

Una pipeline di sviluppo di un software si sviluppa in questo modo:

- Si definisce dapprima il problema che il nostro software deve risolvere
- Dopodichè si definisce la struttura del sistema da sviluppare, ideando un progetto architeturale (come è strutturato in sè per sè) e un progetto di design (ossia come appare l'applicativo)
- Si codifica e si implementa il progetto, utilizzando tecnologie adeguate, ossia quelle efficienti ed efficaci. Qui si sceglie per esempio il linguaggio di programmazione.
- Dopo aver codificato tutto segue una più o meno lunga fase di collaudo, manutenzione e deployment, dove è necessario risolvere la maggior parte dei bug presenti nel codice.

La pipeline è difficile da rispettare, infatti molto spesso i progetti falliscono poiché la deadline non viene rispettata a causa di una pipeline non adeguatamente seguita.

Un progetto fatto bene deve seguire questi due spazi:

- **Problem Space**, ossia lo spazio "reale" delle cose che compongono il problema che il nostro software andrà a risolvere. È di fatto il dominio del nostro progetto.
- **Solution Space**, ossia lo spazio "artificiale" composto dalle soluzioni del problema preso in considerazione. Bisogna quindi mappare il Problem Space all'interno del Solution Space.

Se il problem space è il problema da risolvere, il solution space è come implemento la soluzione a tale problema.

L'obiettivo è riuscire di rendere la mappatura la più semplice ed efficace possibile. Ciò è reso più semplice tramite l'astrazione, che è la via di mezzo tra i due spazi, ossia l'insieme delle teoriche implementazioni per rendere facile e ottimale la mappatura; l'insieme di concetti che rendono ciò possibile si chiama **livello di astrazione**.

Ogni linguaggio di programmazione ha uno specifico scopo, spesso dettato dal paradigma che sfruttano. C è per esempio incentrato su funzioni e istruzioni che sfruttano la memoria e l'hardware nella migliore maniera possibile, rendendo facile l'implementazione di sistemi operativi; Java è invece un linguaggio orientato agli oggetti, utile per implementare software generici.

La programmazione a oggetti **ha diversi vantaggi**:

- Ha poche, ma utilissime, astrazioni chiave (classe, oggetto, metodo, campo). Esse sono orientate a rendere il più possibile semplice **la modellazione di un problem space** e renderlo facilmente mappabile ad un solution space.

- Se realizzato bene, un codice è facilmente modificabile e comprensibile da chiunque voglia interagirvi. Inoltre, sempre per lo stesso motivo, il codice può essere riusabile da chiunque in maniera semplice.
- I linguaggi OOP sono implementabili in linguaggi C-like, ossia simili al C ma con le implementazioni che li rendono orientati agli oggetti.

Cos'è un oggetto?

"Un oggetto è un'entità computazionale dotato di stato, comportamento e identità"

- Ogni oggetto, anche se presente in multiple copie, ha una propria identità, **ossia va trattato singolarmente**
- Ogni oggetto ha un proprio stato, **ossia una informazione che porta con sé.**
- Ogni oggetto **tratta le informazioni al suo interno in modo diverso**, ossia con comportamento diverso.

In modo più dettagliato, ci sono 5 caratteristiche della OOP:

- Tutto è un oggetto: un oggetto è un'entità che fornisce operazioni che gli consentono di operare in un certo modo. Ogni cosa può quindi essere modellata ad oggetto.
- Un programma descrive un insieme di oggetti che decidono cosa fare scambiandosi messaggi. Un oggetto comunica ad un altro che deve svolgere un'azione, e pertanto quest'ultimo invia un messaggio ad un altro ancora...
- Un oggetto ha una memoria fatta da altri oggetti; possono essere composti quindi da altri oggetti. Ad esempio un corso di laurea si compone di vari insegnamenti e vari esami
- Un oggetto è istanza di una classe - ossia un elemento di uno specifico insieme (l'elemento è l'istanza, mentre l'insieme è la classe). Questo implica che gli elementi di una classe hanno tutti la stessa forma, che è determinata dalla classe di appartenenza.
- Tutti gli oggetti di una classe possono ricevere gli stessi messaggi, per il motivo precedente.