



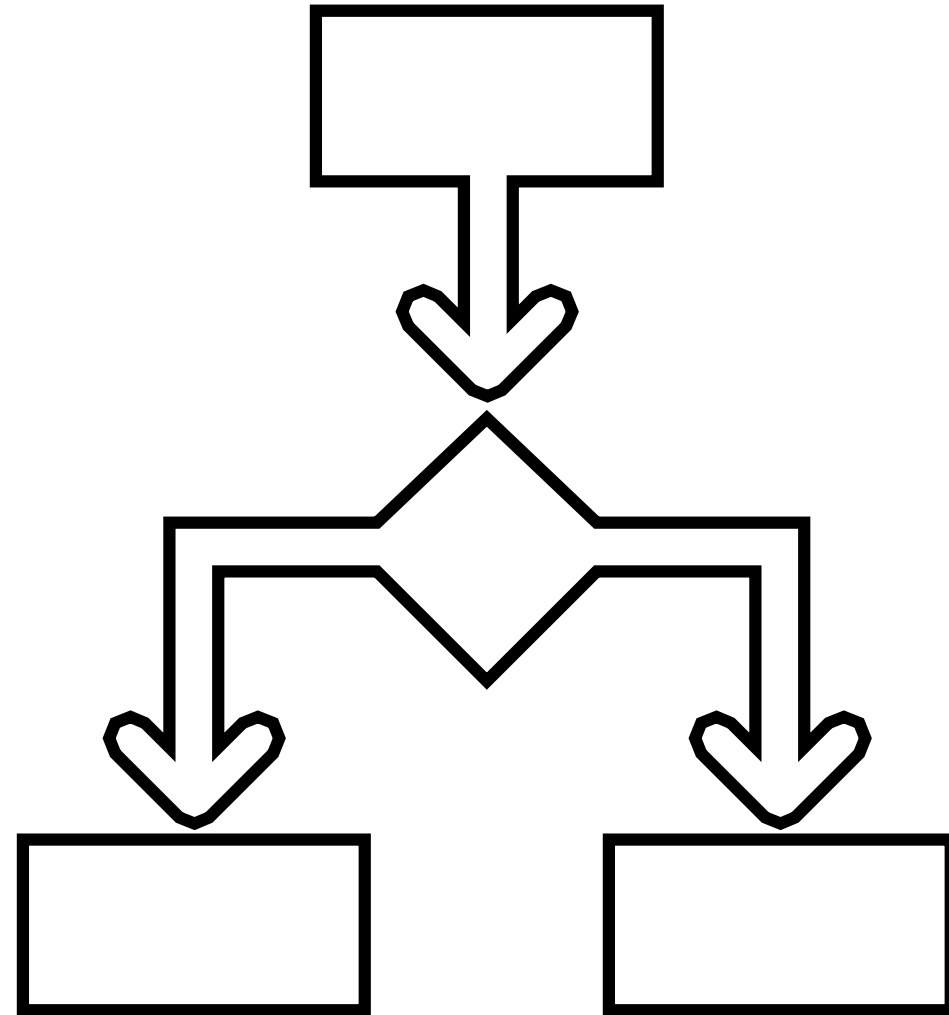
INGEGNERIA DEL SOFTWARE

COSA VEDREMO

**Fattori che
influenzano la
produzione del
software**

**Principali
tecniche di
lavorazione**

FATTORI CHE INFLUENZANO LO SVILUPPO



PROCESSO DI SVILUPPO

Parte dall'analisi del
problema fino alla sua
distribuzione

STEP PRINCIPALI

Analisi dei
requisiti

Determinazione
delle specifiche

Scelta
dell'architettura
del sistema

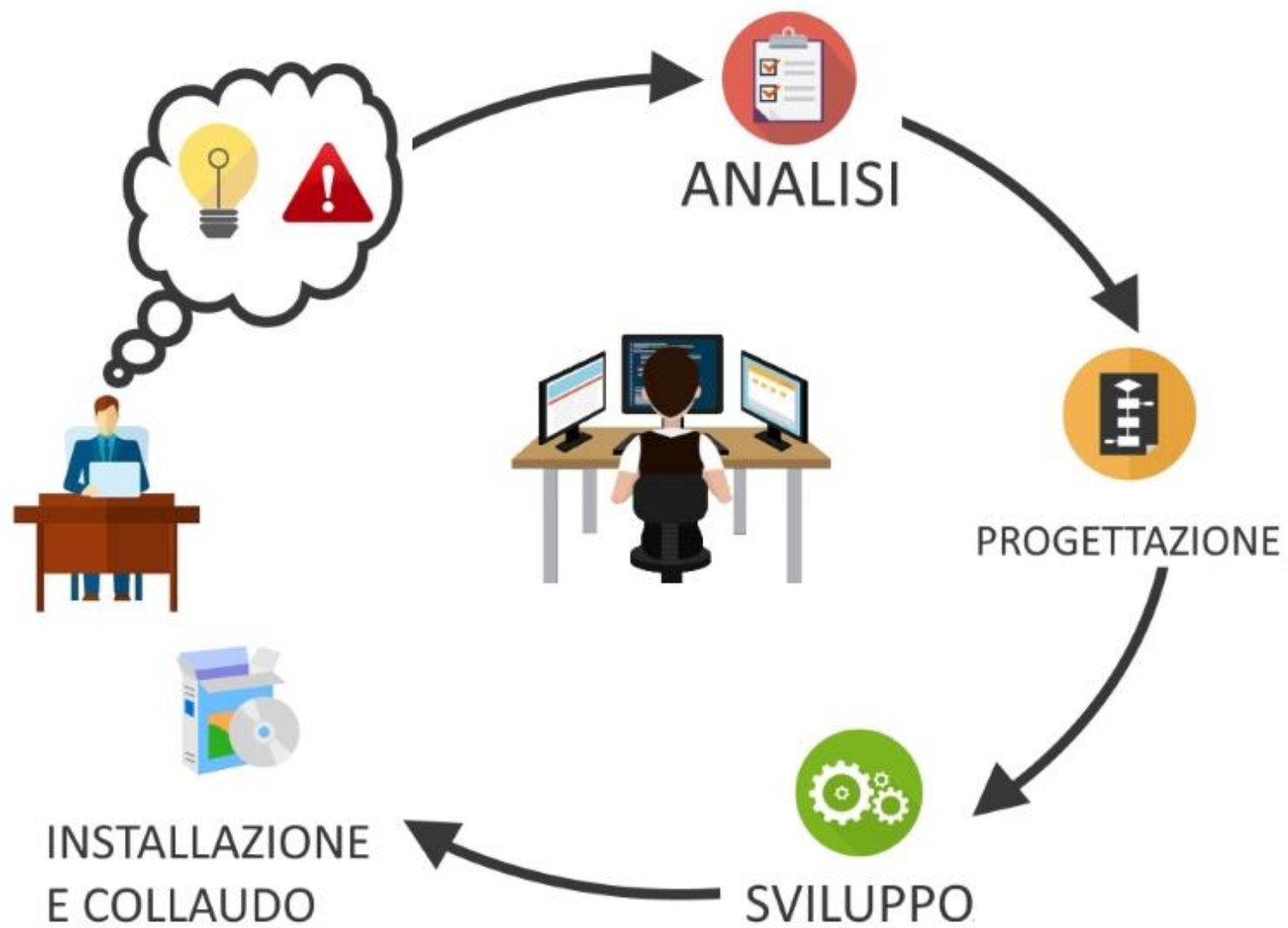
Progettazione

Test

Verifica del
funzionamento

Rilascio

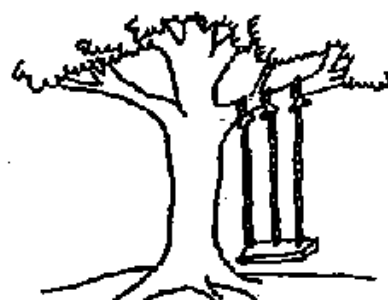
Manutenzione





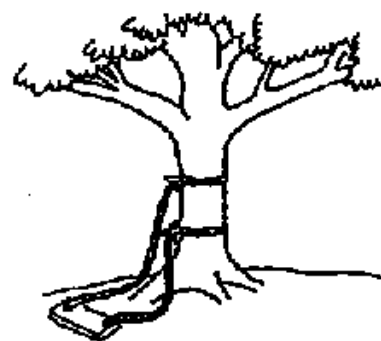
As proposed by the
project sponsor

(proposta dal committente)



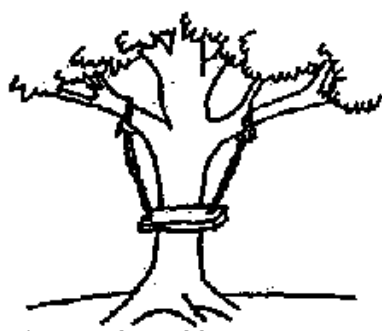
As specified in the project
request

(specificata nei requisiti)



As designed by the
senior analyst

(progettata dall'analista)



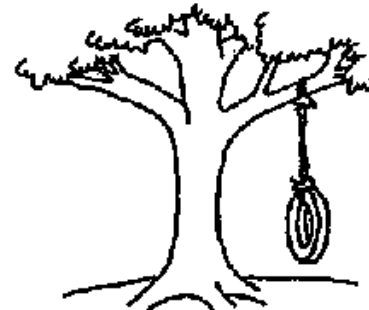
As produced by the
programmers

(prodotta dai programmatori)



As installed at the user's site

(installata presso l'utente)



What the user wanted

(ciò che l'utente voleva)

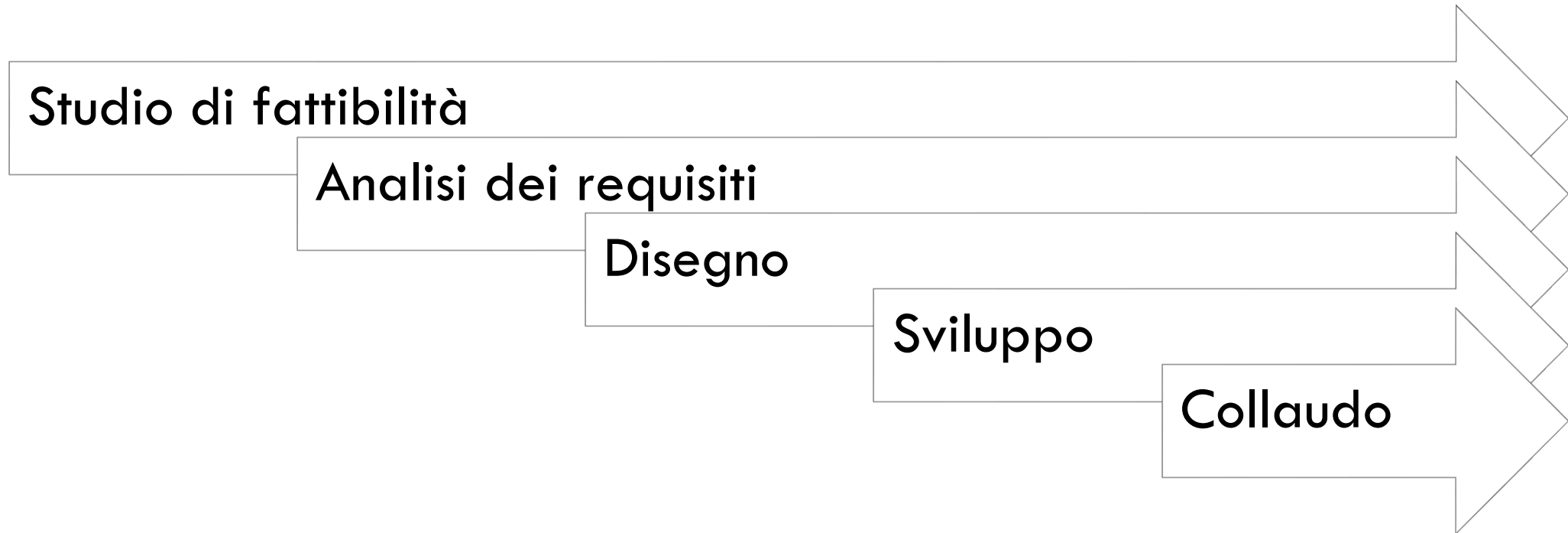
METODOLOGIE DI SVILUPPO DEL SOFTWARE

Tradizionali

Iterative

Agili

METODOLOGIE TRADIZIONALI



METODOLOGIE TRADIZIONALI

**Obbligo di terminare
completamente ogni fase
prima di passare a quella
successiva**



METODOLOGIE ITERATIVE

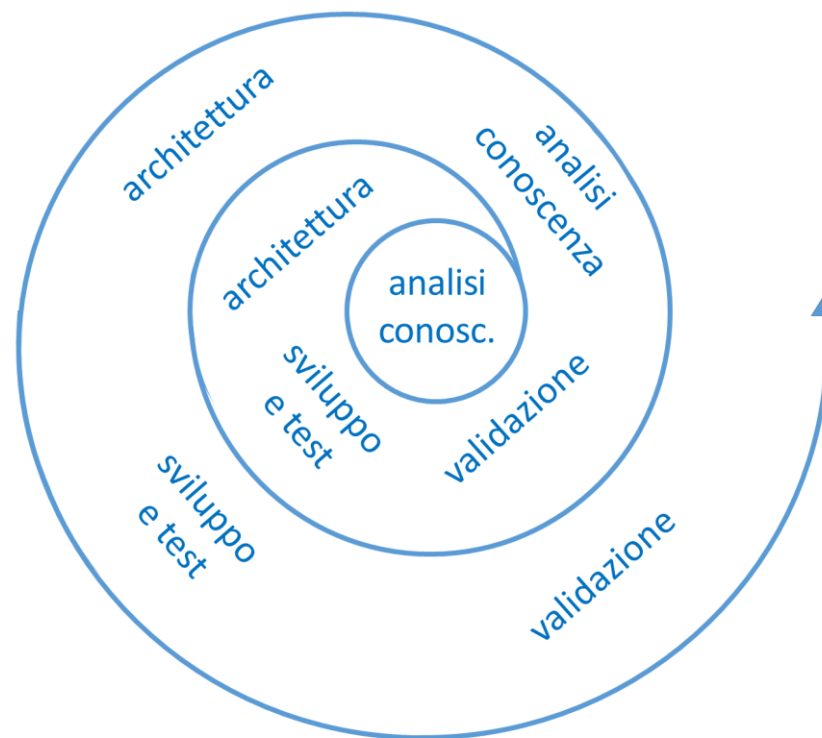
METODOLOGIE ITERATIVE

Utente viene
coinvolto

Cambiare i
requisiti più
frequentemente

METODOLOGIA SPIRALE

analisi delle esigenze (milestones)



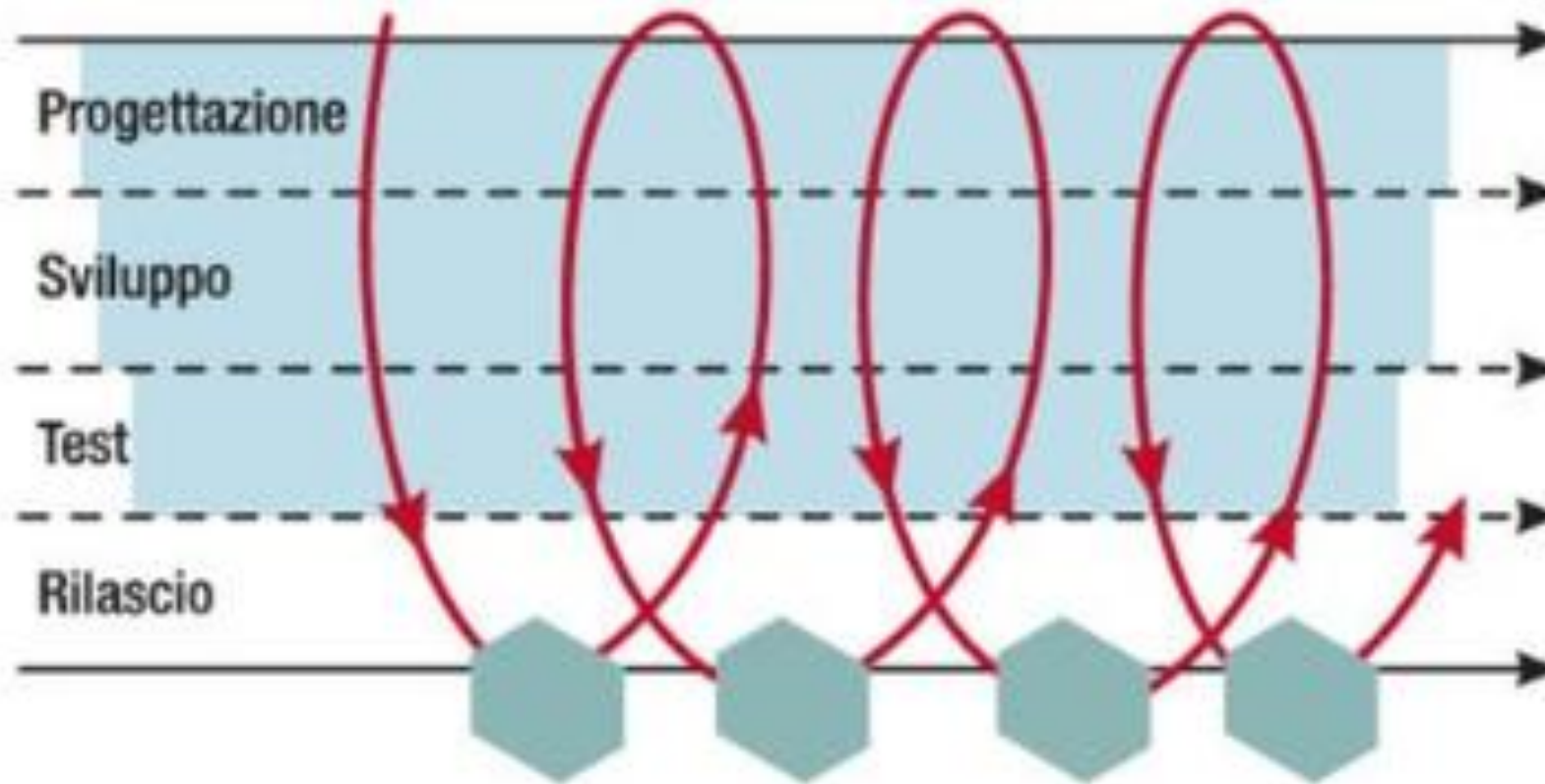
prodotto in esercizio

METODOLOGIA SPIRALE

Rilasci del
software
incrementali

Ogni successiva
versione è
un'estensione
della precedente

Misto tra
tradizionale e
iterativa



METODOLOGIE AGILI |

METODOLOGIE AGILI

Per ogni ciclo
dell'iterazione, viene
definito un elenco di
compiti prioritari

RILASCIO DELL'APPLICAZIONE

Alpha

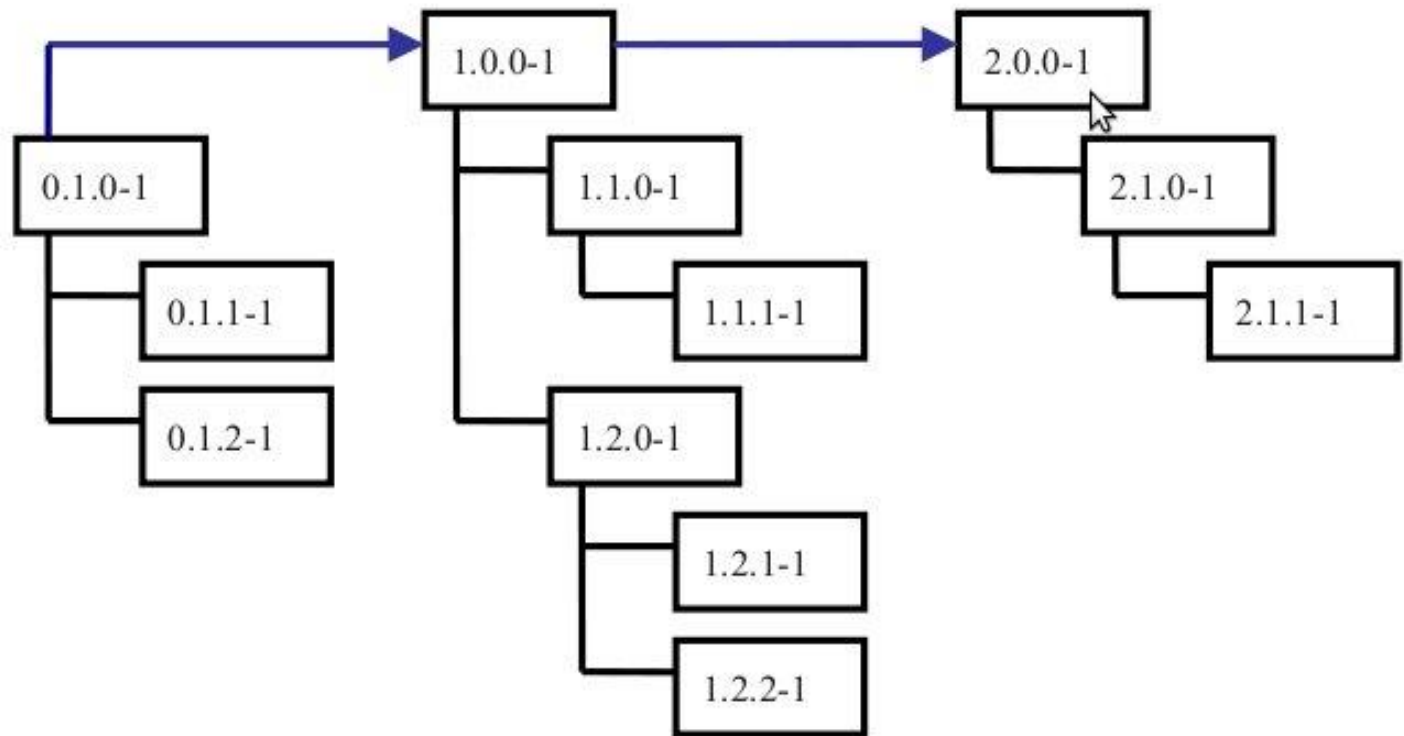
Beta

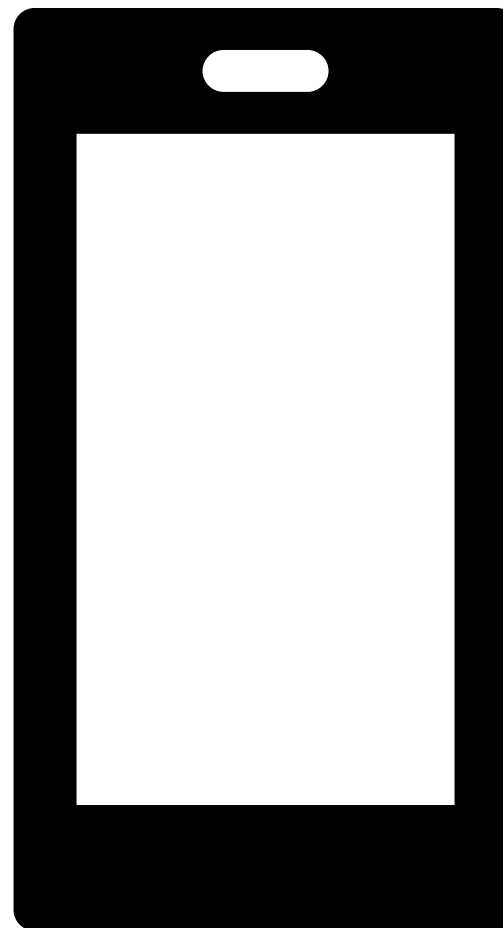
Release
candidate

Release to
manufacturing

Service pack

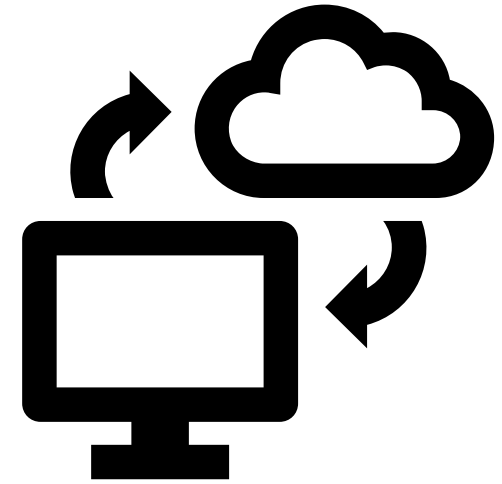
NUMERAZIONE DELLE VERSIONI



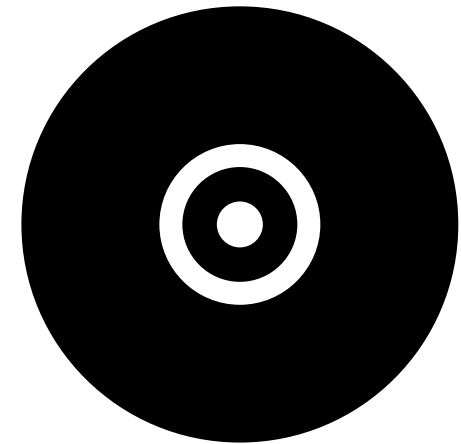


DISTRIBUZIONE APPLICAZIONE

Online



Offline





TECNICHE DI LAVORAZIONE

ANALISI E PROGETTAZIONE

Algoritmo

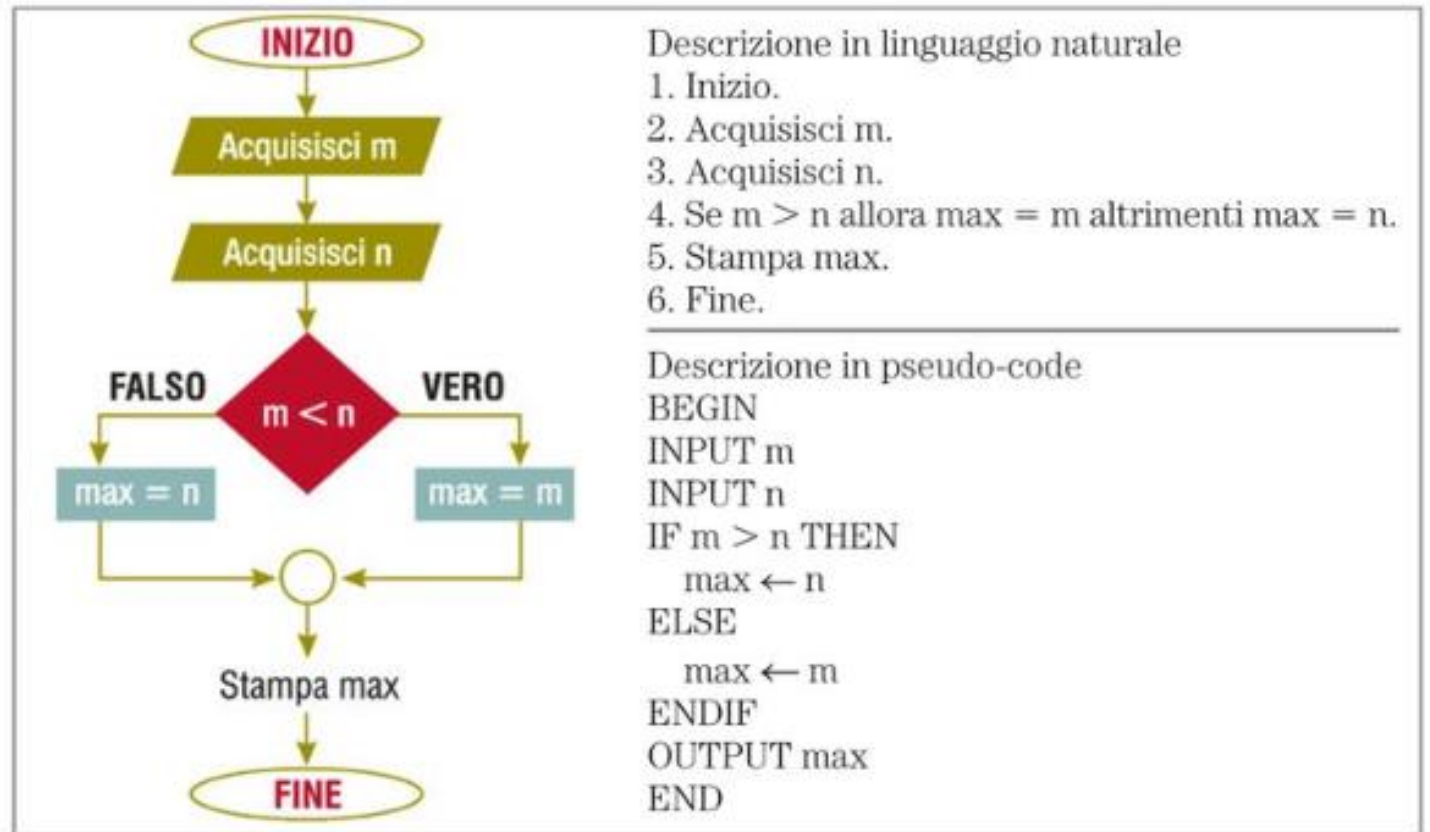
Casi di test

Piani di
prova

Debugging

Analisi in
tempo reale

ALGORITMO



CASI DI TEST

**Verifica la
bontà del
progetto**

**Tabella con
previsioni di
esecuzione**

Programma per il calcolo del massimo tra due numeri	Valori in ingresso	Risultato	Test superato	Commenti
Test 1: primo numero maggiore del secondo	m = 10 n = 3	max = 10	Sì	Nessuno
Test 2: secondo numero maggiore del primo	m = 4 n = 8	max = 8	Sì	Nessuno
Test 3: due numeri uguali	m = 5 n = 5	I numeri sono uguali	No	Viene identificato come maggiore il secondo

CASI DI TEST

PIANI DI PROVA

Tabella con
variabili
come righe

Esecuzione
passo-passo
algoritmo

Modifica dei
valori

Algoritmo per la somma dei primi 5 numeri	Valore variabile	Commento
Sum	$0 = 1 = 3 = 6 = 10 = 15$	
N	$0 = 1 = 2 = 3 = 4 = 5$	

PIANI DI PROVA

DEBUGGING

Sospende il flusso
di esecuzione
tramite
breakpoint

Permette l'analisi
in tempo reale
delle variabili



That's all Folks!