

# **APPENDICE A: NASCITA ED EVOLUZIONE STORICA DELL'OSS**

La nascita ufficiale del movimento del software libero o aperto risale ai primi anni '80, nonostante prima di questo periodo esistessero forme di libera circolazione di codice sorgente tra sviluppatori. Nel 1983 Richard Stallman del MIT ha fondato la Free Software Foundation (FSF) per promuovere lo schema di licenza GNU/General Public Licensing (GPL), con l'obiettivo di proteggere il software libero dall'appropriazione da parte delle imprese commerciali, garantendo l'accesso al codice sorgente e la possibilità per gli utilizzatori di farne copie, di modificarlo e di ridistribuirlo senza pagare né far pagare licenze d'uso. Ispirandosi a questo principio, nell'agosto 1991 Linus Torvalds iniziò un processo condiviso di sviluppo del software mettendo in rete una nuova versione free del kernel Unix e chiedendo contributi alla comunità per individuare e correggere eventuali errori. Dopo poche settimane era pronta la versione di Linux 0.01. Nel 1998 un gruppo di fautori del software libero, tra cui Todd Anderson (Foresight Institute), Eric Raymond (autore dell'ormai celebre libro "The Cathedral and the bazaar"), John Hall e Larry Augustin (Linux International), hanno creato la Open Source Initiative (OSI), che ha fissato dei criteri generali da applicare alla distribuzione di software aperto. Tali criteri sono contenuti nella Open Source Definition e nell'OSI Certification Mark. L'obiettivo della OSI era di rendere il software libero più visibile nella comunità di produttori commerciali, aumentandone quindi il mercato potenziale, al di là della ristretta cerchia di utilizzatori pionieri (Raymond, 2001). Uno dei cardini di questo nuovo approccio consiste nella rinuncia alla clausola propria della GPL ("the license must not insist that all other programs distributed on the same medium must be open-source software").

Nonostante le differenze accennate, per semplicità d'ora in poi ci riferiremo all'OSS per intendere indifferentemente software libero o software aperto.

Come abbiamo già anticipato, per OSS si intende software distribuito con il codice sorgente. Il software aperto può essere distribuito gratuitamente sotto condizioni di licenza che garantiscono agli utilizzatori il diritto di usare, copiare, modificare e distribuire il codice sorgente. Tuttavia l'OSS non deve necessariamente essere distribuito gratuitamente. I distributori possono, infatti, fissare un prezzo che copre i costi di distribuzione del codice sorgente originario (es. il kernel di Linux), eventuali

costi di sviluppo di software aggiuntivo (es. interfacce grafiche e driver per l'installazione di periferiche), i costi di assemblaggio del codice eseguibile e i servizi di supporto collegati. Ma nessun diritto di utilizzazione dell'opera può essere richiesto al cliente nell'ambito dell'OSS.

Come ha notato Eric Raymond, il processo di sviluppo dell'OSS ha le caratteristiche di un 'bazar' dove chiunque (sviluppatori, utilizzatori) può entrare in contatto con un progetto di sviluppo, scaricare il codice sorgente, apportare modifiche (ulteriore sviluppo di linee di codice, debugging, ecc.) e spedirle al project leader, senza ricevere un compenso monetario. Il compito di certificazione e distribuzione delle modifiche proposte è normalmente svolto da un limitato numero di *core developers* che affiancano i *project leaders*.

Quali sono le implicazioni di questa particolare organizzazione del processo di sviluppo per la qualità del software? Secondo i fautori dell'OSS questo processo di sviluppo permette maggiore controllo della qualità, stabilità e sicurezza del software. Oltre al numero potenzialmente elevato di sviluppatori che partecipano al processo di sviluppo, secondo i sostenitori dell'OSS un elemento che gioca a favore della qualità del software aperto consiste nelle forti motivazioni di sviluppatori e utilizzatori, che scelgono liberamente di contribuire all'ispezione e correzione di errori del software aperto secondo la logica del 'Bazar'. Nel modello tradizionale di sviluppo ("la cattedrale") invece la rigida divisione gerarchica del lavoro tra sviluppatori e il numero limitato di contributori ridurrebbe le possibilità di controllo e di correzione di errori e allungherebbe i tempi di sviluppo (Raymond, 2001). In realtà, non esiste una ben documentata evidenza empirica che dimostri la relazione tra qualità del software, tempi di sviluppo e modalità di sviluppo. Inoltre, anche nel campo del software proprietario, ormai da tempo si sono diffusi modelli di sviluppo che permettono una elevata flessibilità e creatività da parte degli ingegneri del software (es. il rapid prototyping, l'incremental e evolutionary development e il ciclo di vita a spirale) (Fuggetta, 2001). Inoltre, vi sono limiti organizzativi che frenano la crescita dimensionale dei progetti OSS e ne allungano i tempi di sviluppo (Cubranic e Booth, 1999).

Infine, sul piano della sicurezza, l'apertura del codice sorgente non è condizione sufficiente per garantire livelli più elevati di sicurezza. Per esempio, una recente indagine condotta da Netcraft conclude che 'essendo oltre il 50% i siti Internet potenzialmente vulnerabili, esistono le condizioni per un'epidemia di attacchi sia verso siti basati su Microsoft IIS che su siti "Apache" (<http://www.netcraft.com/survey/> 1 luglio 2002).