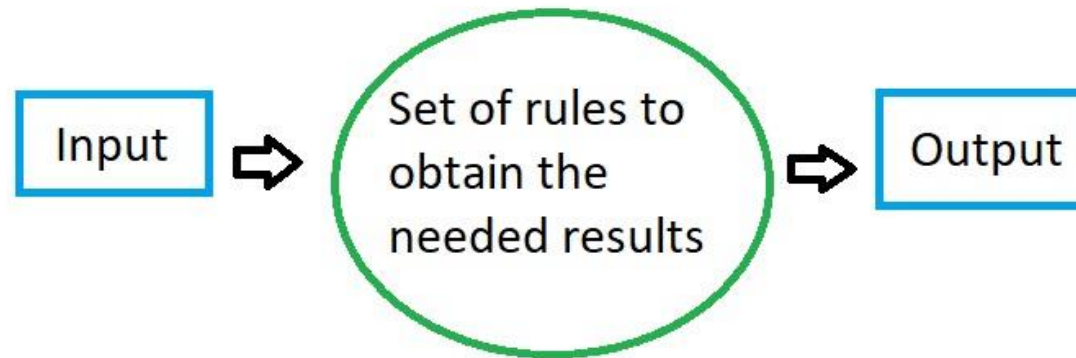


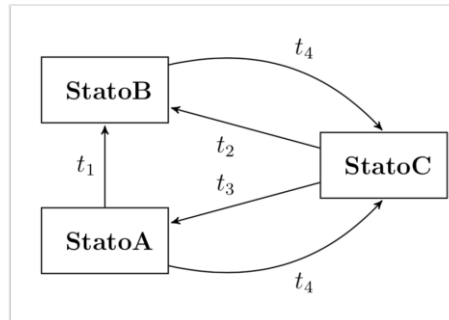


Cos'è un algoritmo

Un **algoritmo** è un insieme di istruzioni che devono essere eseguite per risolvere un **problema**

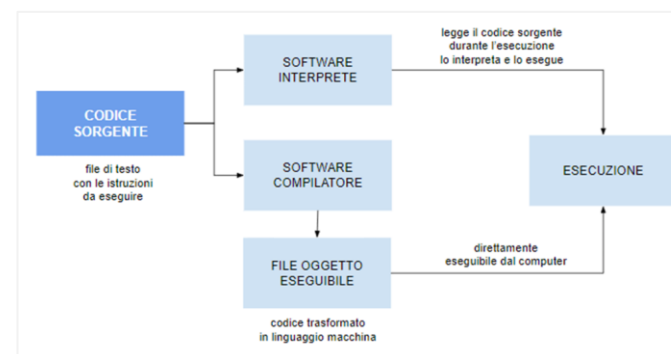
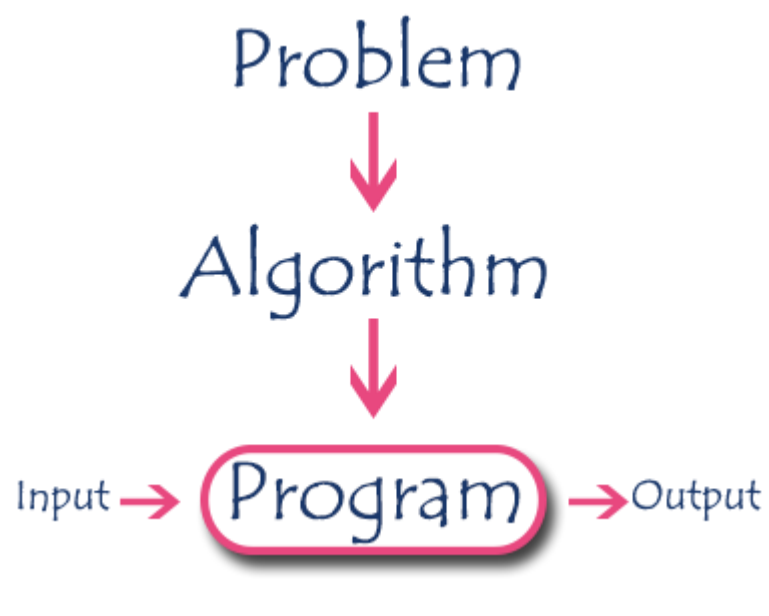


Il contesto dei dati (interni, esterni, dinamici, statici, ...) in un determinato momento può generare output e stati diversi



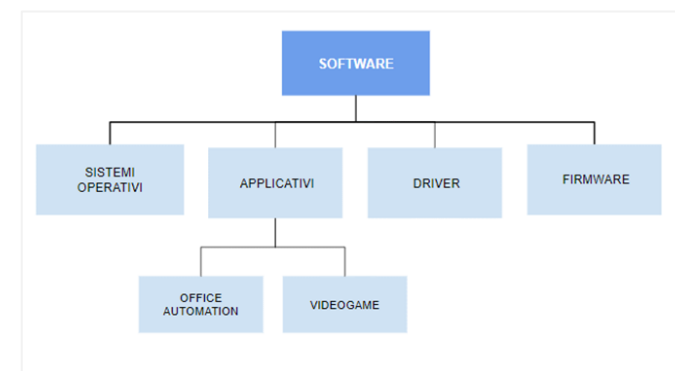
Programma vs Software

Un **programma** è la traduzione di un algoritmo, tramite un linguaggio di programmazione (codice sorgente), in una forma eseguibile o interpretabile dal computer. Un **software** è un insieme di programmi anche con diversi linguaggi.

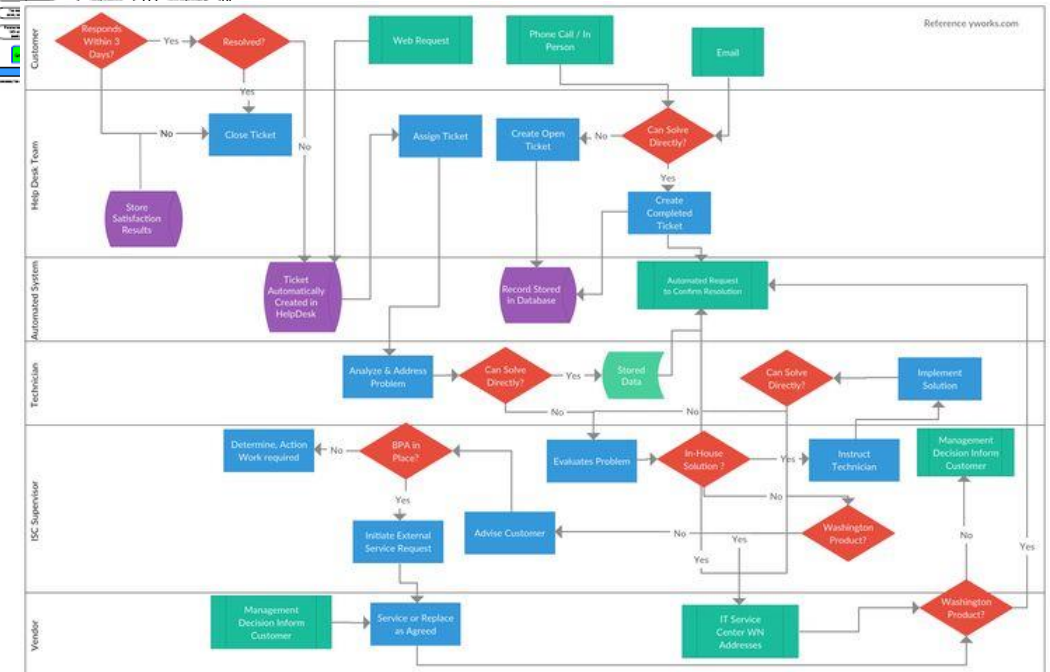
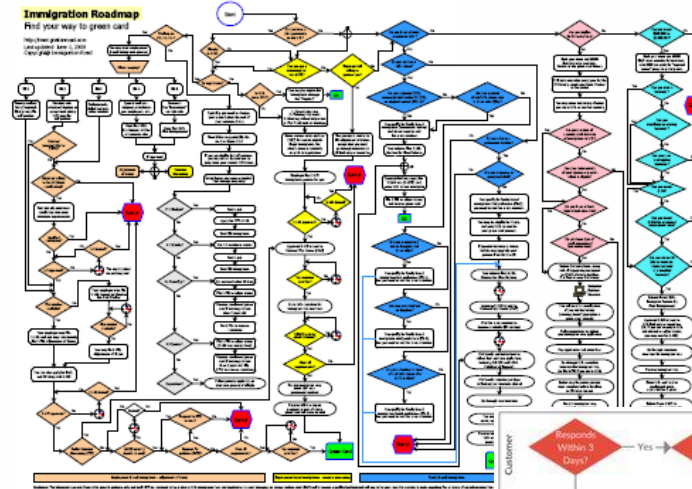
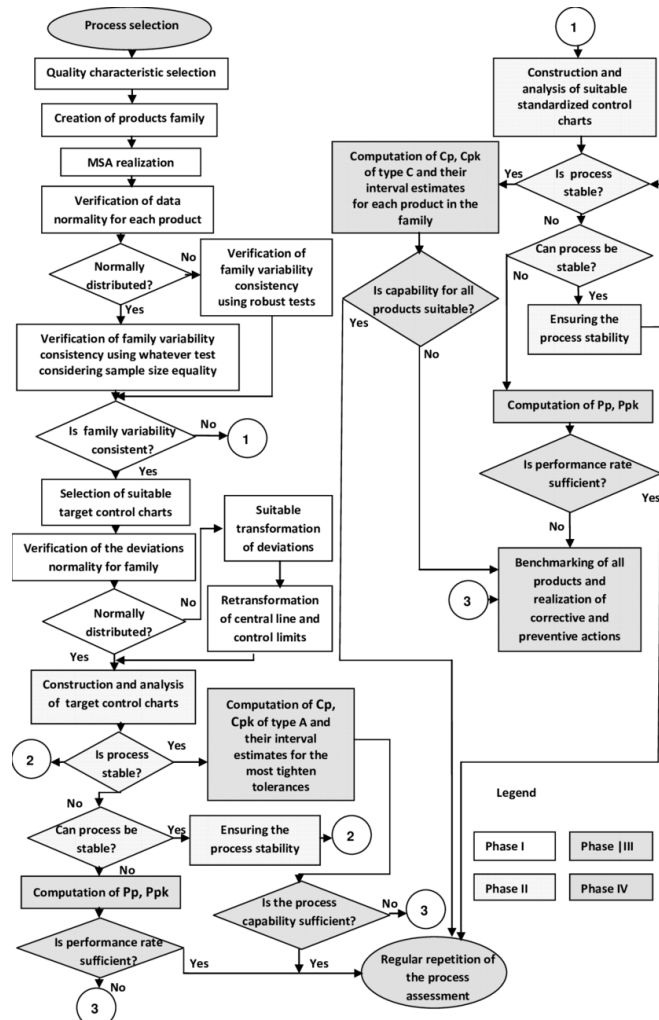


Nei linguaggi **compilati**, il compilatore produce un eseguibile in codice macchina

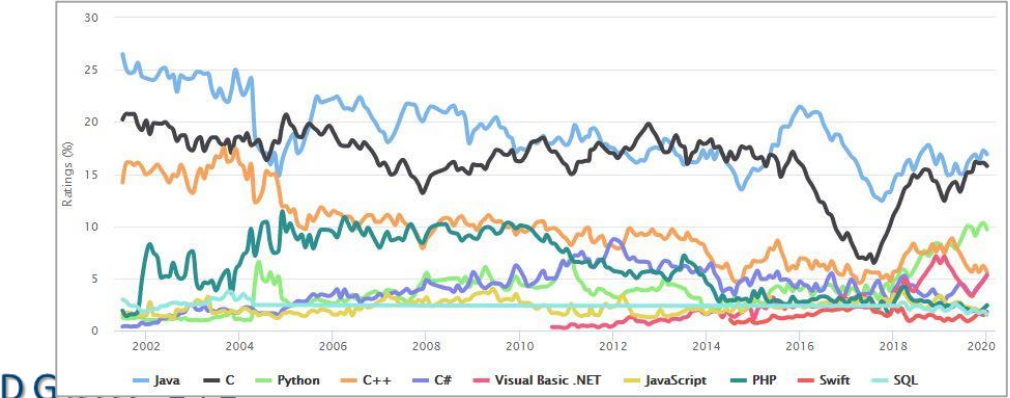
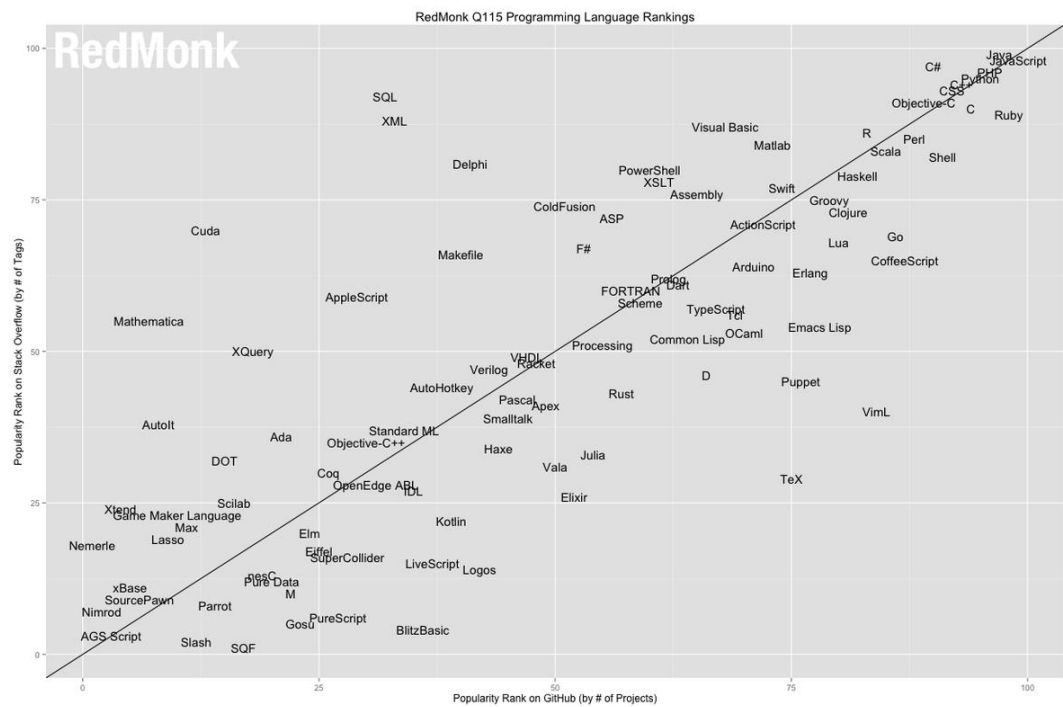
Nei linguaggi **interpretati**, le istruzioni del codice sorgente vengono eseguite direttamente, senza essere trasformate in codice macchina.



Descrivere un algoritmo: metodo classico procedurale

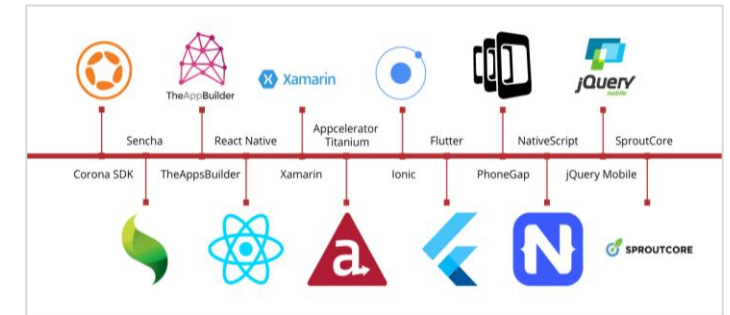
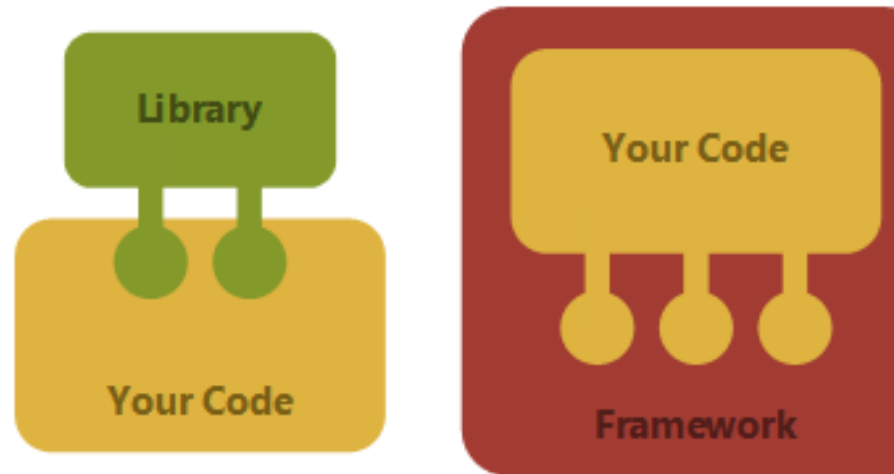
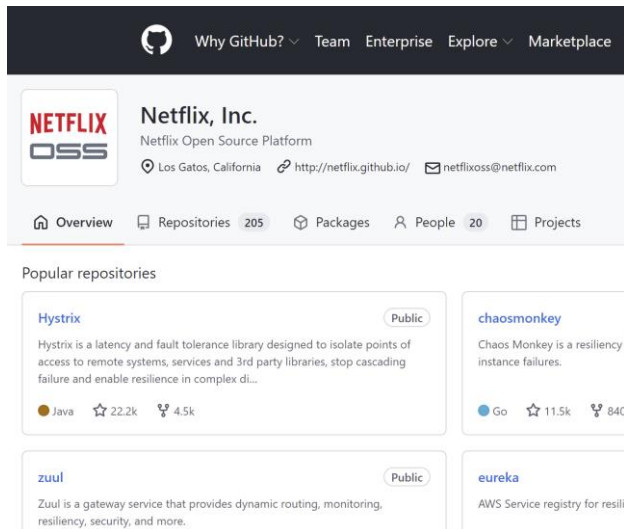


Linguaggi di programmazione



Software, Librerie, Framework....

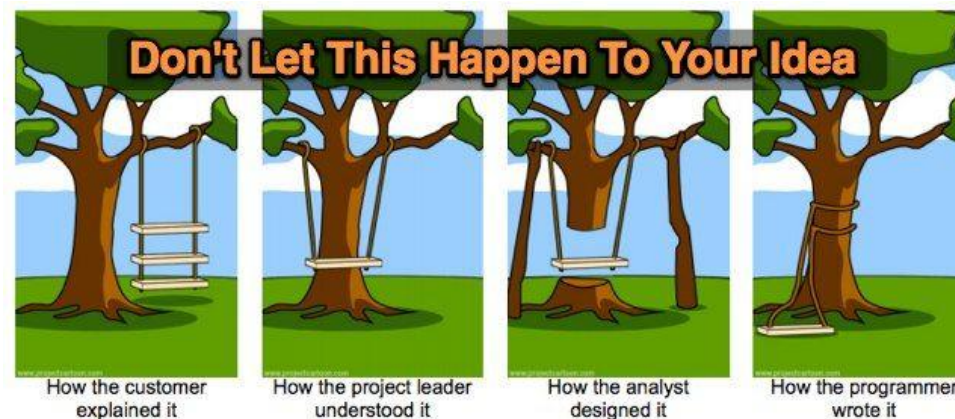
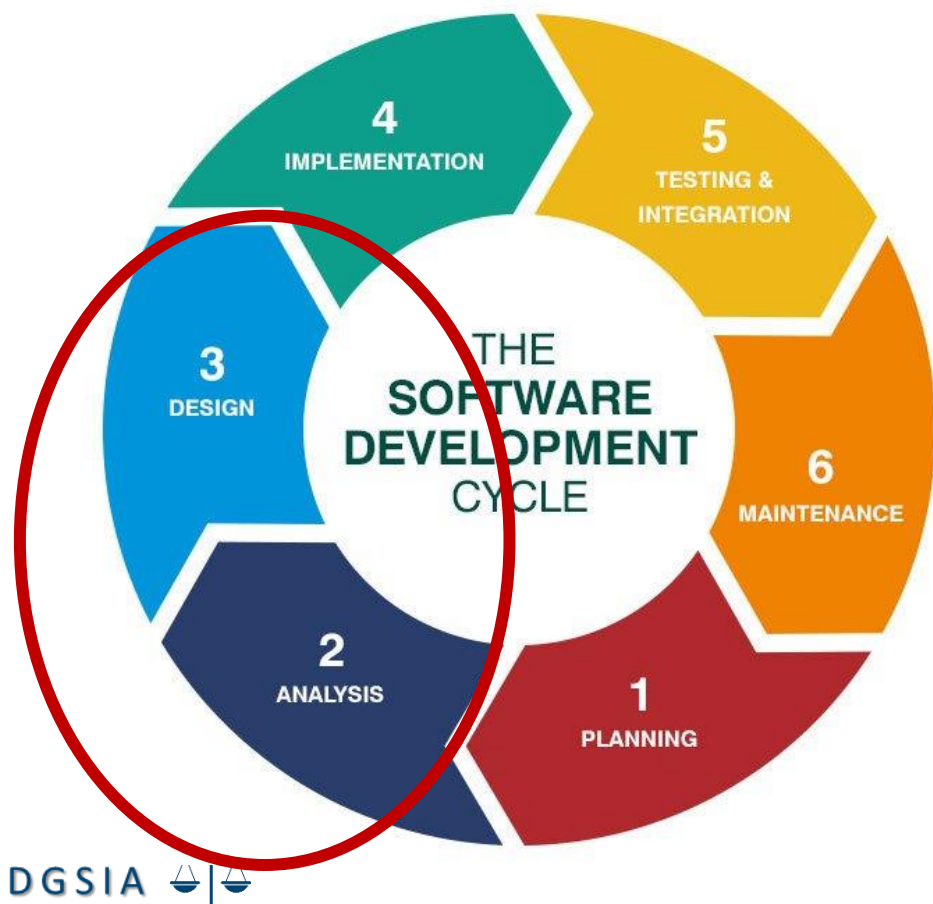
Le **librerie** sono un insieme di funzioni (sempre codice) che un programmatore «invoca» per una specifica funzionalità (es. conversioni date). Migliorano l'organizzazione di un programma, permettono di non riscrivere più volte le stesse cose, spesso si usano librerie già pronte (es. open-source, ma attenzione al tipo di licenza).



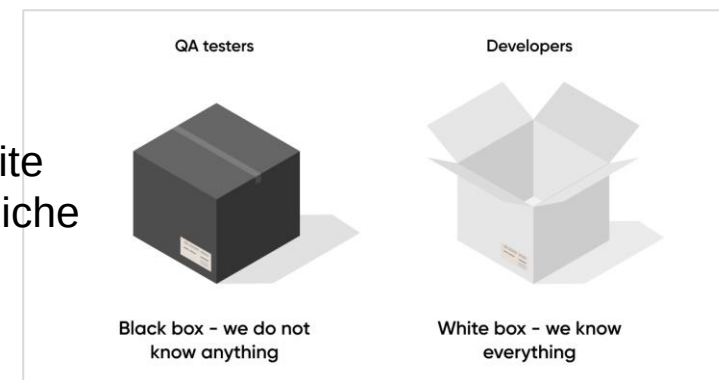
Un **framework** è una «piattaforma» di base su cui costruire il programma, con componenti di base (interazione sistema operativo, elementi di grafica, ...). Il processo di sviluppo è «semplificato» poiché gli sviluppatori non devono reinventare la ruota ogni volta. Anche questi possono essere open o non, richiedono

Il ciclo di vita del software ... come si «produce»

Nella fase di analisi (requisiti utente) e progettazione (specifiche tecniche) del ciclo di vita del software, si definisce cosa deve fare il software.

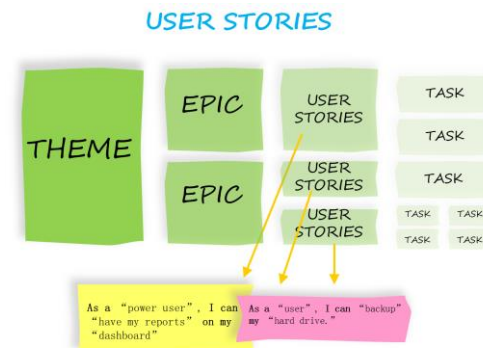
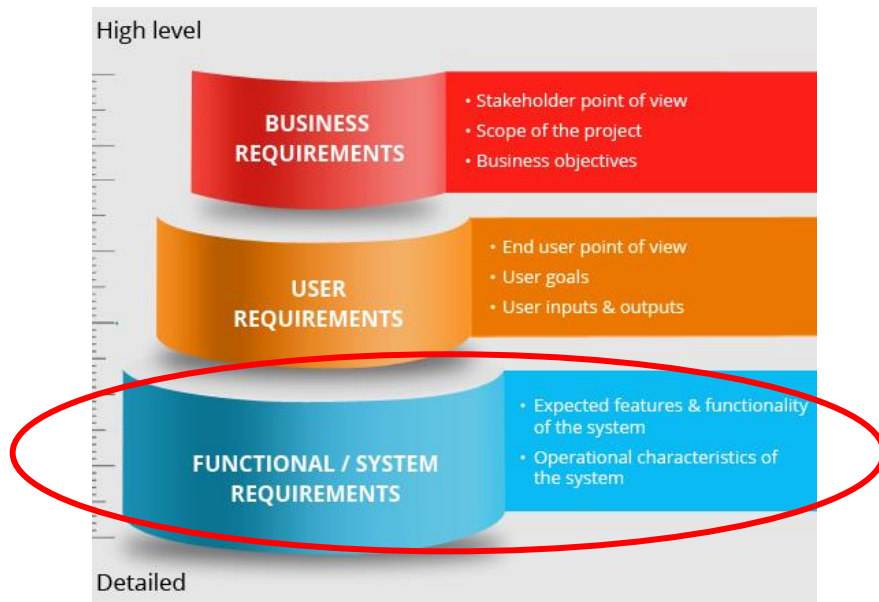


Termini Black Box e White Box si riferiscono a tecniche del test del codice



Requisiti e specifiche

I requisiti possono avere dei dettagli diversi rispetto a chi li deve definire ed interpretare. Ci sono molteplici tecniche e metodologie di rappresentazione. Ma ci devono sempre stare. Uno sviluppatore deve poter avere una descrizione di cosa deve implementare.



Self-Service Home Cinema

1. Introduzione

1.1 Scopo del documento dei requisiti

Il presente documento (PRD) presenta i requisiti funzionali e non funzionali per il sistema Self-Service Home Cinema, che consentirà agli utenti di gestire la propria libreria di film e di noleggiare i film in formato HD.

1.2 Scopo del prodotto

Il sistema software consentirà al cliente di gestire la propria libreria di film e di noleggiare i film in formato HD. Il sistema sarà in grado di gestire la propria libreria di film e di noleggiare i film in formato HD.

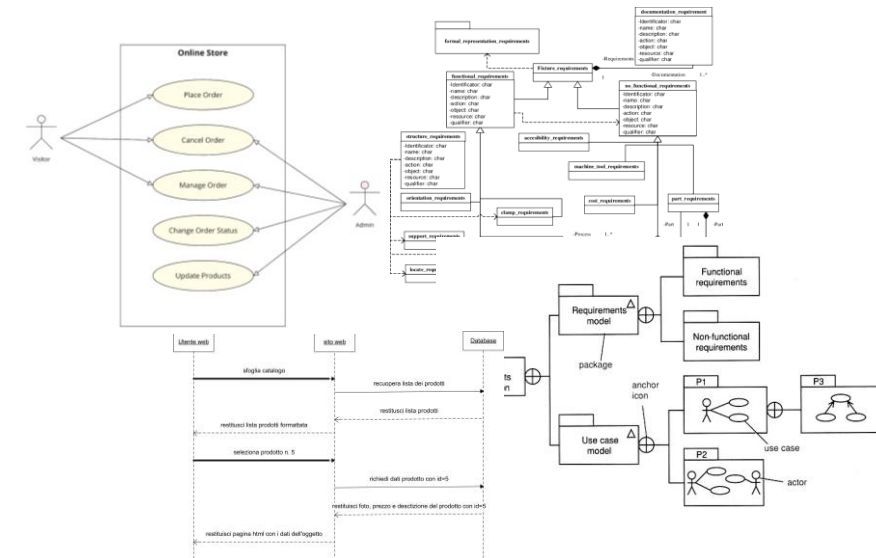
1.3 Descrizione del resto del documento

Il presente documento (PRD) è il primo di una serie di documenti che descrivono il sistema Self-Service Home Cinema. Il documento descrive i requisiti funzionali e non funzionali del sistema.

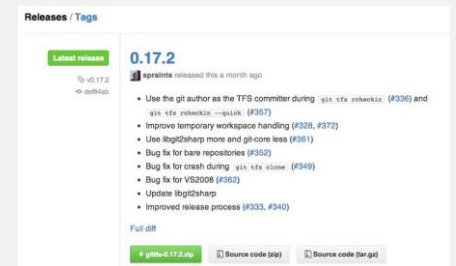
2. Descrizione generale

2.1 Prospettiva e funzioni del prodotto

Il sistema software è composto da una serie di moduli che consentono al cliente di gestire la propria libreria di film e di noleggiare i film in formato HD. Il sistema sarà in grado di gestire la propria libreria di film e di noleggiare i film in formato HD.

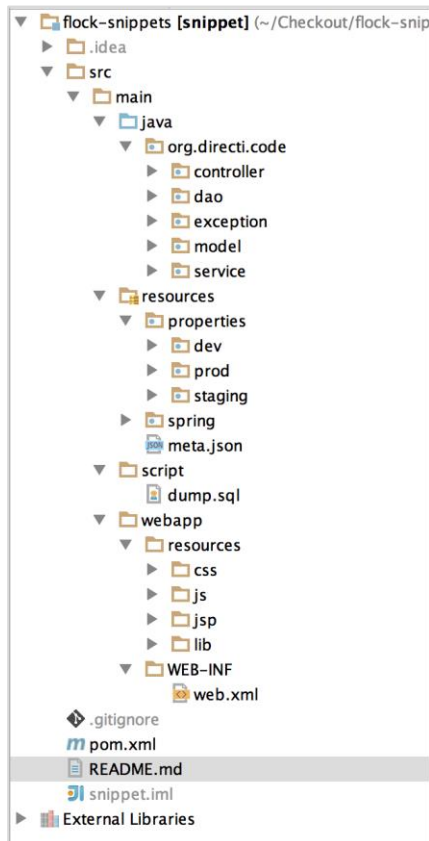


Possibilità di collegare un singolo requisito ad una release software

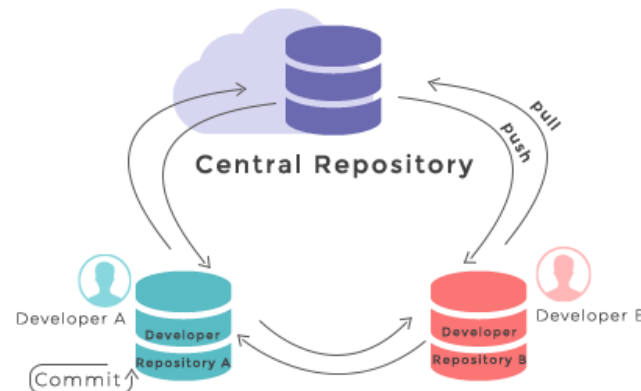


Dove sta il codice e come si aggiorna

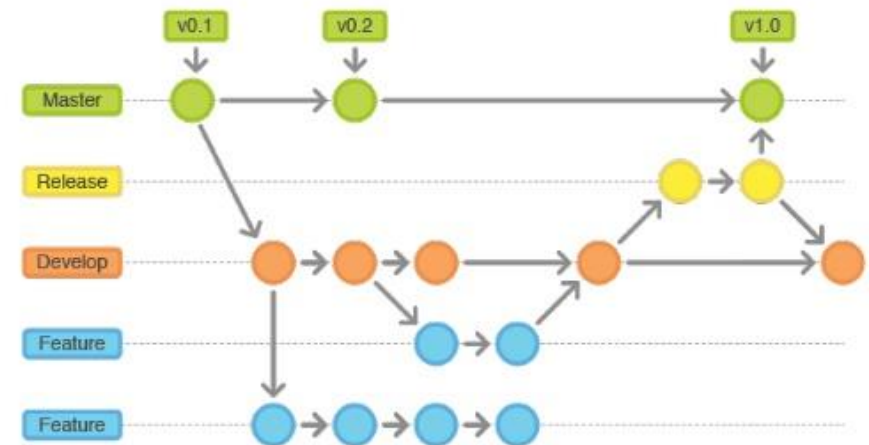
Un **repository** del codice può essere visto come una struttura di cartelle e file, spesso organizzati a regole predefinite dalla tipologia di linguaggio utilizzato.



Software development and version control: lavorando in più persone, con versioni diverse del software e continue evoluzioni, è necessario utilizzare strumenti che aiutano gli sviluppatori ad archiviare e gestire il loro codice, ed a tracciare e controllare le modifiche delle release (versioni)



Release Branches



<https://github.com/Alfresco>
<https://github.com/uber>

Codice.... Esempi semplici

<https://jsfiddle.net/macloo/bvwvd0ao/>

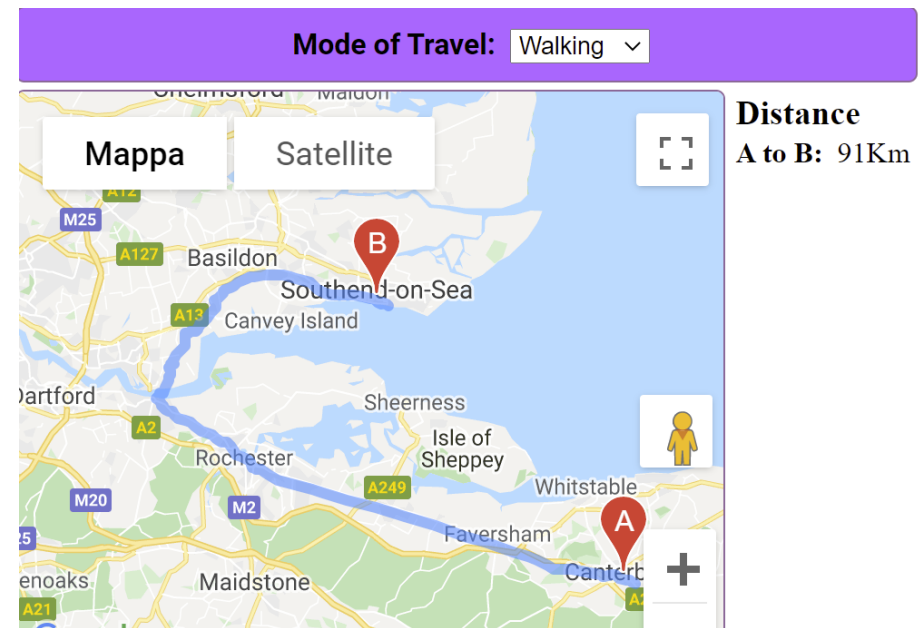
Type a number:

Type a different number:

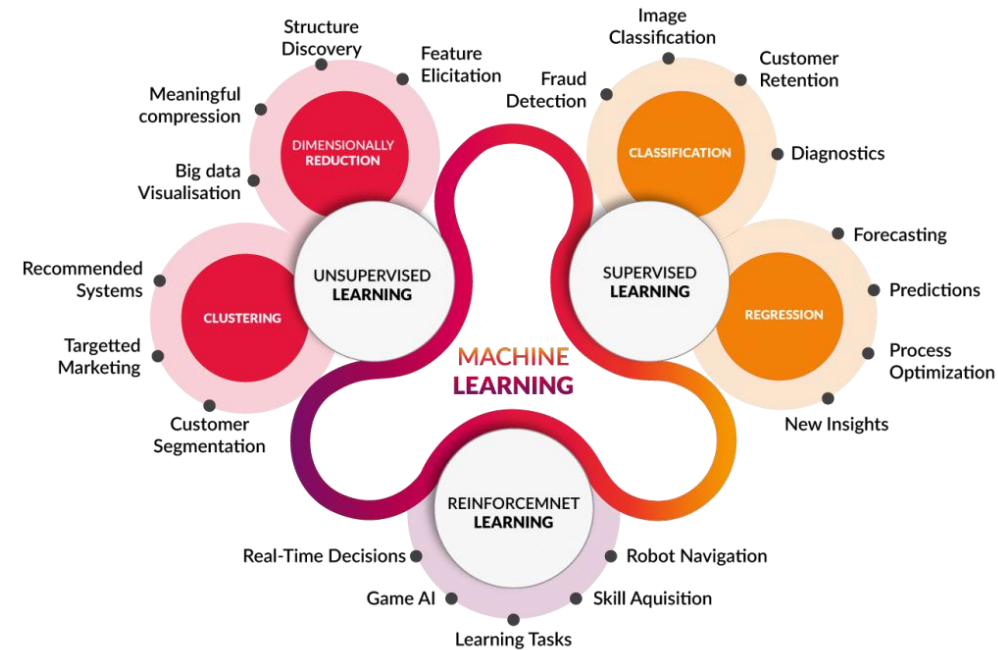
[Get the Larger](#)

The larger of the two numbers is: 4

<https://jsfiddle.net/user2314737/qxrg4o3y/>



Machine Learning...l'importanza dei dati

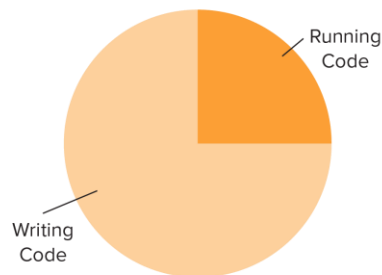


```

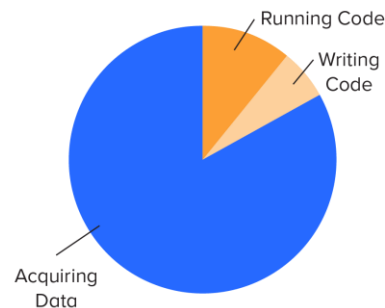
61 # Show dataset
62 show_dataset(X, Y)
63
64 # Split dataset
65 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25)
66
67 # Create logistic regressor
68 lr = LogisticRegression()
69 lr.fit(X_train, Y_train)
70 print('Logistic regression score: %.3f' % lr.score(X_test, Y_test))
71
72 # Compute CV score
73 lr_scores = cross_val_score(lr, X, Y, scoring='accuracy', cv=10)
74 print('Logistic regression CV average score: %.3f' % lr_scores.mean())
75
76 # Show classification areas
77 show_classification_areas(X, Y, lr)
78

```

How People Think
ML Works



The Reality



Training Data Needs

Training data Validation data Test data

