

**SUPSI**

# Problem Solving

Loris Grossi, Fabio Landoni, Andrea Baldassari

Contenuto realizzato in collaborazione con: T. Leidi, A.E. Rizzoli, S. Pedrazzini

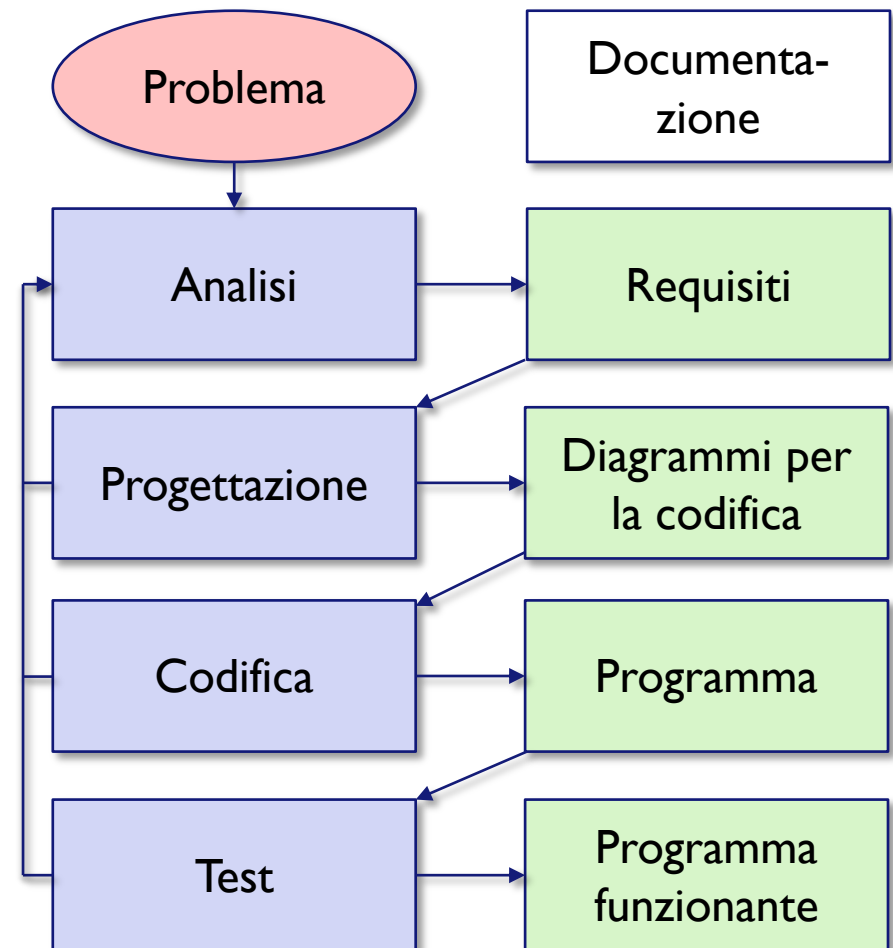
Fondamenti di Informatica

Bachelor in Ingegneria Informatica

# Metodologia di risoluzione di un problema

Come approcciare la risoluzione di un problema in informatica?

1. Formulazione del problema.
2. Analisi del problema.
3. Progettazione.
4. Implementazione.
5. Test.
6. Documentazione.





## Problem solving: formulazione del problema

Prima di affrontare la soluzione occorre **capire esattamente il problema**. Non serve a nulla risolvere il problema sbagliato.

Il problema deve essere precisato in modo che sia totalmente definito.

Devono essere eliminati gli aspetti poco rilevanti.

Spesso, in pratica, è più difficile capire esattamente la natura del problema che trovarne una soluzione.



## Problem solving: analisi del problema

L'analisi del problema ha lo scopo di **chiarire, dettagliare e documentare** le funzioni, i servizi e le **prestazioni** che devono essere offerti da un sistema software o programma, al fine di risolvere un dato problema nel contesto in cui esso dovrà operare.

Si devono identificare:

- I dati in ingresso (input).
- I risultati desiderati (output).
- Le tecniche da utilizzare per ottenere i risultati richiesti.

Le **informazioni raccolte** nella fase di analisi (specifica dei requisiti) rappresentano il **punto di partenza per la progettazione** di un prodotto software e per l'intero processo della sua realizzazione, validazione e manutenzione.



## Problem solving: progettazione

Sulla base della specifica dei requisiti prodotta dall'analisi, la fase di progettazione **definisce come tali requisiti saranno soddisfatti**, entrando nel merito della **struttura** che dovrà essere data al **sistema software** da realizzare.

Si progettano le strutture dati con cui lavorerà il programma e gli algoritmi risolutori ricercando formule e relazioni matematiche. Gli **algoritmi vengono descritti mediante schemi, diagrammi di flusso e/o pseudocodice**.

Procedimenti complicati di calcolo vengono trasformati in una successione di operazioni elementari (**divide et impera**).

Questa fase permette di sviluppare un documento nel quale si descrive la struttura di massima (architettura di alto livello) e le caratteristiche dei singoli componenti (moduli).



## Problem solving: test della progettazione

Prima di passare alla fase di codifica, la successione logica delle operazioni da seguire viene controllata sulla carta.

Vengono inoltre controllate la struttura di massima e le caratteristiche dei singoli moduli.



## Problem solving: codifica o implementazione

L'implementazione, detta anche sviluppo o codifica del prodotto software, è la **fase di realizzazione del software** che **concretizza la soluzione software attraverso la programmazione**, ovvero la stesura di programmi.

Gli schemi, i diagrammi di flusso e/o gli pseudocodici realizzati durante la fase di progettazione vengono **tradotti utilizzando un linguaggio di programmazione**.

Si genera così un **codice eseguibile** con un elaboratore elettronico.

Il linguaggio di programmazione è lo strumento utilizzato nella fase di implementazione.



## Problem solving: test dell'implementazione

Questo procedimento è utilizzato per **individuare le carenze di correttezza, completezza e affidabilità delle componenti software** in corso di sviluppo.

Consiste nell'eseguire il software da collaudare, da solo o in combinazione ad altro software di servizio, e nel **valutare se il comportamento del software rispetta tutti i requisiti** identificati durante la fase di analisi.

Si verifica il buon funzionamento del programma prestando maggiore attenzione ai casi particolari in modo da scoprire eventuali bugs.





## Problem solving: documentazione

Ogni fase prevista deve essere documentata.

Si prepara la documentazione per l'uso del prodotto e la descrizione della metodologia utilizzata.

Ci sono vari tipi di documentazione:

- Documentazione di sviluppo.
- Documentazione per il committente.
- Istruzioni per l'uso.

## Il concetto di algoritmo

Ogni problema, che deve essere affrontato mediante un calcolatore elettronico, deve essere espresso in modo logico e preciso.

Il concetto di **algoritmo** (dal nome di un matematico arabo Al-Khwārizmī) permette di formalizzare i passi da seguire per risolvere un problema.

L'algoritmo consiste in una **sequenza ordinata d'istruzioni** che definiscono delle azioni precise da compiere sui dati del problema. L'algoritmo è un procedimento che **risolve un determinato problema attraverso un numero finito di passi elementari**.

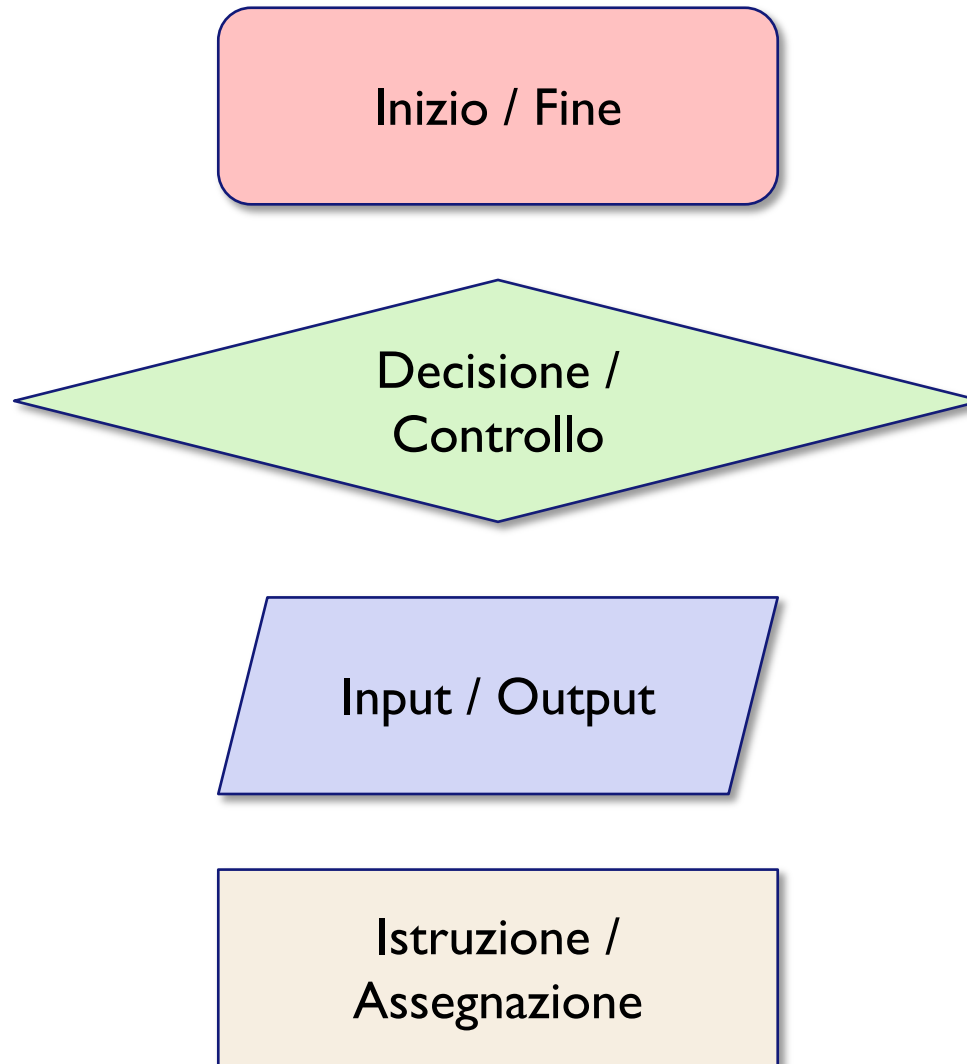
Le istruzioni, una volta eseguite fedelmente, portano, un passo dopo l'altro, alla risoluzione di un dato problema.

## Diagramma di flusso

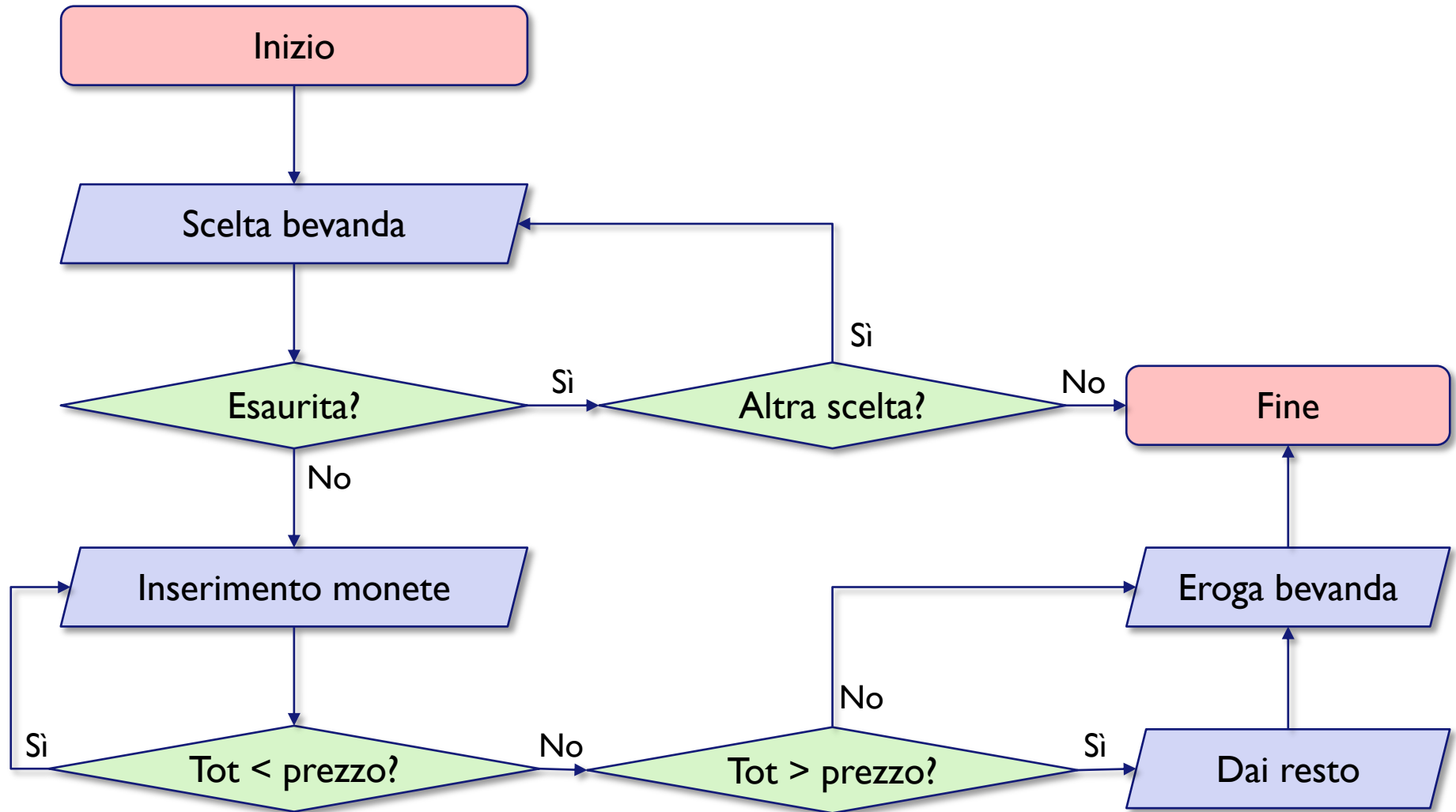
I diagrammi di flusso (flow charts) mettono a disposizione un insieme di simboli che aiutano nella descrizione grafica del flusso di controllo e di istruzioni di un algoritmo.

I diagrammi di flusso possono essere sviluppati a diversi livelli di dettaglio.

## Diagramma di flusso: simboli utilizzati



## Esempio: distributore automatico di bevande



## Riepilogo

- Problem solving
- Algoritmo
- Diagramma di flusso