

Il Pensiero Computazionale ¹

o come pensare da informatici

Daniele Francesco Santamaria

Università degli studi di Catania

October 3, 2023

¹Tratto da “Abbiamo davvero bisogno del pensiero computazionale?” M. Lodi, S. Martini, E. Nardelli

Cos'è il pensiero computazionale?

Il pensiero computazionale è l'insieme dei processi mentali usati per modellare una situazione (analizzare, comprendere e rappresentare), formulare un problema e risolverlo alitmicamente, ovvero deve essere risolvibile da un agente computazionale (non consapevole?) che non conosce il significato delle istruzioni e dei dati che deve elaborare.

Il pensiero computazionale richiede metodi, pratiche e competenze.

Cosa vuol dire algoritmo?

Un Algoritmo è un insieme finito e formale (interpretabile in modo diretto, univoco e non ambiguo, sintassi e semantica ben precisi) di istruzioni elementari (quindi non scomponibili ulteriormente, principio di atomicità) che deve essere eseguito in tempo finito (principio di terminazione), condurre sempre al medesimo risultato (principio di determinismo), ed essere applicabile per la famiglia di problemi individuata (principio di generalità).

Quali sono i processi mentali?

I processi mentali sono strategie mentali utili per risolvere problemi

- Pensiero logico: usare la logica e il ragionamento per convincere (-si) di qualcosa, stabilire e controllare fatti.
- Scomposizione di problemi: dividere un problema complesso in semplici sottoproblemi, risolvibili in modo più semplice; modularizzare; usare il ragionamento compositivo.
- Astrazione: liberarsi dei dettagli inutili per concentrarsi sulle informazioni/idee rilevanti.
- Riconoscimento di pattern: individuare regolarità/schemi ricorrenti nei dati e nei problemi.
- Pensiero algoritmico: usare il pensiero algoritmico per progettare una sequenza ordinata di passi (istruzioni) per risolvere un problema, ottenere un risultato o portare a termine un compito.

Quali metodi adopera il pensiero computazionale?

I metodi sono approcci operativi utilizzati per pensare computazionalmente

- Automazione: automatizzare soluzioni; usare un computer o una macchina per eseguire compiti ripetitivi o noiosi.
- Programmazione: usare alcuni concetti di base della programmazione (cicli, eventi, istruzioni condizionali, operatori logici, ecc.).
- Raccolta, analisi e rappresentazione dei dati: raccogliere informazioni e dati, interpretarli trovando schemi ricorrenti, rappresentarli in maniera appropriata; memorizzare, recuperare e aggiornare dati.
- Parallelizzazione: eseguire compiti simultaneamente per raggiungere un obiettivo comune, pensare “in parallelo”.
- Simulazione: rappresentare dati e processi (del mondo reale) tramite modelli; eseguire esperimenti su tali modelli.
- Valutazione: analizzare le soluzioni implementate per giudicarne la bontà, in particolare per ciò che riguarda la loro effettività e la loro efficienza in termini di tempo impiegato o di spazio occupato.

Quali pratiche adopera il pensiero computazionale?

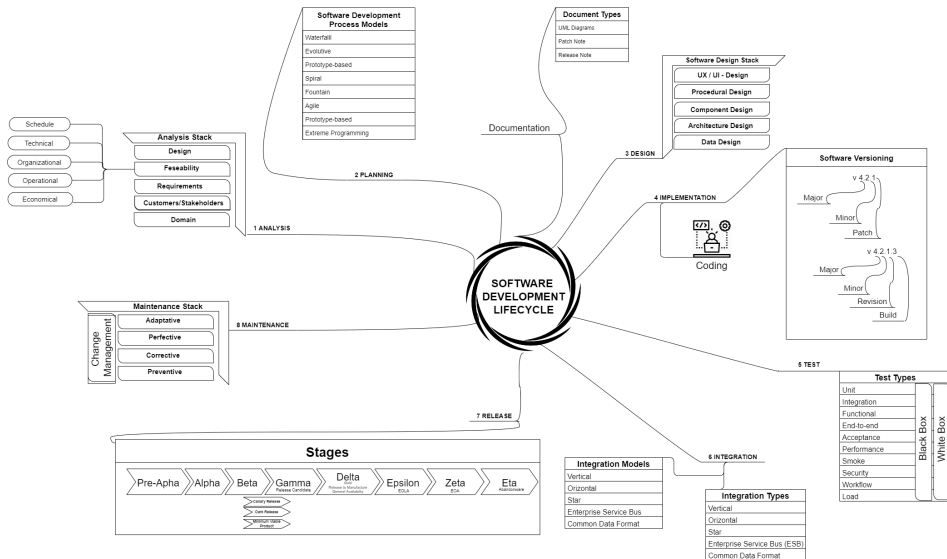
Le pratiche sono attività impiegamente per realizzare soluzioni

- Iterare: un progetto viene sviluppato attraverso ripetizioni di un ciclo “progetta-costruisci-verifica” (nelle metodologie di sviluppo software incrementali e iterative), costruendo in modo incrementale il risultato finale. Approccio dall’alto (Top-Down) o dal basso (Bottom-Up).
- Prova e sbaglia, tinkering: per imparare spesso bisogna sbagliare. Imparare dal gioco, dall’esplorazione e dalla sperimentazione.
- Testare e correggere gli errori (debug): verificare che le soluzioni funzionino provandole concretamente; trovare e risolvere i problemi (bug) in una soluzione o in un programma.
- Riuso: costruire la propria soluzione basandosi su/utilizzando anche codice, progetti o idee già esistenti.
- Scrivere e leggere documentazione: spiegare adeguatamente la propria soluzione in modo tale che sia accessibile, adattabile ed eventualmente modificabile. Lo scrittore di documentazione è soprattutto un lettore di documentazione.

Quali competenze sviluppare?

Il pensiero computazionale richiede delle competenze personali da sviluppare

- Creare: progettare e costruire artefatti, usare la computazione per essere creativi ed esprimere se stessi.
- Comunicare e collaborare: connettersi con gli altri e lavorare insieme con un obiettivo comune per creare qualcosa e per ottenere una soluzione migliore.
- Riflettere, imparare, fare meta-cognizione: usare l'informatica per riflettere e comprendere gli aspetti computazionali del mondo.
- Tollerare l'ambiguità: avere a che fare con problemi reali, aperti e non totalmente specificati a priori.
- Perseverare quando si ha a che fare con problemi difficili: essere a proprio agio nel lavorare con problemi difficili/complessi, essere determinati, resilienti e tenaci.



Una persona non comprende davvero un argomento se non quando riesce ad insegnarlo ad un computer. (Donald Knuth).

Sono convinto che l'informatica abbia molto in comune con la fisica. Entrambe si occupano di come funziona il mondo a un livello abbastanza fondamentale. La differenza, naturalmente, è che mentre in fisica devi capire come è fatto il mondo, in informatica sei tu a crearlo. Dentro i confini del computer, sei tu il creatore. Controlli – almeno potenzialmente – tutto ciò che vi succede. Se sei abbastanza bravo, puoi essere un dio. Su piccola scala. (Linus Torvalds)

L'informatica non riguarda i computer più di quanto l'astronomia riguardi i telescopi. (Edsger Dijkstra)