



Dipartimento di  
Elettronica e Informazione

---

# Fare software nel 2008: l'Open Source e il ruolo delle imprese

---

*Eugenio Capra*  
eugenio.capra@polimi.it

IBM Softwareland, Monza, 18 settembre 2008

# Cos'è l'Open Source?

*Un modello di licenze in  
accordo con le definizioni  
OSI*

*Linux*

*Software con il codice  
sorgente disponibile*

*Software scritto su  
base volontaria nel  
tempo libero*

*Un nuovo modello di  
marketing e di business*

*Un modello cooperativo di  
sviluppo del software che si  
basa sulle comunità*

*Software a cui tutti possono  
contribuire*

# L'Open Source non è solo un particolare tipo di licenza, ma rappresenta diversi modelli

 Focus di questa presentazione

**Modello di licenza e distribuzione**

**Modello ideologico**

**Modello di sviluppo e di governance**

**Modello di business**

# Indice

- L'Open Source come modello di sviluppo
- Il ruolo delle imprese nei progetti Open Source
- “Open” al di là del codice

# Molti hanno in mente i progetti Open Source basati su comunità...

ESEMPIO

*jEdit*

## Aspetto

## Fatti

### Descrizione

- Editor di testi Java (850k download all'anno)
- Ospitato su SourceForge

### Sviluppatori

- 150 sviluppatori, tutti volontari

### Stile di gestione

- Chiunque può contribuire
- Il diritto di *commit* è accordato in base al merito
- Le decisioni sono prese discutendo informalmente i problemi sui forum e nelle mailing list – *lazy consensus*
- Non ci sono scadenze e ciascuno prende volontariamente in carico i problemi non assegnati che vuole affrontare

### Comunicazione

- Mailing list, forum, canali IRC
- Gli sviluppatori non si incontrano mai di persona

# ...ma alcuni progetti Open Source di fatto sono aziende...

ESEMPIO

## *SugarCRM*

### Aspetto

### Fatti

#### Descrizione

- Piattaforma di CRM (1M di utenti)
- Copia ombra su SourceForge, ma SugarCRM è anche un'azienda

#### Sviluppatori

- La maggior parte del codice è scritto da 11 impiegati di SugarCRM

#### Stile di gestione

- I processi di sviluppo sono molto simili a quelli delle imprese tradizionali
- Le roadmap e le scadenze sono definite dal management
- Sui forum si stimolano le discussioni e si chiede l'opinione della comunità tramite sondaggi online, ma le decisioni finali sono sempre prese dal management

#### Comunicazione

- La maggior parte degli sviluppatori lavorano nello stesso luogo e si incontrano regolarmente

# ...e alcune comunità sono influenzate indirettamente da aziende IT

ESEMPIO

## *Eclipse*

### Aspetto

### Fatti

#### Descrizione

- Piattaforma di sviluppo e supporto run-time
- Comunità a sè stante con più di 195 progetti attivi

#### Sviluppatori

- Il 75% dei 578 committer sono pagati (66 aziende coinvolte). Il 30% sono dipendenti IBM, di cui 20 per la piattaforma core

#### Stile di gestione

- Il modello di gestione è strutturato e basato su organi di governo gerarchici
- La Fondazione Eclipse (più di 65 aziende) sostiene e guida il progetto su base democratica
- Ogni azienda ha la propria agenda, che sostiene indirettamente tramite i propri committer – “do-ocracy”

#### Comunicazione

- Mailing list, bugzilla e phone conference
- I dipendenti della stessa azienda si possono incontrare, certe volte anche settimanalmente

# Un possibile modello di partecipazione di impresa a progetti Open Source: IBM

ESEMPIO

## Approccio

- IBM partecipa a circa 150 progetti OS con circa 1000 sviluppatori (600 sul kernel di Linux)
- Gli obiettivi principali sono creare open standard e promuovere progetti SW ritenuti utili

- La governance del progetto è lasciata alla comunità, gli obiettivi aziendali sono portati avanti tramite “do-ocracy”
- La partecipazione di dipendenti a progetti OS viene approvata internamente dopo che sono stati valutati tutti gli aspetti legali, tecnici, strategici e di business
- Gli sviluppatori interni devono bilanciare il loro ruolo nella comunità e nell’azienda: sono liberi per le questioni tecniche, ma vi è un coordinamento interno per le questioni strategiche o legate al business
- Si fa training specifico Open Source per insegnare ai dipendenti a mettersi in gioco
- Si rispettano le regole e i processi della comunità

# Esiste un continuum fra *open* e *closed*: verso un modello per misurare la *openness*

## Caratteristiche

---

### Obiettivi

---

- Identificare le dimensioni che caratterizzano i diversi approcci di governance
- Valutare la *openness* di un progetto SW
- Il modello è basato sullo studio continuo di 75 progetti Open Source significativi, tramite interviste faccia a faccia con project manager e community manager
- Definisce criteri quantitativi e scale di valutazione per alcune dimensioni chiave
- Non valuta la “bontà” di un progetto

# Abbiamo analizzato 75 importanti progetti Open Source

## Variabile

## Valore

**Criteri**

- Più famosi e diffusi
- Sia commerciali che comunità (SF, Apache, Tigris, Java)

**Interviste in presenza**

- 26

**Sviluppatori**

- 80 in media

**Dimensione**

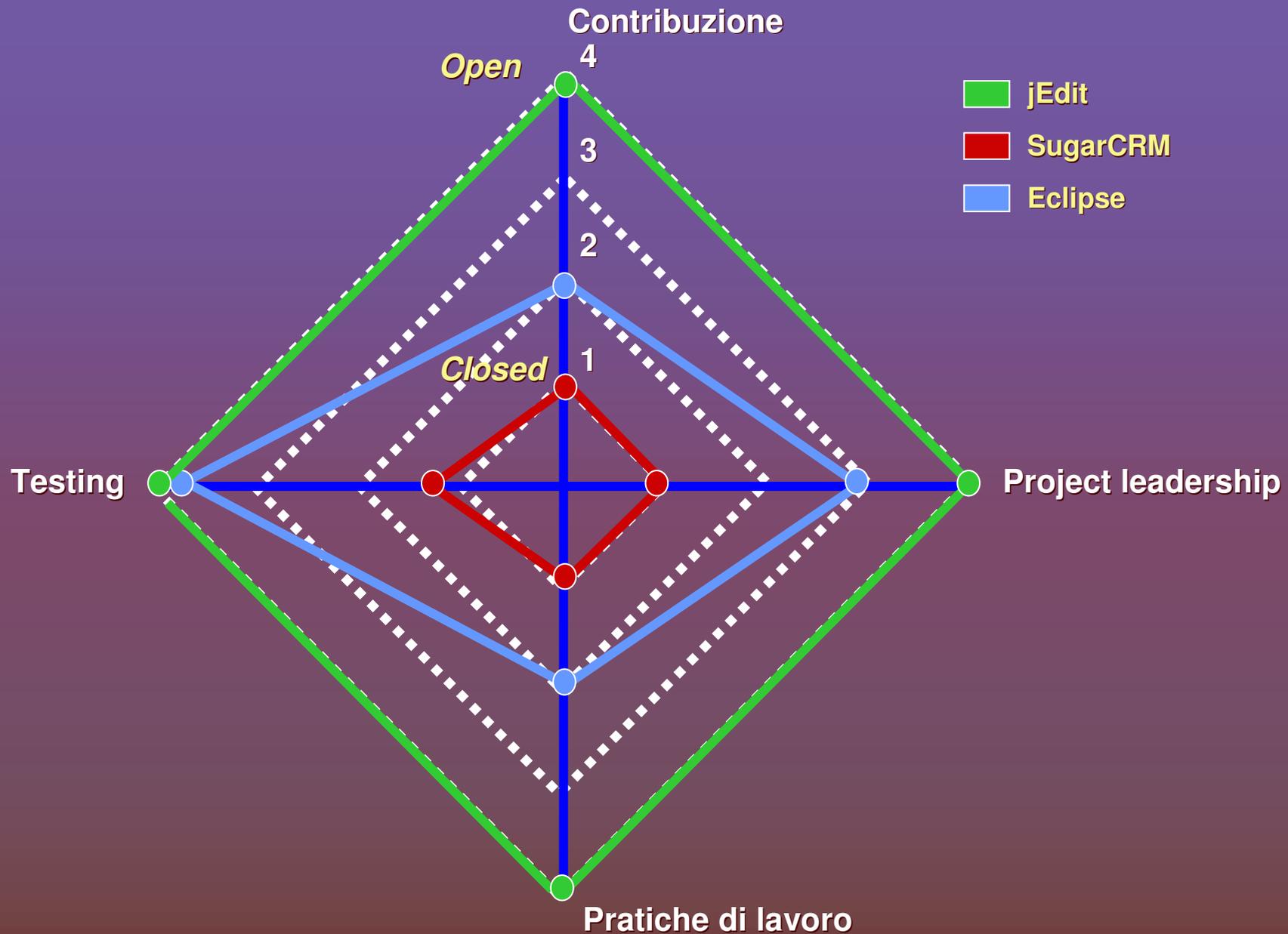
- 900 KLOC in media

## Esempi di progetti

- MySql
- SugarCRM
- OpenOffice
- Eclipse
- Mozilla
- JavaDB
- PostgreSQL
- OpenSolaris
- jEdit
- ...

# Il modello di *openness* applicato

ESEMPIO



# Il modello di *openness* applicato: Eclipse

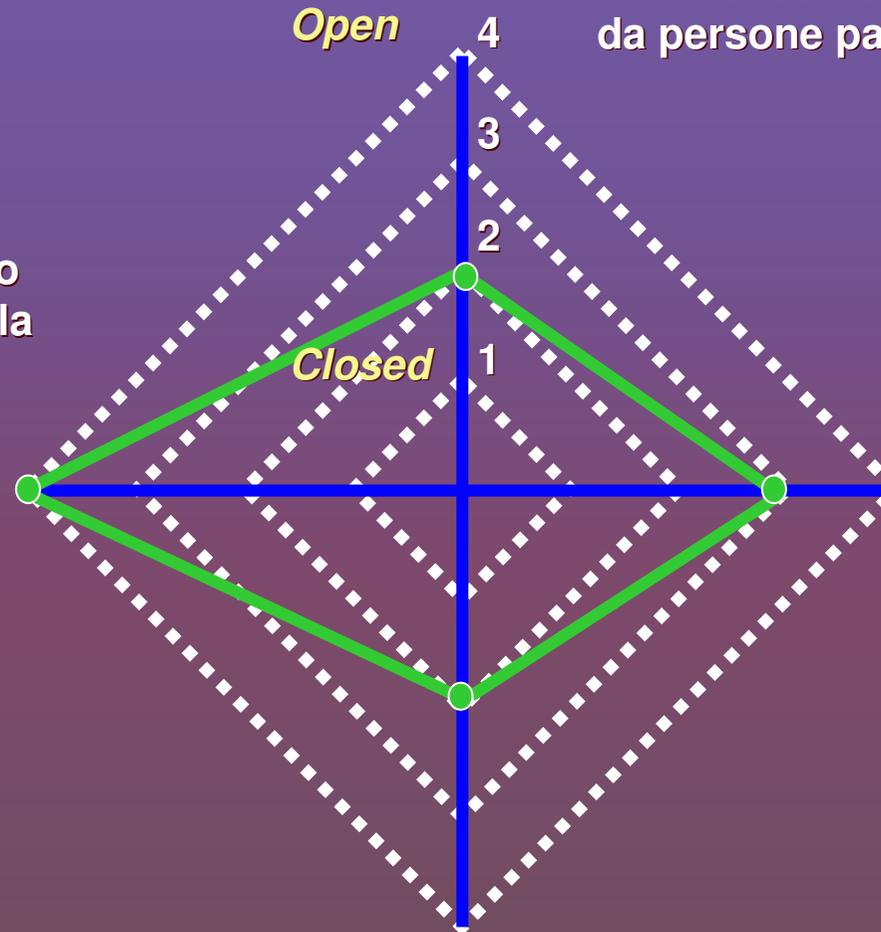
ESEMPIO

## Contribuzione:

- 75% del codice è scritto da persone pagate

## Testing:

- Il testing è lasciato completamente alla comunità: ogni committer è responsabile dei suoi contributi



## Project Leadership:

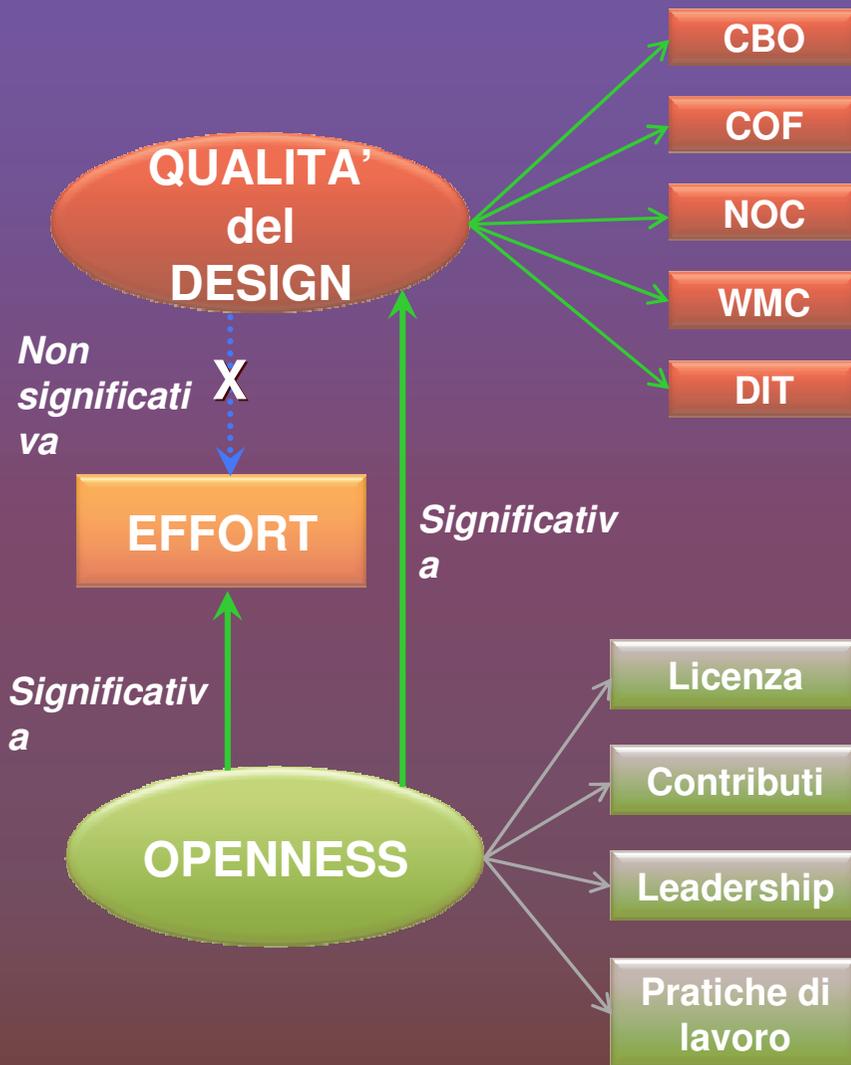
- Le decisioni vengono prese dalla comunità, ma le imprese influenzano indirettamente la governance del progetto tramite i committer

## Pratiche di lavoro:

- Largo utilizzo di mailing list e phone conference
- Gli sviluppatori che lavorano per la stessa azienda però spesso si incontrano di persona

# La relazione fra *openness*, qualità ed effort di sviluppo

## Modello statistico di regressione



## Risultati

- Gli approcci più *open* implicano una maggiore qualità del design in quanto questa abilita lo sviluppo distribuito e cooperativo
- Fare software *open* richiede uno sforzo aggiuntivo di coordinamento, quindi "costa" di più
- L'Open Source non va visto solo come uno strumento di efficienza, ma soprattutto di efficacia

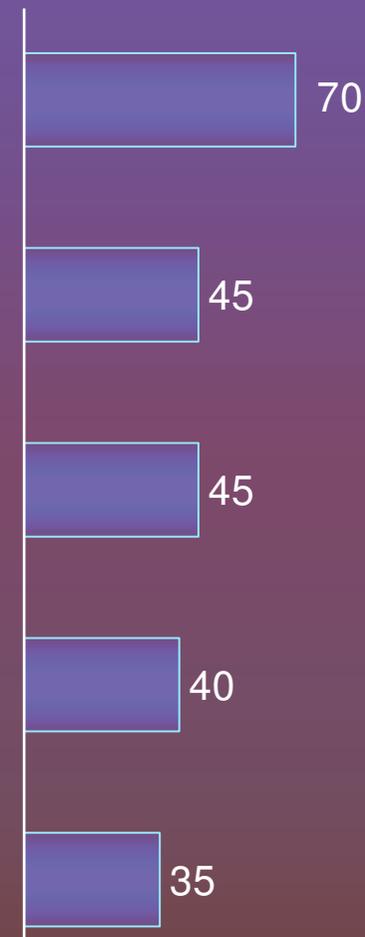
# Quali sono le best practice *open* che portano al successo di un progetto software?

*Sondaggio fra i project manager del campione*

## Best practice

- Interazione continua ed estensiva con la comunità
- Opportunità per tante persone di contribuire
- Utilizzo estensivo di altri componenti OS
- Controllo scrupoloso del diritto di commit e processo di revisione del codice
- Opportunità di far testare versioni preliminari a molti utenti

## Rilevanza %



# Indice

- L'Open Source come modello di sviluppo
- Il ruolo delle imprese nei progetti Open Source
- “Open” al di là del codice

# Un risultato provocatorio

L'Open Source non è solo il mondo di "smanettoni" che scrivono codice di notte e nei week end per pura soddisfazione personale

Molte imprese adottano l'Open Source come strategia aziendale

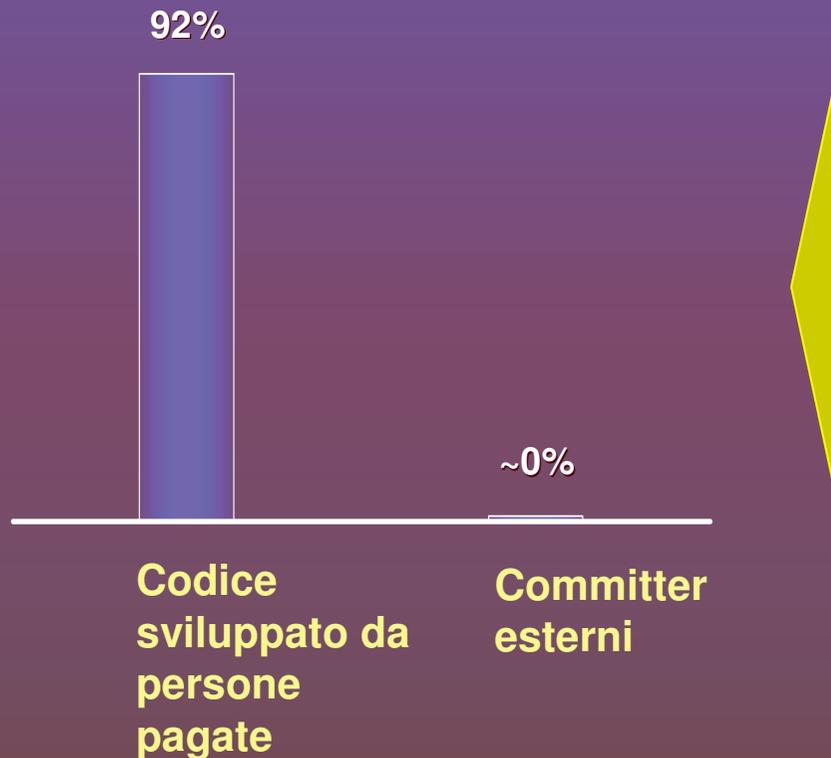
Alcune comunità Open Source sono indirettamente sostenute e guidate da imprese

# Le imprese sono imprese

## Progetti Open Source commerciali



### Statistiche dal nostro campione



### Perché un'impresa adotta strategie open?

- Fa bene all'immagine dell'azienda
- La comunità è una piattaforma di marketing
- Per creare nuovi modelli di business
- Per avere subito feedback dagli utenti, incluso il testing
- In ogni caso, le imprese mantengono il controllo del codice e garantiscono la qualità tramite QA e testing interni

# Abbiamo approfondito lo studio anche per i progetti basati su comunità con un nuovo sondaggio



<u>Variabile</u>	<u>Valore</u>
Progetti	<ul style="list-style-type: none"><li>• 1.411</li></ul>
Sviluppatori	<ul style="list-style-type: none"><li>• 4,5 + 2,2 amministratori in media</li></ul>
Età	<ul style="list-style-type: none"><li>• 40 mesi in media</li></ul>
Versioni rilasciate	<ul style="list-style-type: none"><li>• 8 in media</li></ul>

## Criteria di selezione

- SourceForge
- Scritti in Java
- Status  $\geq$  Beta
- Attivi

# Possibili approcci di partecipazione di imprese a progetti Open Source



## Approccio

## Attività svolte per il progetto

**Sviluppo**

- Scrittura codice
- Bug fixing
- Packaging

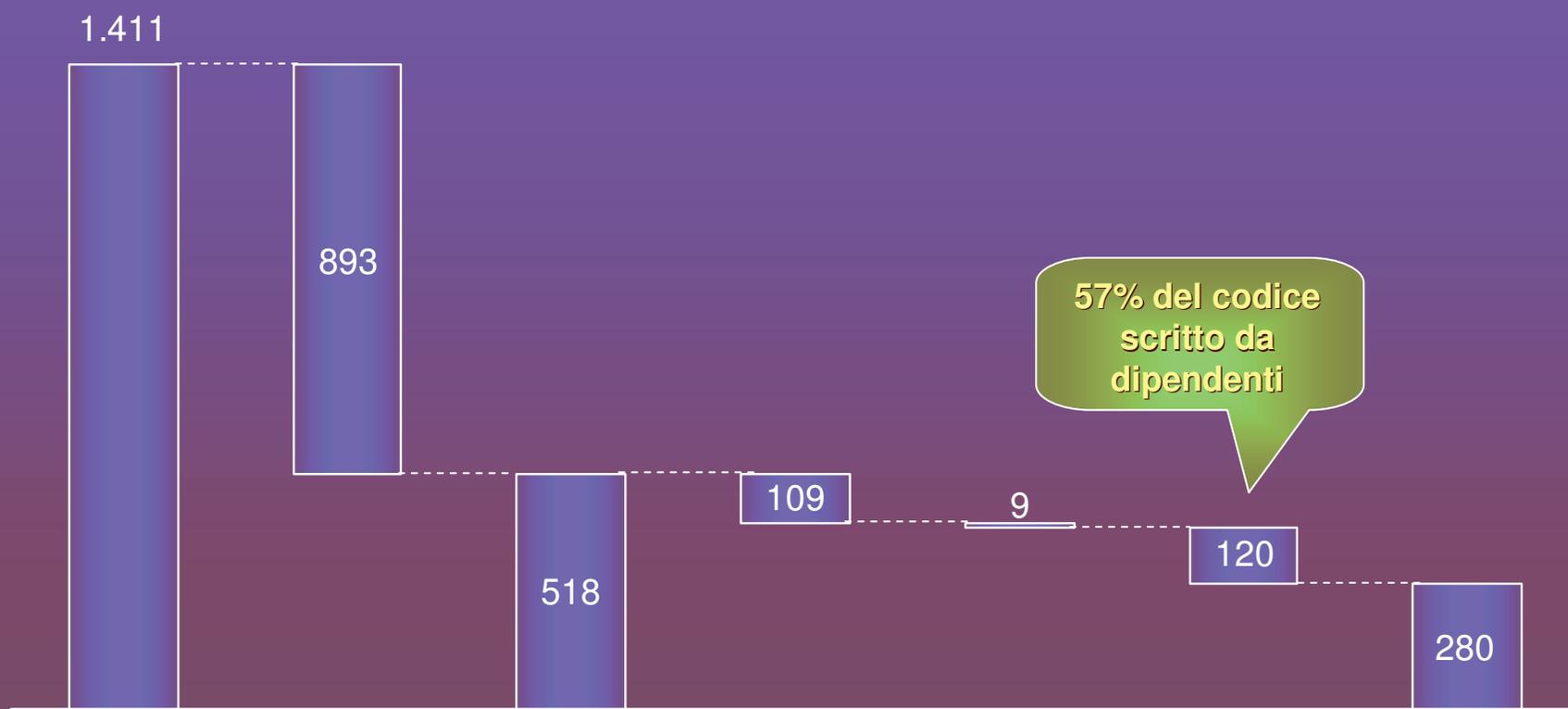
**Supporto**

- Animazione e gestione forum
- Supporto utenti
- Documentazione
- Supporto finanziario e infrastrutturale
- Distribuzione e marketing

**Management**

- Coordinamento
- Definizione requisiti
- Pianificazione

# La partecipazione delle imprese ai progetti Open Source basati su comunità



Percentuale su totale

100	63	37	8	1	9	19
-----	----	----	---	---	---	----

Percentuale su progetti partecipati

100	21	2	23	54
-----	----	---	----	----

# Perché le imprese sostengono i progetti Open Source?

- Spesso accelera lo sviluppo di una nuova tecnologia attraendo più persone che possono contribuire al progetto
- I componenti *open* sono accettati più facilmente dalla comunità
- E' un buon meccanismo per cooperare con altre imprese e sviluppare open standard
- Permette di perseguire i propri obiettivi di business stando dietro le quinte
- ***Open non vuol dire gratis!***

# Indice

- L'Open Source come modello di sviluppo
- Il ruolo delle imprese nei progetti Open Source
- “Open” al di là del codice

# La *openness* va ben al di là del codice: l'open computing

## Cosa può essere *open*

- Codice
- Protocolli standard (es. TCP/IP, XML, ...)
- Formati standard (es. PDF)
- Architettura
- ...

**La openness  
crea valore per  
la società**

## Vantaggi

- Meno vincoli per gli utenti
- Più cooperazione fra imprese
- Sviluppo più rapido della tecnologia
- Laboratorio mondiale per la ricerca

# Come l'Open Source può essere utile al Green IT: la ricerca al Politecnico di Milano

ESEMPIO

■ Impatto dell'Open Source

## Fasi della ricerca

- Definire nuove metriche di qualità e complessità
- Implementare un analizzatore di codice per misurare le nuove metriche
- **Selezionare applicazioni open source confrontabili**
- **Analizzare il codice aperto per misurare le metriche green**
- Misurare i consumi effettivi a parità di input e di condizioni al contorno per diverse configurazioni hardware
- Confrontare i dati ottenuti

## Obiettivi della ricerca

- Misurare e confrontare l'efficienza energetica di algoritmi e applicazioni
- Fornire nuovi parametri di merito per la scelta di un'applicazione
- Verificare in che modo la qualità del software impatta sui consumi

# L'opinione di un venture capitalist della Silicon Valley

*Se la tua start-up fonda il proprio business solo sul fatto che il codice è chiuso è destinata a fallire. La maggior parte dei software di successo può essere replicata da un gruppo di bravi studenti in pochi mesi. Sono la tua idea di business, il tuo mercato di riferimento e il giusto timing che determineranno il tuo successo*

eBay

YouTube

PayPal

# Grazie per l'attenzione

**Eugenio Capra**

Dipartimento di Elettronica e Informazione

Politecnico di Milano

via Ponzio, 34/5

20133 Milano

Tel +39 02 2399 4014

Fax +39 02 700 502 112

*[eugenio.capra@polimi.it](mailto:eugenio.capra@polimi.it)*

