



Il software

1. Introduzione

Il termine software è costituito dall'unione di due parole:

- soft (morbido);
- ware (manufatto, componente, oggetto).

La parola che otteniamo indica un programma o un insieme di programmi in grado di funzionare su un elaboratore. Per essere ancora più chiari, la parola software rappresenta qualsiasi applicazione che, quotidianamente, utilizziamo sui nostri personal computer.

L'origine del termine software risale alla Seconda Guerra Mondiale. I tecnici dell'esercito inglese erano infatti impegnati nel decriptare i codici tedeschi di Enigma, dispositivo elettromeccanico per cifrare e decifrare messaggi, di cui già conoscevano la meccanica interna (l'hardware) grazie ai servizi segreti polacchi.

Da allora il software ha subito innumerevoli variazioni, fino a diventare una componente fondamentale per tutti gli utenti che si accingono all'utilizzo di un personal computer: da quelli meno esperti, che si trovano ad utilizzarlo per svolgere le azioni quotidiane, a quelli esperti che lavorano allo sviluppo di nuovi software.

Oramai siamo talmente abituati a utilizzare software in qualunque momento della giornata, che tendiamo a dimenticarci della straordinaria importanza che rivestono nelle nostre vite. Usiamo software per scrivere o per navigare sul web, per scattare e modificare fotografie, per inviare mail o messaggi su WhatsApp. Non solo: sempre più spesso i software regolano il funzionamento delle nostre automobili, dei termostati di frigoriferi, lavatrici, senza neanche farcene rendere conto.

L'elenco potrebbe andare avanti per molto, diventando lunghissimo. La maggior parte delle azioni quotidiane vengono eseguite attraverso un programma informatico. Pensate che uno dei guru della Silicon Valley, soprannome per la parte meridionale della San Francisco Bay Area, nella California settentrionale, negli Stati Uniti d'America, che funge da centro globale per l'alta tecnologia e l'innovazione, Marc Andreessen, ha scritto che *"i software stanno divorando il mondo"*. La sua è sicuramente un'affermazione forte, dietro la quale non si nasconde nessuna

critica, bensì la constatazione di come gli strumenti digitali che ogni giorno ci permettono di compiere le nostre attività, abbiano raggiunto un'eccezionale influenza su tutti noi.

Questa affermazione non vale solo per i comuni software che utilizziamo tutti i giorni, ma anche per software più elaborati e complessi che stanno alla base delle più grandi scoperte e innovazioni degli ultimi anni. Pensate al concetto di intelligenza artificiale. Quando ne parliamo non stiamo facendo altro che parlare di una serie di software in grado di svolgere azioni complesse e anche di apprendere da ciò che accade durante lo svolgimento, quasi a simulare l'intelligenza umana.

Esistono software che hanno applicazioni importantissime nell'industria. È il caso di Echelon, sistema sviluppato grazie alla partnership tra Eni e la statunitense Stone Ridge Technology. Tecnicamente, è un "simulatore avanzato di giacimento per l'ottimizzazione del monitoraggio, sviluppo e produzione di petrolio e gas". Usiamo termini più semplici. Echelon è un programma capace di simulare il comportamento dei giacimenti. Eni lo utilizza per interpretare informazioni dalle quali scaturiscono importanti decisioni aziendali. Questo per far comprendere quanto il concetto di software influenzi decisioni quotidiane, sia per un'utenza che lo utilizza per azioni semplici che per un'utenza che lo utilizza per decisioni importanti.

Ci si sofferma, sempre più sulla potenza dei computer e si presta meno attenzione all'operato dell'uomo, che permette di mettere a frutto la capacità di calcolo delle macchine. Riprendendo un celebre slogan, si potrebbe dire che "la potenza è nulla senza il controllo". Una frase semplice ma forte, che riflette la necessità di trovare un equilibrio.

2. Storia del software ed evoluzione del software

Se vogliamo trovare un inizio alla storia del software, possiamo affermare che tutto ebbe inizio con interminabili sequenze binarie composte dai simboli 0 e 1, ovvero la programmazione in codice binario, nata oltre un secolo prima dei primi prototipi di computer.

Dobbiamo infatti andare al 1725, anno in cui Basile Bouchon, un impiegato nell'industria tessile, ha un'intuizione straordinaria: dare a un telaio i comandi necessari per disegnare automaticamente sulla stoffa le fantasie da lui desiderate. Nella sua idea questa azione poteva essere svolta utilizzando una scheda perforata. Questa scheda avrebbe rappresentato proprio la sequenza di 0 e 1. I simboli sarebbero stati descritti da un'area bucata o non bucata, in pratica sempre due alternative. Chiaramente questa situazione sembra lontanissima da quello che viviamo oggi, ma racchiude in sé sempre lo stesso concetto: trovare un modo per impartire dei comandi ad un calcolatore per ricevere in risposta dei risultati.

Il primo mattone della programmazione è stato posto quindi quasi tre secoli fa.

Dobbiamo fare un salto avanti di un secolo per incontrare la persona che è stata definita "la prima programmatrice della storia": Ada Lovelace. Ada Lovelace nasce a Londra nel 1815 dal poeta Lord Byron e dalla matematica Anne Isabella Milbanke e mostra fin da giovanissima un'intelligenza matematica fuori dal comune. È un'adolescente quando avviene nella sua vita un incontro molto importante con il professore di Matematica dell'università di Cambridge Charles Babbage. Quest'ultimo stava progettando in quel periodo due prototipi di computer: la Macchina Differenziale, in grado di eseguire alcune equazioni in maniera automatica e senza errori e la Macchina Analitica, in grado di eseguire qualunque forma di calcolo.

I costi eccessivi impedirono la costruzione delle due macchine, ma Babbage portò avanti il progetto grazie a conferenze in cui spiegava la teoria alla base delle sue invenzioni.

Ada Lovelace, nel frattempo diventata contessa, venne incaricata di tradurre il manoscritto in inglese. L'anno successivo terminò il lavoro. Le sue note erano talmente abbondanti da triplicare

la lunghezza del testo originario. In questo modo la Lovelace, dopo aver mostrato a Babbage e al mondo la comprensione perfetta dei due prototipi, approfittò per scrivere quello che è universalmente riconosciuto come "primo programma informatico di tutti i tempi". Il compito di questo software era di calcolare i numeri di Bernoulli. Per la prima volta accadde che un dispositivo venisse utilizzato per svolgere un compito attraverso una serie finita di regole. Per la prima volta si progettò un vero e proprio programma a partire da un algoritmo.

Ada Lovelace può essere quindi considerata una visionaria. Riuscì infatti ad intuire che i numeri possono rappresentare molto più che semplici quantità, immaginando che, in un futuro a lei remoto, i computer avrebbero potuto comporre musica, disegni ed essere impiegati nelle più importanti materie scientifiche.

Ada Lovelace fu fonte di ispirazione per il matematico britannico Alan Turing, che durante la Seconda Guerra Mondiale cercò di decifrare "Enigma", la macchina con cui i tedeschi criptavano le loro comunicazioni. Ada Lovelace viene infatti citata da Turing nel suo testo fondamentale: "Computing Machinery and Intelligence". In questo testo Turing esamina la possibilità che, un giorno, le macchine possano pensare. Se ci pensate è la fase che stiamo vivendo oggi. La computazione dei software si spinge sempre più verso il concetto di intelligenza artificiale e quindi di pensiero simulato.

Ma torniamo alla storia. Siamo ormai giunti all'alba dei primi veri software. Le intuizioni di Turing sono di fondamentale importanza per un altro informatico tra i più importanti della storia: l'ungherese John von Neumann. Grazie alle intuizioni relative alla Macchina di Turing), von Neumann, nel 1945, crea infatti l'EDVAC (Electronic Discrete Variables Automatic Computer): la prima macchina digitale programmabile tramite software e basata su quella che venne battezzata "architettura di von Neumann".

È grazie a questi pionieri che vengono poste le basi nello sviluppo del software. Piano piano, le prime rivoluzionarie applicazioni dei software, iniziano a fare la loro comparsa nei laboratori di ricerca. Dal mondo della scienza ci spostiamo gradualmente verso le funzionalità intese per il

consumo di massa come ad esempio i giochi. Vediamo alcune tappe che possiamo schematizzare:

- 1961: nasce Spacewar, il primo videogioco della storia.
- 1965: viene creato al MIT un software che mette in collegamento vari computer e invia brevi messaggi di testo: è l'alba delle email.
- 1975: nasce Telenet, il pilastro di Internet.
- 1983: viene lanciato Microsoft Word, che ha mandato in soffitta le macchine da scrivere.
- 1992: il formato JPEG ha rivoluzionato per sempre il nostro rapporto con la fotografia.

Gli esempi potrebbero andare avanti all'infinito. A partire dalla seconda metà del 20° secolo, i software hanno iniziato a diffondersi e a cambiare sia la nostra quotidianità sia il mondo della ricerca avanzata. La combinazione tra programmi sempre più sofisticati e hardware sempre più potenti è all'origine di alcuni degli strumenti scientifici più importanti dei nostri tempi. Il software Spinnaker, ad esempio, utilizzato da un supercomputer, è stato il primo a riprodurre (parzialmente) il funzionamento del cervello umano in tempo reale. Altri software consentono oggi di produrre nuovi farmaci più rapidamente, di simulare il big bang e un domani, forse, anche di prevenire i terremoti. Nati dall'intuizione di un lavoratore dell'industria tessile, i software hanno rivoluzionato ogni settore immaginabile.

La storia dell'evoluzione del software va per forza legata all'evoluzione dell'hardware. Possiamo affermare che una minaccia alla velocità di elaborazione, oltre ai costi, proviene dal software stesso. Ciò che per un utente finale conta non è tanto la velocità di elaborazione del processore, quanto piuttosto la velocità effettiva di elaborazione del codice. Questa viene calcolata come il tempo che occorre al processore per eseguire un'operazione. Pensiamo ad operazioni comuni come scrivere un testo, modificare una foto e così via.

Uno studio molto interessante è stato condotto da Nathan Myhrvold, direttore dell'Advanced Technology Group della Microsoft. Myhrvold ha effettuato uno studio sui prodotti

Microsoft calcolando le linee di codifica per le successive release dello stesso software. Facciamo due esempi:

- Basic: 4.000 linee di codice nel 1975 a 500.000 nel 1995
- Word: 27.000 linee di codice nel 1982 a 2.000.000 nel 2002

Comprendiamo quindi come la continua aggiunta di nuove funzionalità al software esistente, per adeguarlo ai tempi e alle aspettative dell'utenza, giustifica la costante richiesta di processori più veloci, memorie sempre più grandi e più ampie capacità di I/O (Input/Output).

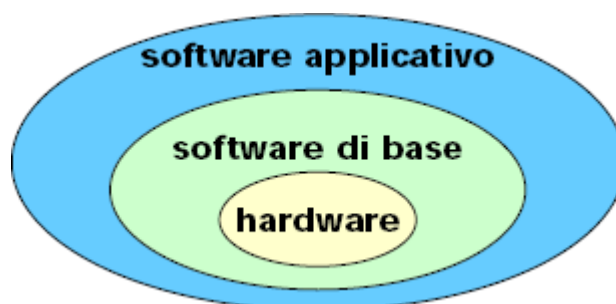
Si può quindi notare come anche le altre tecnologie si siano evolute di pari passo. Myhrvold traccia un parallelismo con la legge di Moore: "abbiamo aumentato la dimensione e la complessità del software ancora più rapidamente di quanto non prevedeva la legge di Moore", "gli utenti del software hanno sempre consumato le maggiori capacità di elaborazione ad una velocità uguale o superiore a quella con cui i produttori di chip le mettevano a disposizione" (Stewart Brand, 1995). Da questo comprendiamo quanto le due evoluzioni siano legate tra loro. Se vogliamo sono legate in modo bidirezionale, l'una influenza l'altra. Il software richiede capacità computazionali che impattano sulle componenti hardware. Le componenti hardware, dal canto loro, influenzano totalmente il funzionamento del software.

3. Tipi di software

L'hardware e il software sono organizzati a livelli (o strati). Ciascun livello corrisponde a una macchina dotata di un proprio insieme di funzionalità. Ogni macchina è caratterizzata da un proprio linguaggio, formato dalle istruzioni che quella macchina sa eseguire. Ciascun livello fornisce un linguaggio più semplice da utilizzare rispetto a quello del livello sottostante. Ciascun livello è realizzato in termini del linguaggio del livello immediatamente sottostante.

Il software può essere diviso in due grandi categorie:

- il **software di base**: racchiude in sé sia il software di sistema, necessario a far funzionare l'elaboratore, ovvero il sistema operativo, sia quello utilizzato dagli sviluppatori di programmi per facilitare il loro lavoro;
- il **software applicativo**: comprende invece tutti quei programmi utilizzati dagli utenti per gestire, per esempio, la posta, la contabilità, per redigere una lettera, creare una presentazione, inviare un messaggio, ecc.; oppure applicativi creati ad hoc per risolvere un determinato problema.



Software e hardware sono organizzati a livelli

Nell'organizzazione a livelli l'hardware rappresenta l'unica macchina reale, mentre gli strati software corrispondono a cosiddette macchine virtuali. Le diverse macchine e i relativi linguaggi divengono sempre più astratti, in altre parole il significato di ciascuna operazione è sempre più

vicino alla logica dell'utente e più lontano dalla logica del calcolatore. Pertanto man mano che si sale nella gerarchia le diverse macchine e i relativi linguaggi diventano anche via via più semplici da usare.

4. Licenze d'utilizzo e distribuzione

C'è un concetto molto importante da affrontare, seppur senza entrare nei dettagli legali dell'argomento, e cioè le licenze di utilizzo e distribuzione. Ci capita infatti spesso di imbatterci in diversi tipi di licenze sui software e queste condizionano l'utilizzo che l'utente può farne.

La licenza d'uso è un documento che accompagna il software e specifica i diritti e i doveri di chi lo riceve e di chi lo diffonde. Tutte le licenze d'uso traggono il loro valore legale dalle norme sul diritto d'autore, il copyright.

Esistono diversi tipi di licenze:

- licenze libere
- licenze Open Source
- licenze proprietarie

Le licenze di utilizzo e distribuzione del software libere ed Open Source sono numerose, ma quelle effettivamente diffuse sono poche. Per l'89% si tratta di GPL, LGPL e BSD (licenza storica di Unix, tornata in uso dall'avvento di Linux).

Alcune licenze libere sono:

- GNU-GPL (la licenza libera più diffusa)
- GNU-LGPL
- BSD
- Creative Commons

Ogni tipo di licenza differisce dagli altri per vari aspetti. Le varie licenze analizzano, in linea di massima, i seguenti aspetti:

- Il permesso d'autore (copyleft)
- Il diritto d'autore (copyright)
- Il software proprietario
- Il software libero (free software)

- Il software Open Source

Analizziamo le differenze di contratto di licenza d'uso per il software. Queste possono essere così schematizzate e descritte:

Adware:

Software gratuito, che però obbliga l'utente alla visualizzazione di banner pubblicitari in qualche punto della schermata. Attenzione: molti adware sono anche spyware, trasmettono cioè informazioni sull'utente, allo scopo di selezionare gli annunci più adatti da trasmettere.

Cardware:

Detto anche "Postcardware" o "Mailware", è software gratuito in cui l'autore chiede agli utenti di mandargli una cartolina o un'e-mail. In pratica una forma di freeware, che si paga con un semplice gesto di cortesia verso l'autore. Esistono varie altre forme simili in cui l'autore richiede all'utente di farsi vivo, magari con idee e suggerimenti (ideaware, commentware ecc.).

Commerciale:

Questo è un software che si può usare legalmente solo dopo averlo acquistato, come quello che si trova nei negozi.

Demo:

È un programma dimostrativo, distribuito per far conoscere il software. I giochi in versione demo permettono di iniziare il gioco, ma non di finirlo. Un software in versione demo può, per esempio, permettere di editare un file ma non di salvare la versione modificata. Alcuni demo possono avere funzionalità utili, ma nella maggioranza dei casi è necessario acquistare la versione completa per poter utilizzare il software.

Freeware:

Questa dicitura significa che il programma è distribuito gratuitamente. È dunque perfettamente funzionante e non richiede alcun pagamento.

GPL:

Acronimo di GNU Public License, cioè licenza pubblica GNU, dove GNU è a sua volta l'acronimo, scherzoso e ricorsivo di GNU's Not UNIX (GNU non è UNIX). Si tratta di una particolare

forma di software open source, la cui diffusione deve rispondere a precise regole codificate nella licenza.

Il software GPL è "software libero": può cioè essere parte di pacchetti commerciali, ma è liberamente copiabile. Non solo: è liberamente studiabile e modificabile, purché le versioni modificate vengano ugualmente messe a disposizione come software libero. Il sistema operativo Linux, per fare un esempio, è GPL: può essere liberamente copiato, anche se esistono delle "distribuzioni" acquistabili da chi vuole avere, oltre al nudo sistema operativo, anche i manuali, il supporto tecnico e altri programmi a corredo.

Open source:

Software il cui codice sorgente è liberamente utilizzabile e a disposizione del pubblico. Non sempre questo significa che il programma si possa usare liberamente senza pagare una licenza. Esempi tipici di software open-source sono il browser Mozilla, il sistema operativo Linux, la suite per ufficio Open Office.

Shareware:

Sono shareware quei programmi che, pur essendo liberamente distribuibili, richiedono un pagamento per poter essere utilizzati in tutte le loro funzioni e senza limitazioni di tempo. A volte il programma è pienamente utilizzabile, e il pagamento è lasciato all'onestà dell'utente. Più spesso, invece, il programma ha alcune limitazioni alla funzionalità, crea artificialmente ritardi e fastidi all'utente (come la visualizzazione di schermate di ammonimento), oppure smette di funzionare dopo un certo tempo, e per ottenere che funzioni in modo completo e permanente occorre pagare per ottenere l'apposito codice di sblocco. Gli shareware che si limitano ad ammonire l'utente vengono scherzosamente detti "nagware" (dal verbo "to nag", che significa, più o meno, "tampinare").

Trial: Versione di prova. È un altro modo di definire lo shareware che prevede una scadenza dopo un periodo di utilizzo gratuito, in seguito al quale occorre acquistare la versione completa.