

# Applicazioni libere nelle scuole

**Libera conoscenza e software libero,  
introduzione e motivazioni del corso**

**Giovanni A. Cignoni**

**Pisa, aprile-maggio 2009**

- **Assunzioni e motivazioni di fondo**
- **Peccati originali ... e vie di redenzione**
- **Miti, leggende e una prospettiva laica**
  - **Filosofia:** etica e interessi industriali
  - **Licenze:** fondamenti e trappole
  - **Prodotti:** quantità e qualità
  - **Processi:** cattedrali e bazaar
  - **Economie:** risparmi e sostenibilità
- **Organizzazione del corso**

# Assunzioni e motivazioni di fondo

- **A proposito dell' *open source***
  - Cosa possiamo assumere noto
  - Almeno, genericamente e comunemente noto
- **Luoghi comuni**
  - È gratis, è il futuro, è migliore, è di moda, è una moda
  - È equo e solidale, è politicamente corretto
  - Non è M\$
- **Ne discuteremo meglio, diamoli per veri**
  - Sono comunque un'opinione comune radicata
  - Se ne parla, è difficile ignorarlo

- **Saper usare le applicazioni comuni**
  - Disegnare, scrivere, fare i conti, fare ricerche
  - Competenze di base che la scuola deve insegnare
- **C'è modo e modo, spesso**
  - Viene proposto solo software commerciale
  - Le famiglie sono condizionate ad acquisti costosi
  - Risorse (già scarse) destinate all'acquisto di licenze
- **Giudizio, inevitabile e impietoso**
  - Scuola compromessa, elitaria, sprecona e antiquata
  - Consapevolezza dei più svegli e attenti

- **Bambini svegli**
  - E genitori informatici appassionati
- **Un Comune**
  - Con un assessorato alla promozione delle tecnologie digitali per la formazione
- **Un Centro di Competenza**
  - Per dare supporto alle politiche pubbliche di adozione del software libero
- **Un'Associazione**
  - Punto di riferimento nel territorio per utenti e sostenitori del software libero

# Il peccato originale e la via per la redenzione

- **Primi programmatori**
  - Parlano il linguaggio delle macchine
  - Il software è solo binario e non circola
- **Subito dopo**
  - Assembler e Fortran, per cominciare
  - Il linguaggio si arricchisce
  - Appaiono i compilatori (al posto dei minutatori)
  - I programmi (sorgenti) sono ricchi di informazione
  - Variabili, etichette, procedure, commenti
  - I compilatori buttano via tutto
  - Ma la praticità dei binari è evidente

- **Per un bel pezzo**
  - Il software è poco, locale e personalizzato
  - Gli affari si fanno affittando/vendendo le macchine
- **Ma, a un certo punto**
  - Le macchine diventano diffuse, versatili e standard
  - Le applicazioni possibili sono tante
  - Una macchina, tante applicazioni: l'affare si sposta
  - Produrre software diventa un'attività indipendente
  - Costi di infrastruttura bassi
  - Costi di replicazione praticamente nulli
- **Una prospettiva imprenditoriale promettente**

- **Proteggere il prodotto software**
  - Per venderlo in più copie (che ci costano poco)
  - Per proteggersi dalla concorrenza
- **Gran cosa i binari**
  - La forma di codifica più facile e immediata
  - Neanche la peggiore (dipendenze hw, brevetti)
- **Come sempre**
  - Legislatori disattenti, incompetenti e lenti
  - Diritti lesi, dei più e probabilmente del sistema
  - Danno istituzionalizzato e legalizzato
  - Costituzione di inerzie economiche

- **Il software è solo un caso**
  - Legare un'informazione, immediatamente utile
  - A un formato mirato a limitarne e condizionarne l'uso
  - Sempre più sofisticato
- **Informazione codificabile**
  - Testi, disegni, musica, video, ...
  - Sorgenti e binari
  - Formati chiusi, non documentati
  - Legati alle applicazioni
  - Legati all'hardware
  - Legati all'utente e al tempo

- **Indubbiamente coinvolti**
- **Responsabili delle idee**
  - Da sempre pasticciano con le informazioni
  - Binari e sorgenti software sono stati solo l'inizio
  - In fin dei conti, Von Neumann ...
- **Responsabili delle tecnologie**
  - Dai compilatori
  - Ai DRM
  - Passando per linguaggi, formati e protocolli
- **I libri si copiavano meglio, e si prestavano ...**

- ... sono quelli che hanno iniziato a rimediare
- **Una concisa cronologia**
  - Primi '80, MIT AI Lab vs primi Unix proprietari
  - 1984, Gnu's Not Unix (il dilemma dello sviluppo)
  - 1985, Free Software Foundation
  - 1987, X-Windows (X11)
  - 1988-92, Microsoft vs Apple (con Xerox di mezzo)
  - 1989, GPL v. 1
  - 1992, GNU/Linux
  - 1993, Free BSD 1.0
  - 1994, Linux Kernel 1.0
  - 1998, Open Source Initiative
  - 2001, Creative Commons



# Miti, leggende e una prospettiva “laica”

## ■ Free Software Foundation

- Per prima ha posto il problema, hacker accademici
- *Free come free speech, non come free beer*
- Il software deve essere libero per un diritto etico
- Con buone motivazioni di economia sostenibile

## ■ Open Source Initiative

- Tre lustri dopo, imprenditori in difficoltà
- *Open perché contrapposto a closed*
- Il software deve essere aperto per motivi commerciali
- Con buone motivazioni di concorrenza disperata

## ■ Simbiosi curiosa in cordiale antipatia

- **Quattro libertà per il software**
  - Libertà di eseguirlo, per qualsiasi scopo
  - Libertà di studiarlo e adattarlo ai propri bisogni
  - Libertà di distribuirne copie
  - Libertà di modificarlo (migliorarlo) e redistribuirlo
- **Solo 2+2 licenze**
  - GNU General Public License 3.0
  - GNU Lesser General Public License 3.0
  - GNU Affero General Public License 3.0
  - GNU Free Documentation License 1.3
- **Duri e puri, pochi compromessi**

- **Definizione complessa**
  - Articolata in 10 punti
  - Caratteristiche della licenza
  - Evitare di parlare di libertà e di diritti fondamentali
- **Al 3 aprile 2009, 63 licenze (in crescita)**
  - Compromessi
  - Dual licensing
  - Restrizioni di scopo (contro i punti 5 e 6)
  - Free ⇒ open source, open source !⇒ free
- **Molto avvocateschimprenditoriale**

- È ovvio a chi va la nostra simpatia
- D'altra parte
  - Le posizioni dell'FSF non sempre sono pratiche
  - L'FSF non ha lo stesso potere dell'OSI
  - Senza l'OSI, l'FSF non avrebbe il sostegno che ha
- GNU/Linux è un esempio
  - Linux è solo il kernel
  - Ma è spesso citato come tutto il sistema
  - Linux funziona, ma non è un gran kernel
  - Hurd è un bellissimo progetto, ma sempre da finire

- **La GPL come archetipo**
  - Usa il copyright
  - Nullità in caso di violazione
  - Paternità riconosciuta
  - Persistente, la licenza non cambia
  - Virale, la licenza si estende ai derivati
  - Se distribuito, obbligo di disponibilità del sorgente
  - Libertà di distribuzione
- **BSD, un esempio di licenza permissiva**
  - Solo paternità, con protezione dei nomi
  - Il caso della vecchia clausola sulla pubblicità

- **Versioni diverse, licenze diverse**
  - Il copyright stabilisce il diritto di licenza
  - Versioni libere e versioni chiuse
  - Parallele o nel tempo
- **Stesso identico prodotto, licenze diverse**
  - Preamboli con *i se* e con *i ma*
  - Licenza dipendente dall'uso o dal soggetto
  - Violazioni tollerate della definizione OSI
- **Distribuzioni miste**
- **Shareware**

- **Il software libero disponibile**
  - ~200 000 progetti su Sourceforge
  - ~2 000 000 utenti su Sourceforge
  - Progetti dormienti
  - Progetti didattici
  - Progetti abortiti
- **La qualità del software è indipendente**
  - Dalla licenza
  - Dalla disponibilità del codice
- **Una buona comunità può far la differenza**

## ■ Raramente

- Più o meno quanto nel sw proprietario
- Ma, tutto sommato, più di quanto basta
- Difficile trovare aree scoperte

## ■ Alchimie di successo

- Un bravo (spesso dispotico) capo
- Un ristretto gruppo di sviluppatori-decisori
- Una comunità capace di garantire il ricambio
- Una comunità vivace e animata
- Un buon numero di utenti
- Spesso, anche evidenti interessi economici

- **Open source come processo di sviluppo**
- **Eric S. Raymond**
  - Uno dei fondatori dell'OSI
  - Sicuramente un abile comunicatore
- **La Cattedrale e il Bazaar**
  - Dichiarazione di stupore per i risultati di Linux
  - Analisi di Linux come progetto
  - Identificazione di caratteristiche di processo
  - Ripetizione su un esperimento personale
  - Tesi: le caratteristiche determinano il successo

## ■ Le caratteristiche

- Interesse personale (il genio nasce dalla necessità)
- Riutilizzo (i grandi sanno cosa riscrivere)
- Prototipazione coraggiosa (capacità di buttar via)
- Aggregazione (i problemi ti trovano)
- Capacità di abbandonare (meglio se in buone mani)
- Utenti come verificatori (molti occhi vedono meglio)
- Rilasci precoci e frequenti (per soddisfare la platea)

## ■ Sono buone prassi replicabili?

- Caratterizzano, con un po' di enfasi, un contesto
- Non definiscono un processo

- I progetti OS usano il processo bazaar
- I migliori progetti OS usano il processo bazaar
- I progetti OS non hanno norme scritte
- L'OS porterà nel mondo la “cultura del dono”
- Il mondo OS è un paradiso senza conflitti
- M\$ deve essere distrutta

- **Il software non cresce sugli alberi**
  - Produrre software è costoso
  - È un investimento
  - Il modello tradizionale si basa sui ricavi da licenza
  - Ed è un modello collaudato
- **Esiste un modello imprenditoriale FLOSS?**
  - Sì, basato sui ricavi da servizi
  - È sostenibile, ma solo se sei bravo
  - E non potrà mai, generare profitti stratosferici
- **È economicamente ecologico**

## ■ Grandi utenti

- Grandi utenti, offerte commerciali vantaggiose
- Con il FLOSS risparmi contenuti (però tutto fa brodo)
- Spesso minimi sul bilancio complessivo
- Costi di transizione sensibili
- Difficoltà di investimento
- Di cassa e di capitolo di spesa

## ■ Singoli o piccoli utenti

- La fonte ultima di profitti per il sw commerciale
- Con il FLOSS non c'è partita: il risparmio è notevole
- A meno delle copie illegali ...

- **Il sw commerciale muove soldi**
- **Il fisco ci guadagna? Il PIL cresce?**
  - **Produzione estera**
  - **Vendita via preinstallazione e registrazione**
  - **Registrazione come transazione estera**
  - **Neanche l'IVA**
- **Il FLOSS offre maggiori opportunità locali**
  - **Servizi, installazione/manutenzione/personalizzazione**
  - **Sviluppo, partecipazione alla comunità**
  - **Una delle poche vie rimaste per l'informatica italiana**


# Organizzazione del corso

## ■ Calendario

- 04.18 Scrivere,  
OO Writer
- 05.09 Far di conto,  
OO Calc
- 05.16 Raccontare,  
OO Impress
- 05.23 Disegnare,  
OO Draw e GNU Gimp
- 05.30 Divertirsi con fisica e informatica,  
Phun e StarLogo

## ■ Docenti: A. Santi e M. Ghilardi

- **Locali, orari e allarmi**
- **Attestati e frequenze**
- **Account**
- **Materiale**
- **Pagina web**
- **GULP e dopo corso**
- **Giornata del 28 maggio**

- **Non avrai altro sistema che GNU,  
Linux è uno dei suoi kernel**
- **Free Software Song** 
  - *Join us now and share the software;  
You'll be free, hackers, you'll be free (x2).*
  - *Hoarders may get piles of money,  
That is true, hackers, that is true.  
But they cannot help their neighbors;  
That's not good, hackers, that's not good.*
  - *When we have enough free software  
At our call, hackers, at our call,  
We'll kick out those dirty licenses  
Ever more, hackers, ever more.*