc-sa license



Choose by-nd license



Choose by-nc license



Choose by-sa license



Choose by license

Domande

- Cosa è l'open source
- Cosa è il copyleft
- Cosa è il riuso
- Differenze fra l'open source e il free software
- Differenze fra public domain e open source
- Differenze fra open source e riuso