



Il software e il movimento Open Source



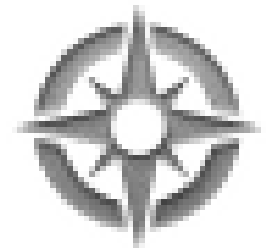
Riccardo Lancellotti

Università di Modena e Reggio Emilia

- **Le origini dell'Open Source**
 - Breve storia del mondo Unix e della filosofia che ne è derivata
- **Concetti chiave dell'Open Source**
 - Concetti di base
 - Licenze di software e contenuti
- **Chi sviluppa Open Source**
 - Sfatare alcuni miti sui nerd
- **Modelli di business Open Source**
 - Perché le aziende investono nell'Open Source
- **Qualità dell'Open Source per l'utente**
 - Perché L'Open Source può essere un'alternativa



Origini dell'Open Source



Cenni storici



1969: Progetto Multics

Cenni storici

- **MULTICS (1969)**
 - Sviluppato da: MIT, General Electric, Bell Telephone Labs
 - Sistema operativo multiuso interattivo
 - **Estremamente** complesso
- **MULTICS introduce elementi di forte innovazione:**
 - Shell
 - File system strutturati ad albero (vedi DOS 1.0)
 - Configurazione testuale
 - Semantica delle operazioni I/O
- **Non funziona in modo adeguato**



1970: PDP-7 e Unix

Cenni storici

- **Dalle ceneri di MULTICS nasce UNIX**
 - D. Ritchie, K. Thompson, 1970
 - Progetto “Space Travel” per PDP-7
- **Successo enorme tra i programmatori Bell Labs e AT&T**
 - Continuo sviluppo di software (1971-1973)
 - Linguaggio C
 - Porting del sistema operativo in C
 - Invenzione del meccanismo di “pipe”
 - Febbraio 1973: prima versione di UNIX
- **Questo entusiasmo arriverà a coinvolgere anche l'ambito universitario** (CMU, Berkeley, ..., Helsinki)



D. Ritchie



B. Kernigam



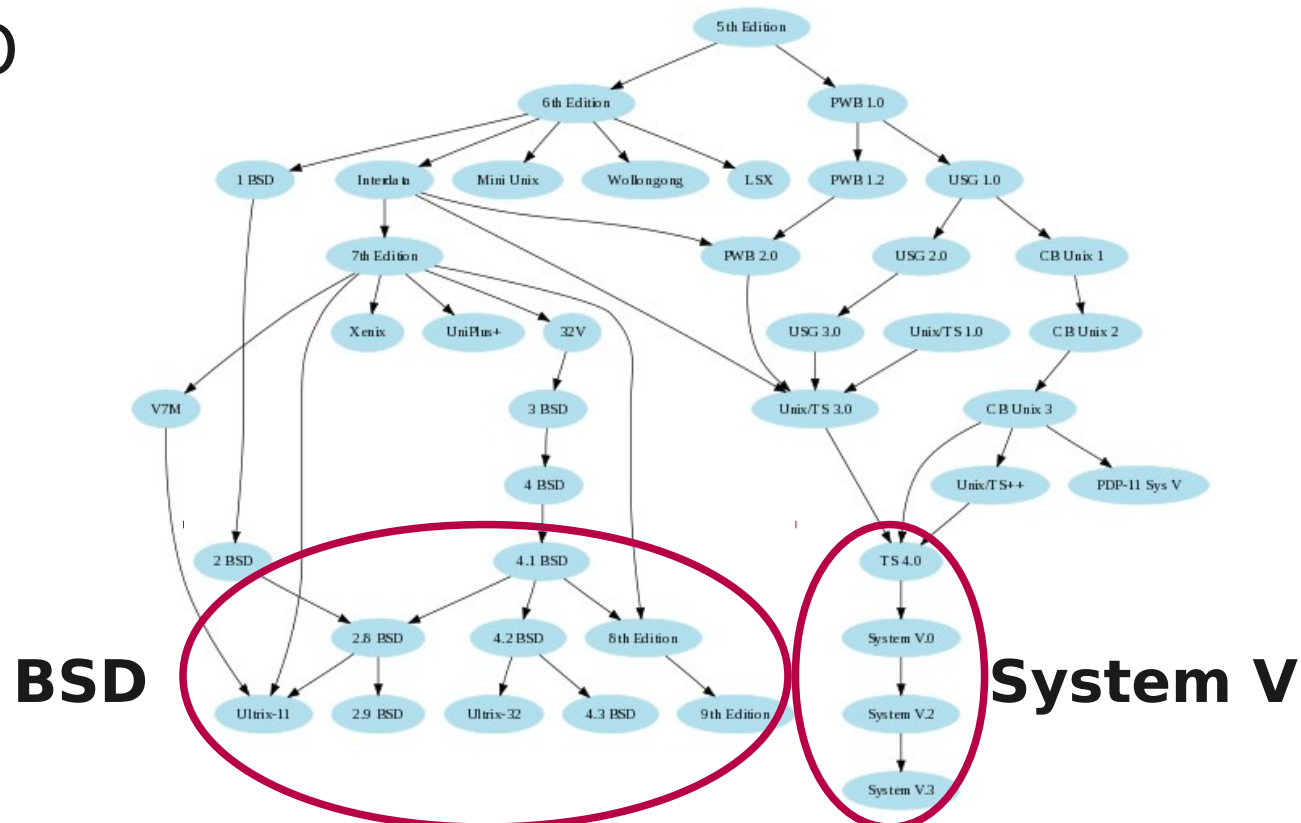
K. Thompson

Cenni storici

- **UNIX presentato ad una conferenza scientifica**
 - K. Thompson, “Symposium on Operating System principles”, Yorktown Heights, Oct. 1973
- **AT&T viene accusata di violare lo “Sherman Antitrust Act” del 1949 (1974)**
 - In base a tale decreto, AT&T non poteva fare affari in campi che non fossero la telegrafia e la telefonia
- **AT&T mise una licenza sul sistema operativo e dichiarò di non impegnarsi nel ramo “software”**
 - UNIX provided “as is”, no support

Cenni storici

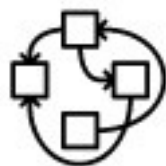
- **L'assenza di supporto ufficiale forzò gli utenti UNIX a scambiarsi informazioni**
 - Idee, manuali, programmi, bug fix
- **Tale scambio culturale portò ad una vera e propria “diaspora” di varianti UNIX**
 - AT&T, BSD, SCO
 - GNU/Linux



Alcune varianti Unix...



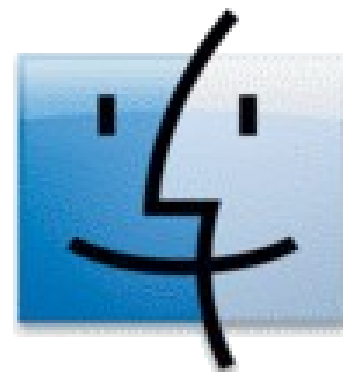
Hp/UX



HURD



IRIX® 6.5



Il progetto GNU

- **Nel 1971, un giovane ricercatore (Richard Stallman) viene assunto nel Laboratorio di Intelligenza Artificiale dell'MIT**
 - Ambiente caratterizzato da una “software sharing community”
- **Veniva utilizzato il PDP-10, dismesso nei primi anni '80**
 - Tutto il software scritto sino allora andò perso
- **A Stallman viene impedito di “riparare” il driver di una stampante laser**



Il progetto GNU

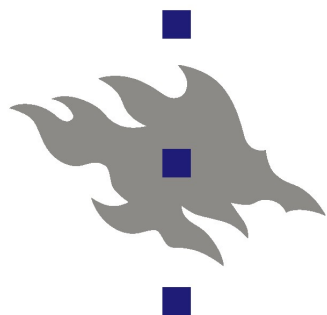
- Stallman decide di scrivere un nuovo sistema operativo compatibile con il modello UNIX
- **GNU:** GNU is not UNIX
- Nel gennaio 1984, Stallman lascia l'incarico all'MIT e scrive il suo primo software GNU, l'editor **Emacs**
- 1985: viene istituita la **Free Software Foundation**, ente caritatevole con il compito di gestire gli aspetti burocratici dietro GNU



Il progetto GNU

- **Un grande problema del progetto GNU è l'assenza di un kernel**
 - Il nucleo del sistema operativo
 - Doveva essere basato su microkernel Mach
 - Mach non era ancora pronto a quel tempo
- **L'ambiente GNU deve appoggiarsi su kernel preesistenti, limitandone di fatto l'adozione**
- **Ma forse è giunto il momento di spostarsi in Finlandia...**

1991: PC 386



UNIVERSITY OF HELSINKI

Il progetto Linux

- Nel 1991, un giovane studente universitario di Helsinki, **Linus Torvalds**, decide di scrivere il pezzo mancante (il kernel)



Il post originale...

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)

Newsgroups: [comp.os.minix](#)

Subject: What would you like to see most in minix?

Summary: small poll for my new operating system

Date: [25 Aug 91](#) 20:57:08 GMT

Organization: [University of Helsinki](#)

Hello everybody out there using minix -

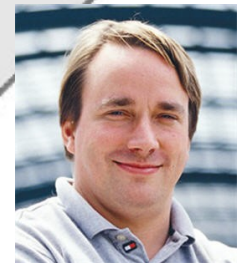
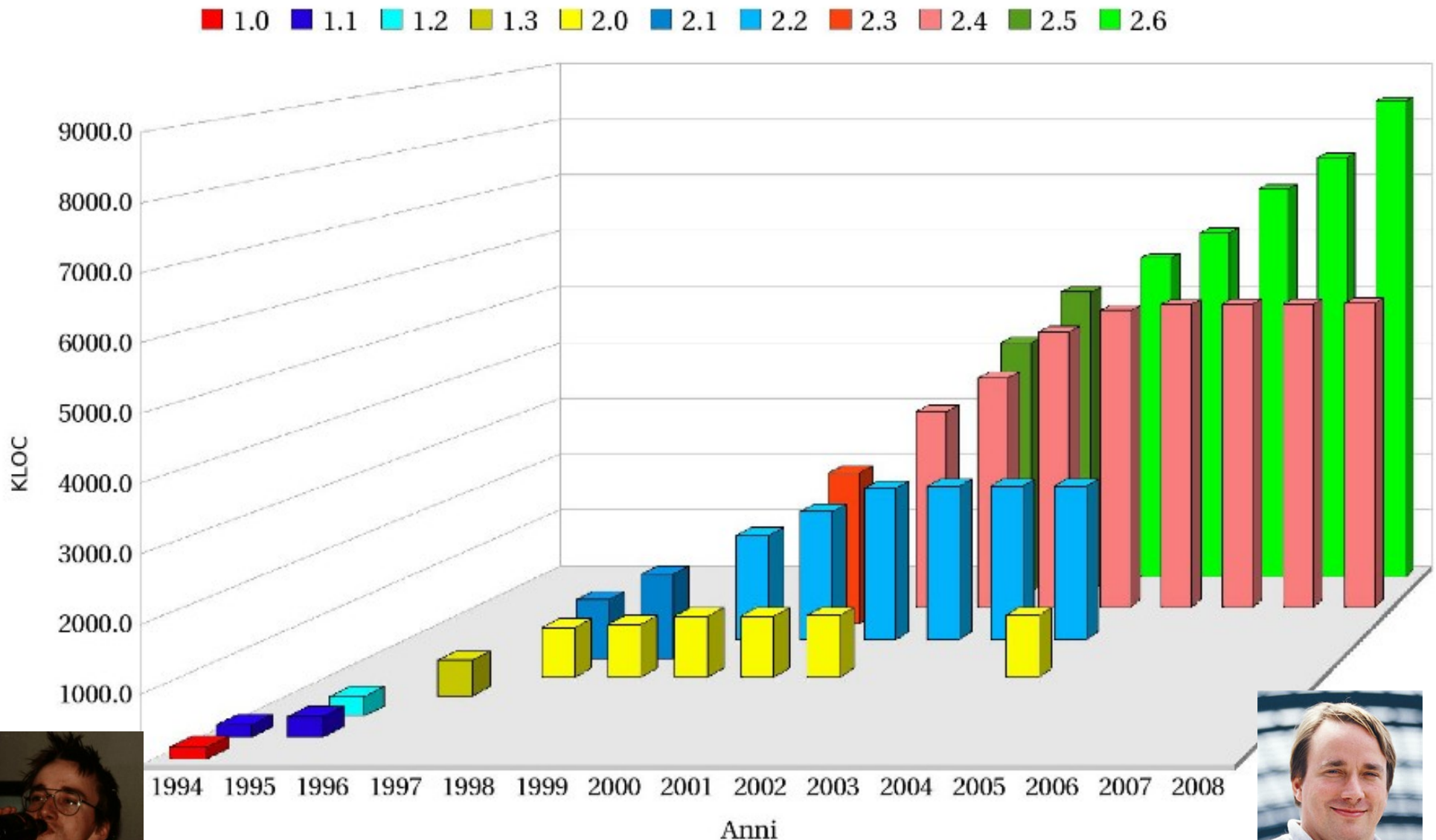
I'm doing a (free) operating system ([just a hobby](#), won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat [...]

Il progetto Linux

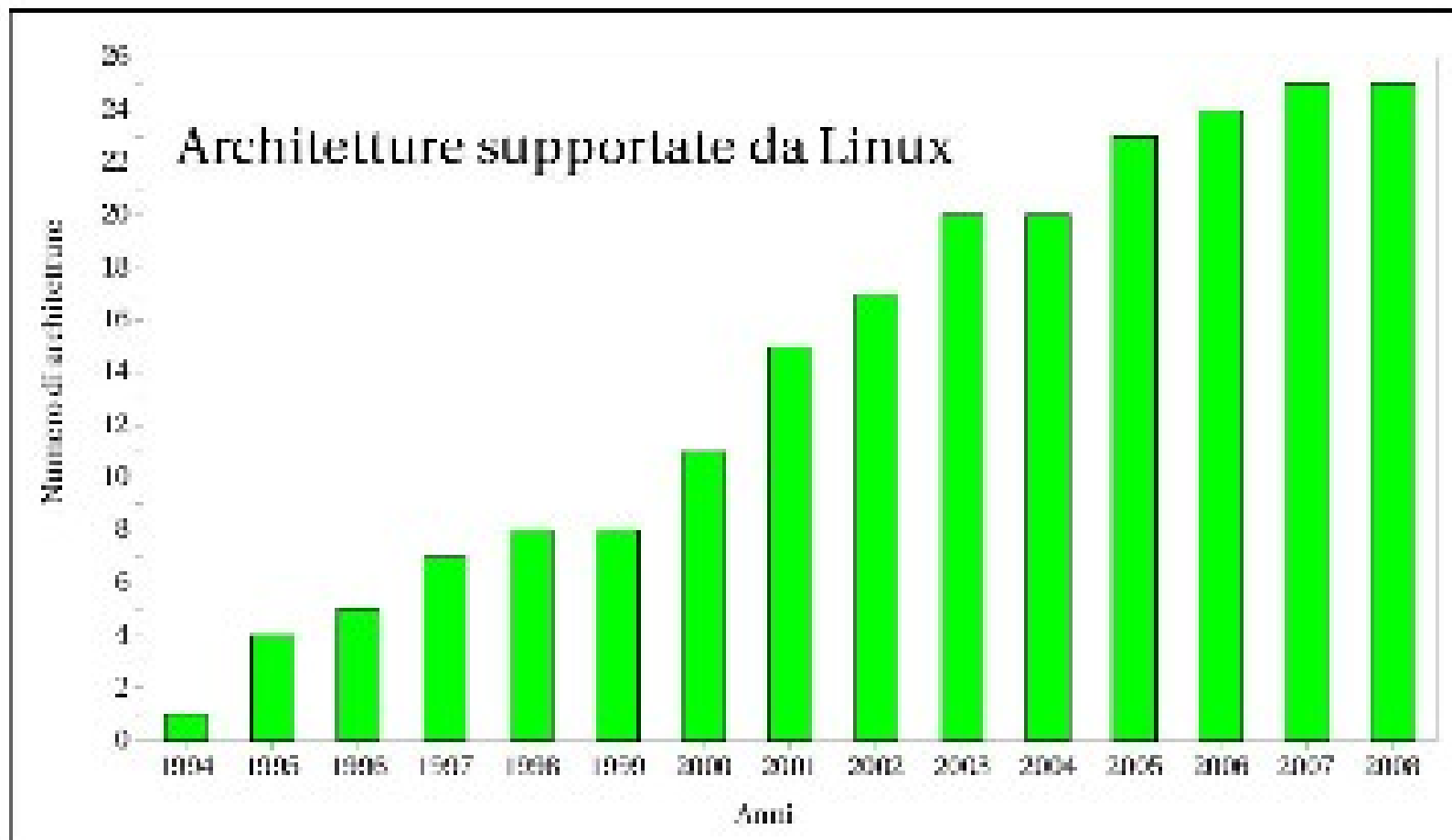
- **Un anno dopo, viene rilasciata la versione 0.99**
- **Nasce l'ambiente GNU/Linux**
- **Dalla prima implementazione si è fatta molta strada...**



Dimensione del codice negli anni

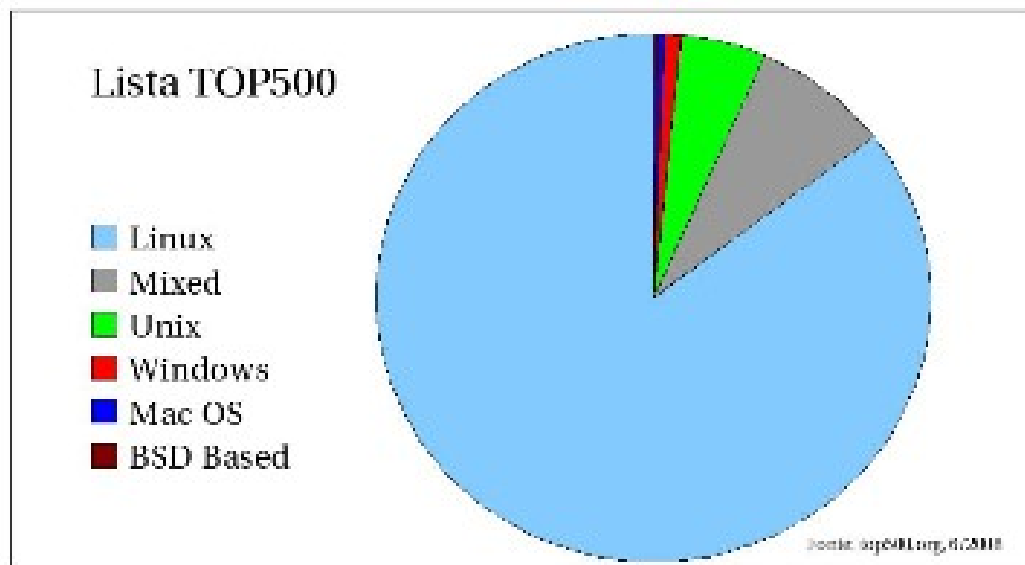


Architetture supportate

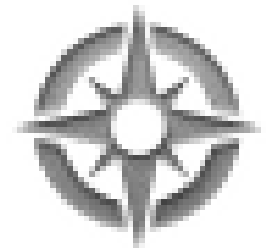


Diffusione di Linux

- **Ad oggi Linux è una scelta molto popolare in molteplici contesti**
 - Server Web
 - Sistemi di rete (es. server farm di Second Life)
 - Applicazioni scientifiche (vedi Top500)
 - Sistemi embedded (es. appliance Linksys, ebook)



Open Source e Open Content, Concetti e licenze



Proprietà intellettuale

- **La proprietà intellettuale è definita nel seguente modo:**
 - Proprietà intangibile che risulta da un'opera di creatività (brevetti, copyright)
- **Essa garantisce ai creatori di un'opera i diritti di sfruttamento univoco per un determinato periodo di tempo**
- **Quattro strumenti di protezione della proprietà intellettuale:**
 - Brevetto
 - Copyright
 - Segreto industriale
 - Marchio registrato

Proprietà intellettuale

- **Brevetto**

- Sovvenzione del governo che fornisce i diritti esclusivi di produzione, uso, vendita di una invenzione per un periodo di tempo limitato

- **Copyright**

- Il diritto esclusivo, garantito legalmente, di pubblicare, riprodurre, vendere una qualsivoglia forma di lavoro artistico, letterario, musicale



Proprietà intellettuale

- **Segreto industriale**

- Consiste nella totalità delle informazioni non note (formule, processi, programmi, metodologie, tecniche) in grado di fornire un valore aggiunto all'azienda

- **Marchio registrato**

- Consiste in un qualunque simbolo distintivo rappresentativo dei prodotti dell'azienda

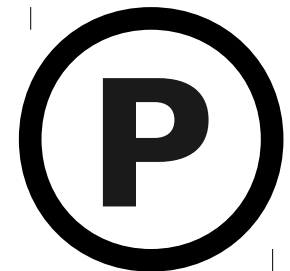


Proprietà intellettuale nel software

- **Il software non può essere protetto con dei marchi registrati**
- **Il software può essere protetto con dei brevetti**
 - I brevetti si applicano al contenuto informativo tecnologico
 - Sono più difficili da applicare alle “righe di codice”
- **Il software è solitamente protetto da **segreti industriali** e **copyright****

Copyright

- **Il copyright fornisce all'autore di un'opera cinque diritti esclusivi:**
 - Riproduzione
 - Modifica
 - Distribuzione
 - Rappresentazione in pubblico
 - Mostra in pubblico
- **Per i contenuti multimediali è comunemente usato anche un simbolo leggermente diverso**



- **Limitazioni del copyright:**

- **Idea:** l'idea espressa dal lavoro creativo non è protetta. Ad esempio, copiare un libro di ricette è un reato, riprodurre le ricette stesse no.
- **Fatti:** i fatti esposti nel lavoro creativo non sono protetti.
- **Creazione indipendente:** se un duplicato esatto del lavoro è creato in modalità del tutto indipendente, non viene violato il copyright.
- **Uso onesto:** l'uso non a fini di lucro è permesso anche a scapito di copia/riproduzione.

Segreto industriale

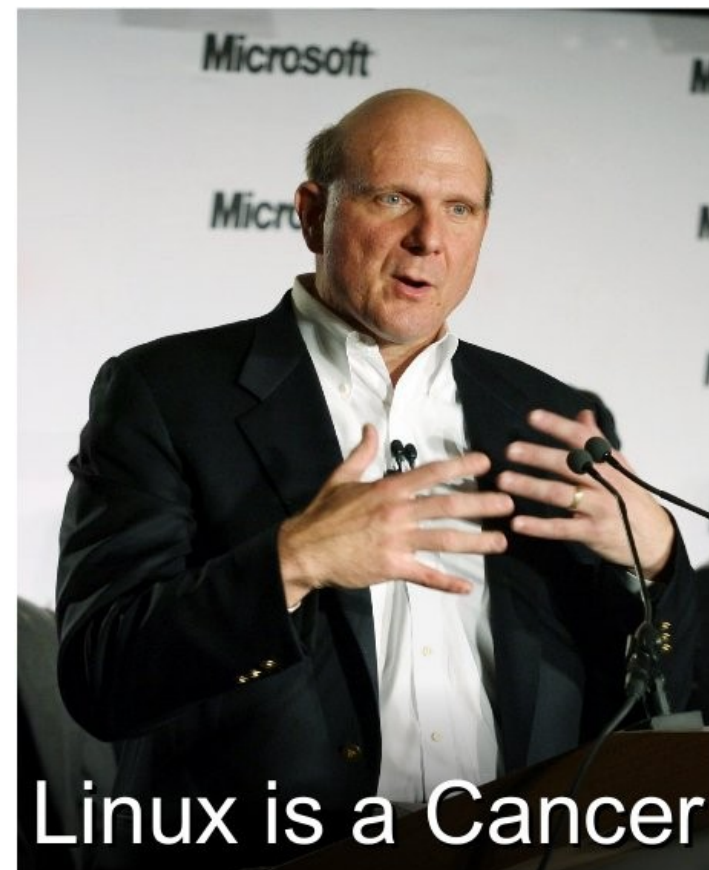
- **Il patrimonio creativo viene mantenuto segreto**
 - Con tutti i mezzi possibili
- **Violare il segreto costituisce un reato**
- **Non Disclosure Agreement:**
 - Accordo stipulato tra una ditta ed una terza parte che ha bisogno di conoscere il segreto industriale
 - La terza parte si impegna a non divulgare i segreti

Free software

- Termine coniato da Richard Stallman
- Software che garantisce all'utente i seguenti diritti:
 - **Esecuzione** del programma
 - **Studio** del funzionamento del programma
 - **Ridistribuzione** delle copia del programma
 - **Modifica** del programma, con vincolo di rilascio delle modifiche al pubblico

Free software

- Termine coniato da Richard Stallman
- Software che garantisce all'utente i seguenti diritti:
 - **Esecuzione** del programma
 - **Studio** del funzionamento del programma
 - **Ridistribuzione** delle copia del programma
 - **Modifica** del programma, con vincolo di rilascio delle modifiche al pubblico
 - **clausola virale**



Gratuità del free software

- **Il fatto che il software free sia gratuito non è in alcun modo garantito**
- **In inglese c'è una ambiguità terminologica**
 - Free=Libero
 - Free=Gratuito
- **Stallman chiarisce il concetto con il motto**
Free as in freedom, not as in free beer



- **Estensione della definizione di Free Software**
- **Open Source Initiative, 1998**
 - Ridistribuzione libera del software
 - Distribuzione del codice sorgente del software
 - Possibilità di creare lavoro derivato dal software
 - Capacità di distinguere il software originale dalle modifiche apportate
 - Nessuna discriminazione verso persone/gruppi
 - Nessuna discriminazione verso campi di ricerca
 - Distribuzione di una licenza di uso
 - Non specifica per il prodotto
 - Non influente su altri sw

Regole dello sviluppo Open Source

- **Il codice deve essere lasciato accessibile a tutti**
 - Codice e standard dati visibili e discutibili (aperti)
 - Processo di produzione visibile
 - Modifica e redistribuzione libera del codice
- **Release early, release often**
 - Rilasciare una versione funzionante del software appena possibile
 - Evitare le lunghe “wish list” ed i progetti completi
- **Premiare i contributi**
 - Solitamente, con delle “menzioni d'onore”

Razionale

- **Un ciclo di release più corto favorisce il feedback da parte degli utenti**
 - Refactoring, extreme programming, sviluppo incrementale
- **Sviluppo del software su una grande “base” di programmatori bravi, appassionati e volontari**
 - Riconoscimento morale (reputazione), non materiale
- **Automazione spinta di ogni fase del progetto**
 - Patch, version control, build, bug tracking

Licenze del software

- La **licenza** è un accordo contrattuale fra programmatore ed utente
- La licenza specifica le restrizioni di uso su:
 - Codice sorgente del software
 - Modifica del software
 - Redistribuzione del software
 - Ossia il **copyright** (diritto di copia)
- Ogni licenza ha un **disclaimer of warranty**:
 - Dichiarazione di non responsabilità verso i danni causati dal software
- **<http://www.opensource.org/licenses/>**

Licenza MIT

- **Garantisce qualunque diritto di copia, uso, modifica e redistribuzione del software**, a patto di includere la licenza MIT nella nuova distribuzione
- **Si perde ogni possibilità di denunciare gli sviluppatori** (clausola “the software is provided *as is*”)

Licenza BSD

- **Garantisce qualunque diritto di copia, uso, modifica e redistribuzione del software, a patto di:**
 - Includere la licenza BSD nella nuova distribuzione
 - Includere puntatori alla licenza nella documentazione (requirement soppresso nel 1989 a seguito di problemi legali)
- **Si perde ogni possibilità di denunciare gli sviluppatori**

Licenza Artistic

- **Garantisce qualunque diritto di copia, uso e modifica locale del software**
 - Permette di ridistribuire i binari modificati
 - Vieta la distribuzione del sorgente modificato
- **Perl**
 - “prominent notice” di modifica dei sorgenti
 - Ridistribuzione del codice richiede il rilascio “free”



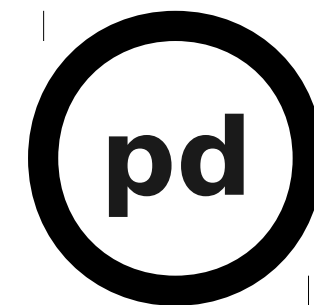
Licenza GPL v2

- **Garantisce qualunque diritto di copia, uso e modifica locale del software**
 - Permette di ridistribuire i binari modificati
 - Permette la distribuzione del sorgente modificato previa notifica testuale
- **Un software contenente una porzione rilasciata GPL, deve essere rilasciato a sua volta GPL**
- **Concetto di “CopyLeft” opposto a “CopyRight”**
 - Libertà di ridistribuire il contenuto a patto di rispettare la licenza



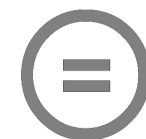
Creative Commons

- **Gruppo di licenze intermedio tra pubblico dominio e copyright**
- **Nasce dall'idea di giuristi di Stanford**
- **Usato tipicamente per la diffusione di contenuti**
- **L'autore può esprimere quali diritti concedere agli utenti**
 - **Copyright**: tutti i diritti sono riservati
 - **Public domain**: nessun diritto riservato
 - **Creative Commons**: alcuni diritti riservati



Licenze Creative Commons

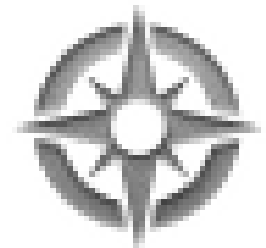
- **Distribuzione:** l'opera può essere copiata e distribuita (**implicito**)
- **No opere derivate:** l'opera non può essere modificata
- **Modifica:** l'opera può essere modificata e usata
- **Attribuzione:** bisogna indicare l'autore originale dell'opera
- **Non commerciale:** non può essere usata per usi commerciali
- **Share-alike:** in caso di modifica l'opera deve essere distribuita secondo le stesse regole



Esempio:



Identikit dello sviluppatore Open Source



Alcuni luoghi comuni

- **L'Open Source è per Nerd**
 - Sviluppato da Nerd
 - Usato da Nerd
 - Impossibile da utilizzare senza una conoscenza molto profonda dell'informatica
- **L'Open Source è sviluppato da hobbyisti**
 - Non c'è garanzia di qualità nel codice prodotto
 - Non c'è continuità nello sviluppo



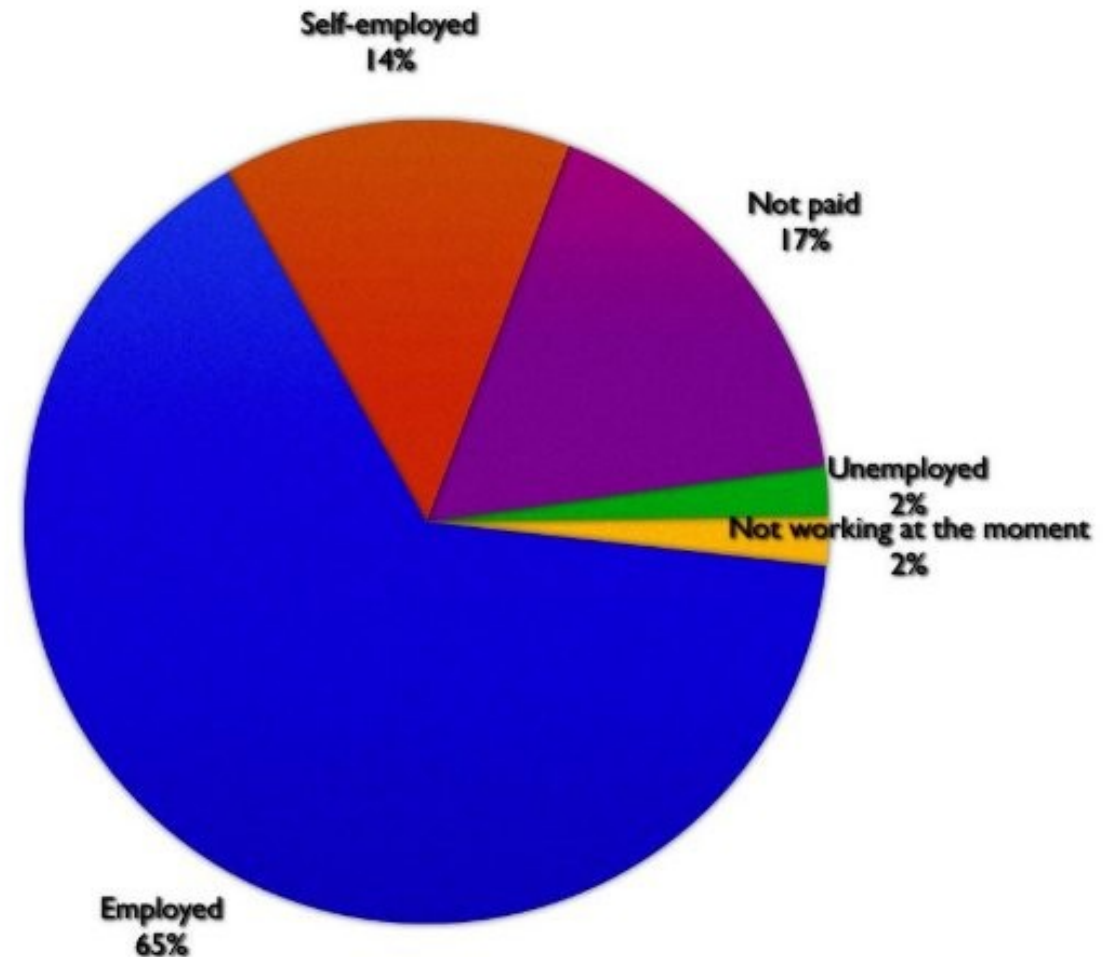
Impiego degli sviluppatori Open Source

- **Alcuni liberi professionisti**

- **Alcuni volontari**

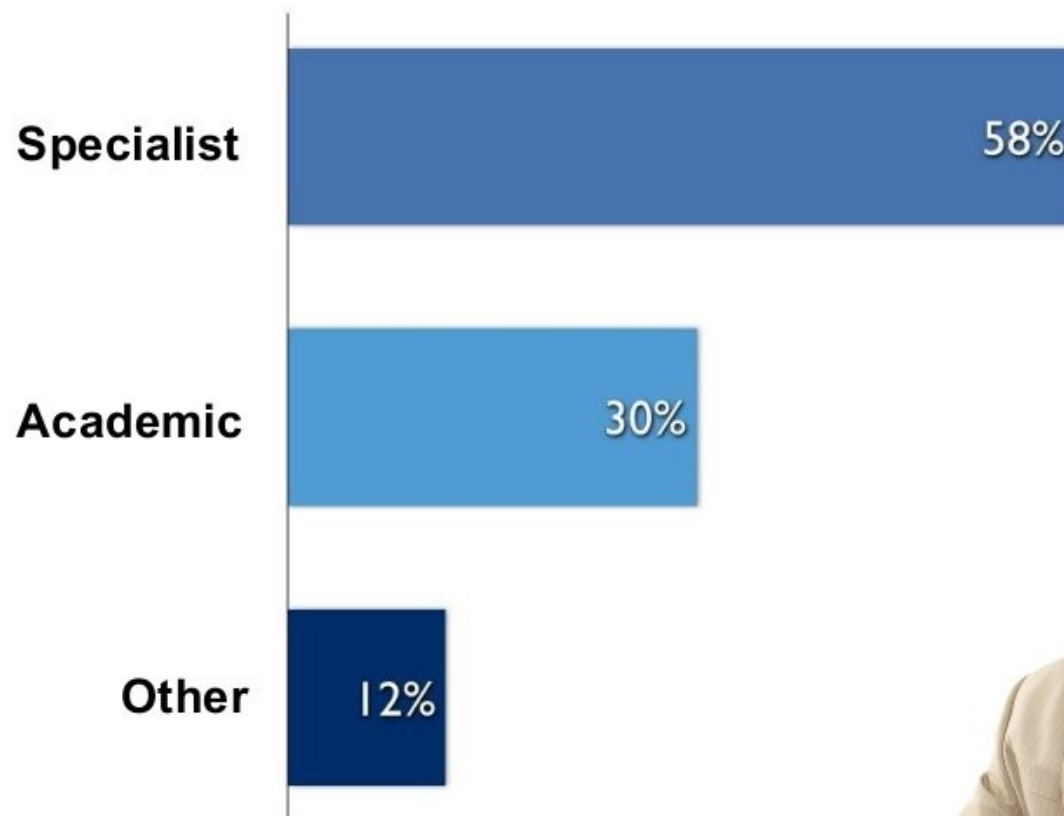
- **Alcuni disoccupati**

- **La maggior parte sono professionisti che lavorano nel settore**



Background degli sviluppatori

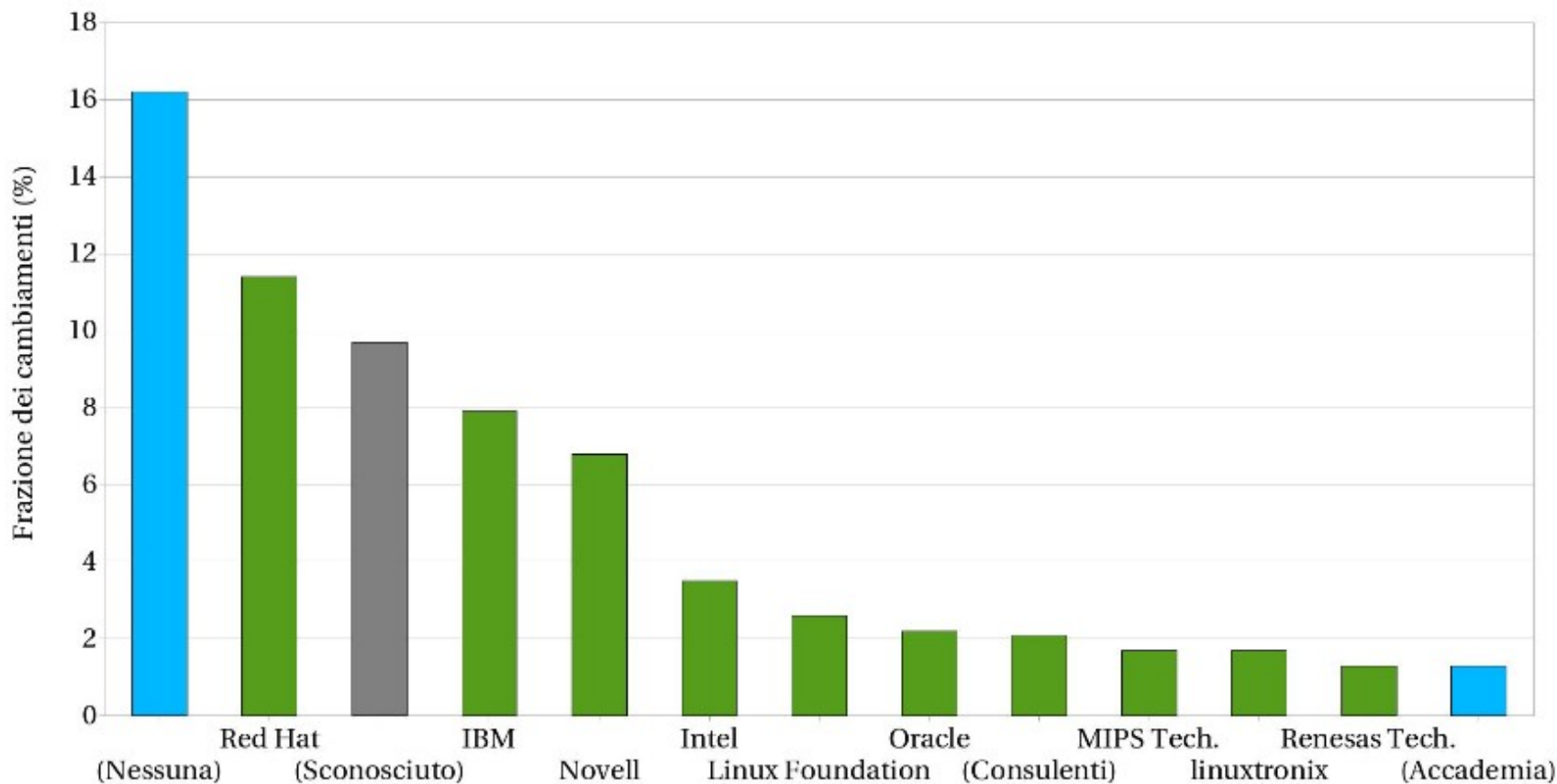
- L'università si conferma culla dell'Open Source
- La maggioranza degli sviluppatori sono specialisti



Contributi al kernel di Linux

- **Anche il kernel di Linux conferma questo trend**
 - Molte grandi aziende
 - Ma nessun contributor dominante

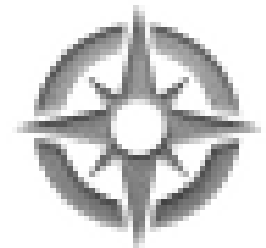
(Fonte: LWN.org)



Una descrizione dall'interno

- **Molti sviluppatori Open Source sono professionisti**
- **Ma con un elemento di passione**
 - “[. . .] it is strange: people who I believe are some of the best programmers in the world choose to work on free software”
[Andrew Morton, 2005]
 - “Even though we’re all paid to do kernel work, the culture is largely unchanged from the earliest days of Linux. We work on it because we love our craft and because it allows us to contribute to our society in the way in which we are most able. Corny but true.”
[Andrew Morton, 2005]

Modelli di business



La crisi del software

- **Pervasività del software e dell'informatica nella vita quotidiana**
 - Molti servizi sono offerti utilizzando infrastrutture software (banche, aziende produttive, amministrazioni pubbliche)
 - Enorme necessità di sviluppo continuo di nuovi software
- **Nonostante queste premesse, molte software house falliscono**

La crisi del software

- **I motivi della crisi**
 - Utenti insoddisfatti
 - Litigi e cause ritardano i pagamenti
- **La scontentezza degli utenti**
 - Il software costa troppo (comunque più di quanto sono disposti a pagare)
 - Il software viene consegnato in ritardo
 - La qualità non è adeguata (politiche di dumping dei prezzi a scapito della qualità per avere contratti)
 - Una volta trovato un fornitore è un contratto di fatti impossibile da rescindere
- **L'Open Source può agire su questi problemi**

Modelli di business Open Source

- **Il software come prodotto non procura fatturato**
- **La revenue si concentra su servizi a valore aggiunto**
 - Modifiche e personalizzazioni
 - Formazione
 - Assistenza e manutenzione
- **La scelta Open Source consente di rivitalizzare progetti software altrimenti fallimentari**
- **Esistono aziende molto importanti che investono pesantemente sull'Open Source**

Alcuni nomi...

Microsoft[®]



redhat.



Sun[®]
microsystems

Novell[®]

DELL[™]

ORACLE[®]

IBM

NOKIA
Connecting People

Google[™]



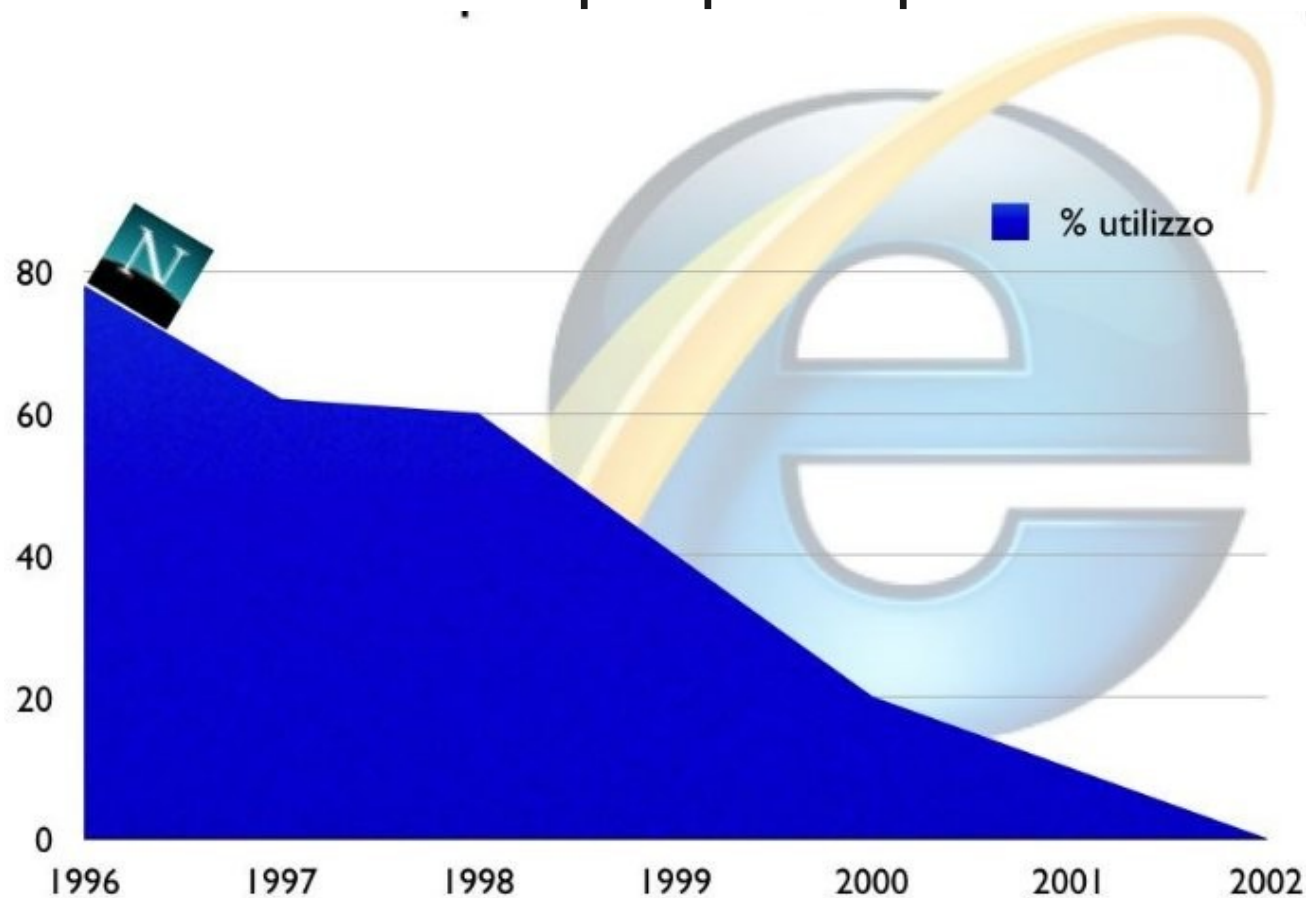
MySQL[®]

BBC

Mozilla Foundation

- **Il primo round della guerra dei browser:**

- Netscape vede un'erosione le proprie quote di mercato
- Declino inarrestabile
- *Nessuno ha mai vinto Microsoft giocando con le sue regole*
- **Spostarsi su un altro terreno**



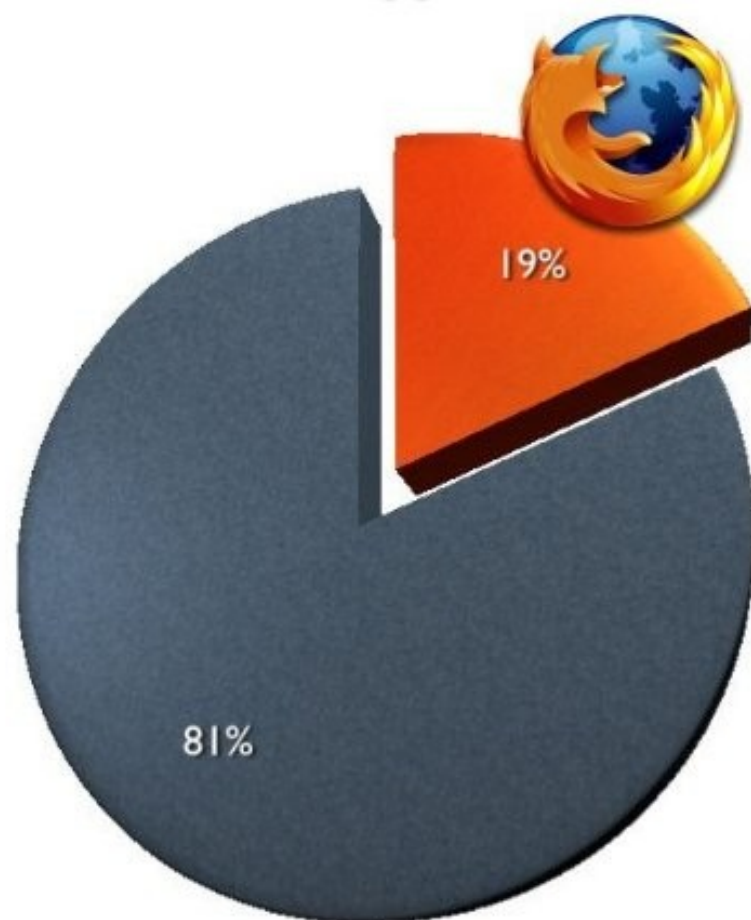
Mozilla foundation

- **Netscape rilascia il codice del proprio browser**
- **Chiunque può contribuire allo sviluppo**
 - Patch
 - Nuove funzioni
 - Supporto per estensioni che non sarebbero state realizzabili (Cryptozilla)



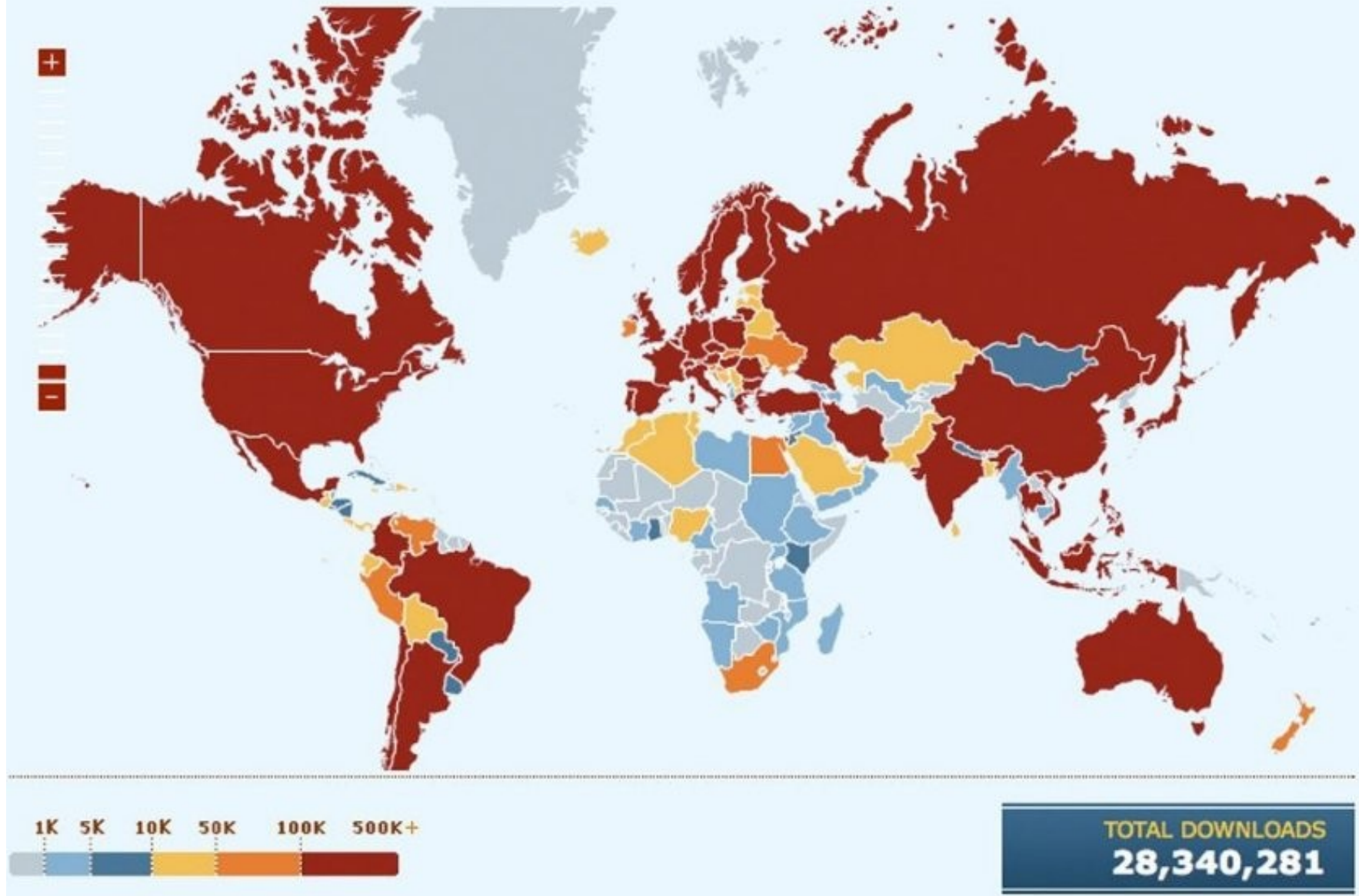
La guerra dei browser oggi

- **Mozilla ha costantemente guadagnato quote di mercato negli ultimi anni**
- **La suite Mozilla è molto diversificata**
 - Firefox
 - Thunderbird
 - Sunbird
- **La strada da fare è ancora lunga, ma lo sviluppo continua...**



Firefox 3 download day

Firefox 3 downloads worldwide as of July 2, 2008



- **IBM WebSphere**

- Piattaforma Web per servizi avanzati
- La base da cui partire per creare nuovi servizi è un Web server

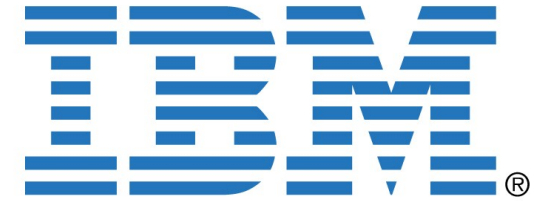
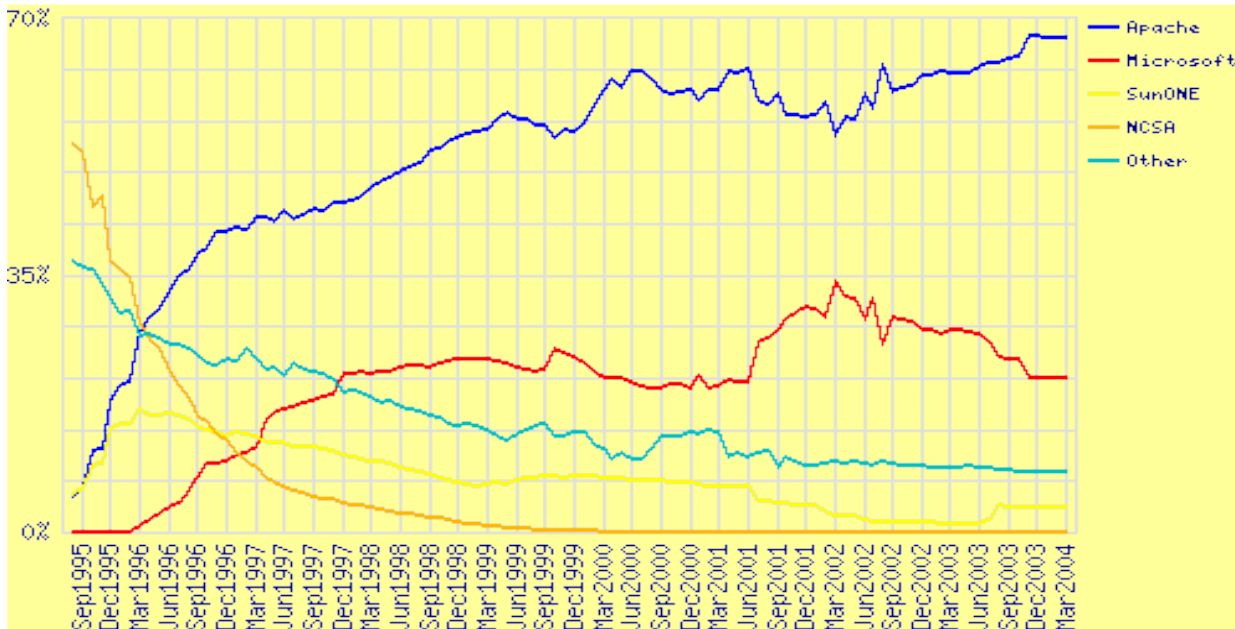


- **IBM non ha un proprio server**

- Costo iniziale per la piattaforma non sostenibile
- Costo di manutenzione della piattaforma antieconomico
- Difficile entrare in un mercato che ha già un chiaro vincitore

IBM WebSphere

- Il mercato dei Web Server ha già un leader: Apache

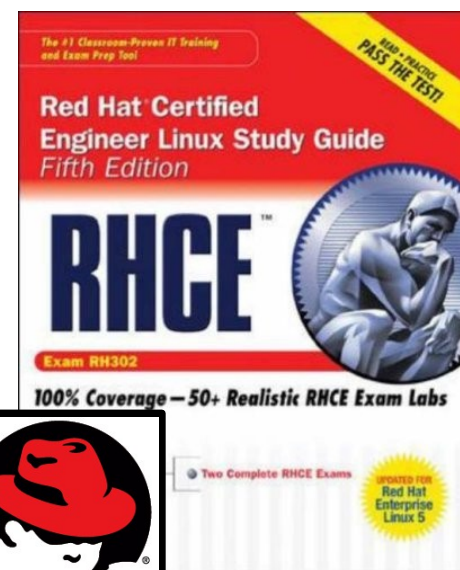


- IBM ha deciso di adottare questo software come parte di WebSphere
- WebSphere aggiunge servizi esterni ad un elemento Open Source



RedHat Enterprise

- **Molte distribuzioni sono a disposizione degli utenti**
- **Il valore aggiunto di RedHat**
 - Supporto tecnico
 - Certificazione di compatibilità con hardware e software
 - Formazione e certificazione dei tecnici
- **La distribuzione è gratuita**
 - Il software è analogo a quanto si trova in Fedora Core
 - **La revenue viene dal valore aggiunto**

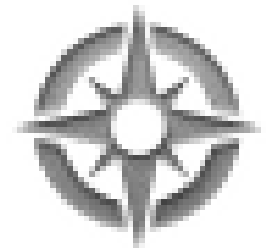


MayKing Open Suite

- **Software gestionale e di business intelligence**
- **Il valore di un gestionale:**
 - Personalizzazione alle esigenze del client
 - Configurazione strumenti per business intelligence
 - Il tempo per un deployment SAP è di circa 2 anni (per tempi di setup)
- **MayKing usa un pacchetto open source come base**
 - Viene offerto il valore aggiunto ovvero la personalizzazione e la formazione



Motivazioni per una scelta Open Source



Punto di vista di un utente

- **Vantaggi del software open source rispetto a software proprietario**

- Ragioni economiche
- Qualità del software
- Motivazioni etiche



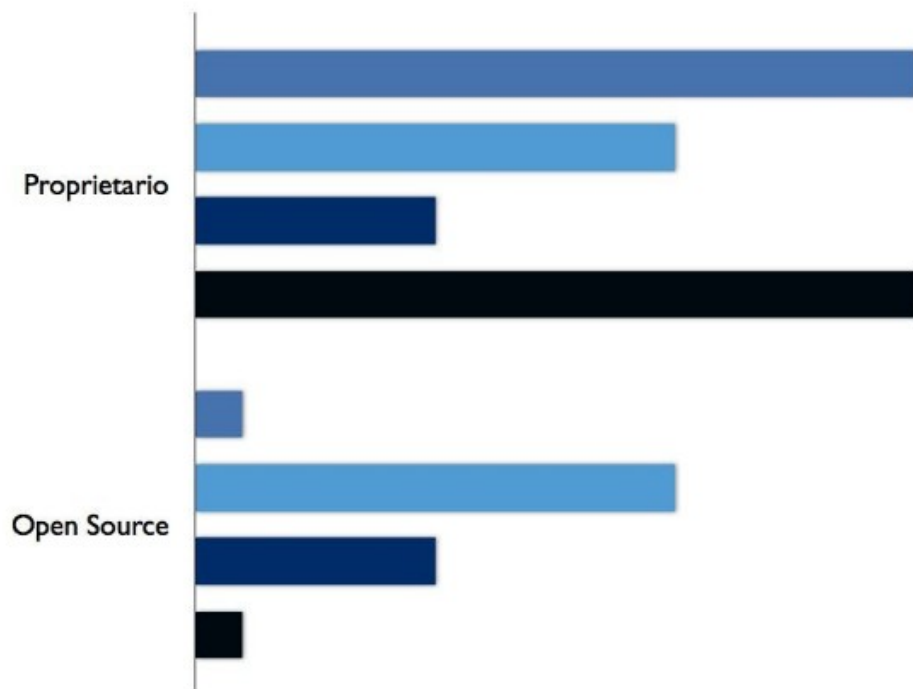
Ragioni economiche

- Il software Open Source tipicamente costa meno del software proprietario
 - A meno di comportamenti illegali



Software TCO
(Total Cost of Ownership)

■ Acquisizione ■ Training
■ Evoluzione ■ Core fixes

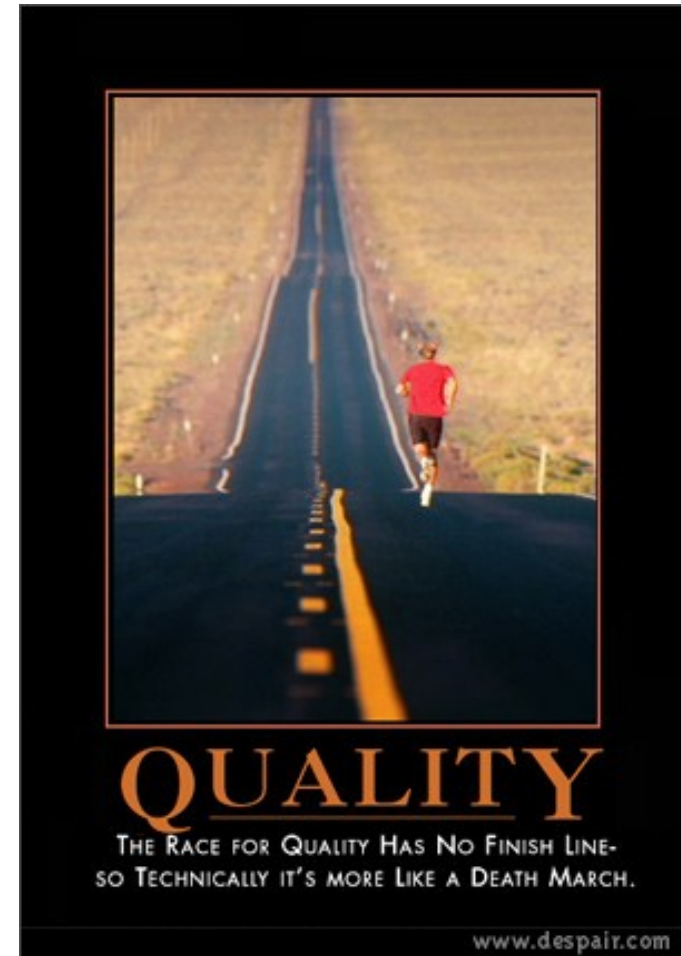


Spiegazione

- **Aree di risparmio**
 - **Acquisizione:** il software open source è tipicamente già disponibile in rete
 - **Core fixes:** lo sviluppo di nuove versioni e correzione di errori è indipendente dagli utenti
- **Costi analoghi al software proprietario**
 - **Evoluzione:** se richiedo una modifica per adattare il software alle mie esigenze è un costo a mio carico
 - **Training:** per trarre il massimo della produttività devo sempre investire in formazione

Qualità del software

- **Ogni utente ha accesso al codice sorgente:**
 - Auditing continuo della qualità del codice
 - Meccanismo di reputazione degli sviluppatori garantisce elevata professionalità
- **Studi indipendenti confermano che il codice Open Source ha qualità non inferiore al software proprietario**
- **Alcuni casi da hall of shame...**



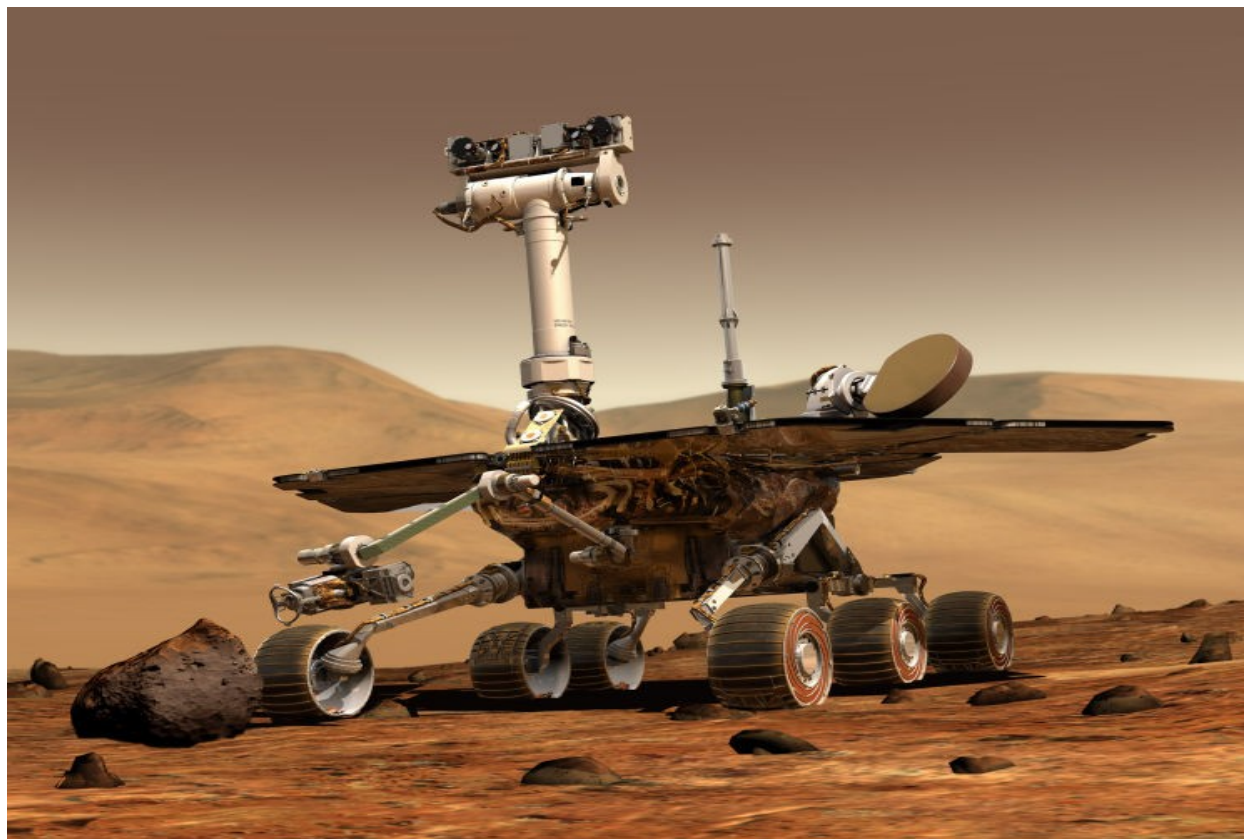
F16 “upside-down”

- **A causa di un errore nel software del pilota automatico, durante una simulazione un F16 si è ribaltato in seguito all'attraversamento dell'equatore**



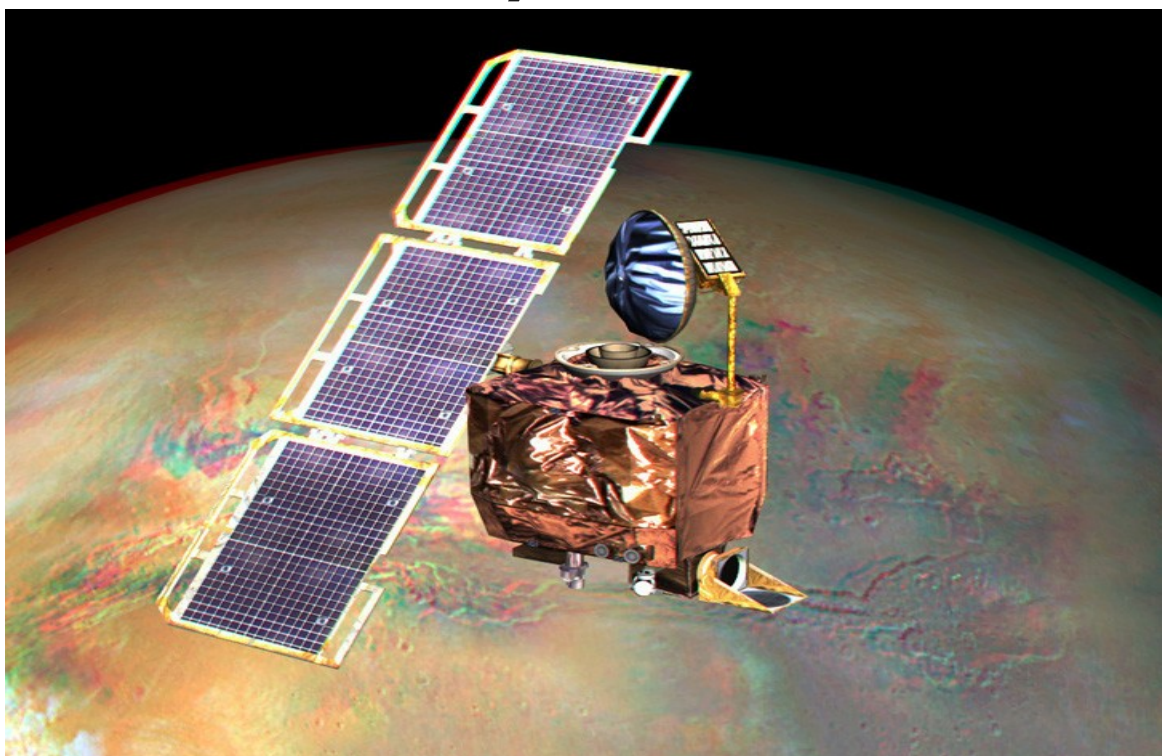
La sonda spaziale Mars Rover

- **Il software del NASA Mars Rover ha subito un blocco a causa di troppi file aperti contemporaneamente**



La sonda “Mars Climate Orbiter”

- La sonda Mars Climate Orbiter si è abbattuta sul suolo marziano a causa di un **“silly mistake”**: diversi gruppi di sviluppo utilizzavano **diverse unità di misura** (anglosassoni vs metriche)



Motivazioni etiche

- **Le spese per il software vengono investite per “il bene comune”**
 - Ogni contributo allo sviluppo del codice viene messo a disposizione della comunità
 - Le spese per lo sviluppo possono essere investite sulle comunità locali
 - Viene mantenuto il controllo su dati e processi (i formati sono per definizione aperti e supportati)



Motivazioni etiche

- **Contributi a disposizione della comunità**
- **Si crea un ciclo virtuoso:**
 - L'utente riceve beneficio da altri
 - L'utente procura beneficio agli altri
- **Piccoli esempi:**
 - Patch per supporto TCL in libgtsourceview
 - Supporto per modem UMTS PCMCIA di TIM
 - Supporto per contact photo in Evolution



Motivazioni etiche

- **Riversare le spese su comunità locali**
- **I soldi spesi in licenze non rimangono sul territorio**
 - **Il caso Microsoft:**
 - Microsoft Irlanda riceve l'80% dei soldi delle licenze vendute in Italia
 - I soldi vengono poi trasferiti negli USA
- **Le spese in software open source**
 - **Il caso LAPAM:**
 - Con i soldi risparmiati in licenze viene finanziata una software house
 - Si producono software gestionali e si offrono corsi su prodotti Open Source



Microsoft®



Motivazioni etiche

- **Marketing FUD (Fear Uncertainty and Doubt)**
 - Cosa succede se un progetto open source viene abbandonato? Chi garantisce assistenza?
 - Non c'è sicurezza nell'adozione di software Open
- **Alcuni controesempi**
 - Un progetto sufficientemente vitale difficilmente sparisce (es. Linux, MySQL)
 - Aziende non solide possono fallire (es. SCO)
 - Progetti software non redditizi possono essere dismessi (es. IBM OS/2)



Motivazioni etiche

- **Applichiamo al contrario le argomentazioni FUD**
 - Cosa succede se un'azienda dismette il supporto di un formato di file?
 - Cosa succede se un progetto software commerciale viene cancellato?
- **Il caso di Neal Stephenson:**
 - Alcuni dei suoi racconti sono irrecuperabili perché salvati in un vecchio formato Mac Word
 - Ora usa solo formati ASCII e TeX
- **L'uso di software open e di formati Open garantisce il pieno controllo dei propri dati!**



Credits

- **Questa presentazione è fatta usando byte riciclati**

- Simone Aliprandi
- Mauro Andreolini
- Marco Cesati
- Matteo Generali
- Francesco Nigro
- Antonio Russo
- ...

- **Qualche approfondimento:**

- “In the beginning was the command line”
N. Stephenson
- Free Software Foundation - www.fsf.org
- Creative Commons - www.creativecommons.org



Domande?





Il software e il movimento Open Source



Riccardo Lancellotti

Università di Modena e Reggio Emilia