



PEGASO
Università Telematica



Indice

1.	CONCETTI GENERALI	3
2.	IL PROBLEM SOLVING	4
3.	TECNICHE DI PROBLEM SOLVING	5

1. Concetti generali

La vita di tutti i giorni è piena di problemi, e decisioni, ma spesso le persone non sanno come gestirli. Molto di quello che pensiamo di un comportamento problematico può essere visto come una conseguenza di un comportamento inefficiente. L'individuo non è capace di risolvere determinati dilemmi della sua vita. I problemi non sono gestibili quando essi sono concepiti con termini generali:

- «tutto va per il peggio»;
- «egli non cambierà mai»;
- «non c'è speranza»;
- «mi sembra di avere il mondo sulle spalle».

Nel momento in cui ci scrolliamo di dosso questi modi di pensare possiamo risolvere parecchi problemi. Il problem solving è definito come un approccio sistematico di definizione del problema (questione o situazione che presenta incertezza, perplessità o difficoltà) e crea un buon numero di soluzioni possibili senza giudicare tali soluzioni. Mentre Il decision making rappresenta l'azione di riduzione delle possibilità, di scelta di una procedura e determinazione delle potenziali conseguenze delle azioni.

Il problem solving è processo cognitivo finalizzato al raggiungimento di un obiettivo in cui non c'è un metodo ovvio di soluzione del problema. È possibile insegnare tecniche di problem solving?

2. Il problem solving

Prevede diverse fasi:

- Problem finding -> rendersi conto del disagio
- Problem setting -> definire il problema
- Problem analysis -> scomporre il problema principale in problemi secondari (WBS)
- Problem solving -> eliminare le cause e rispondere alle domande poste dal problema
- Decision making -> decidere come agire in base alle risposte ottenute
- Decision taking -> passare all'azione

Problem setting: Diagnosi della situazione e definizione del problema. Metodo teorico e pratico che serve a trasformare un disagio in un problema, in una questione ben definita. Precede il problem solving, che trasforma il problema ben definito in un progetto, da gestire secondo le tecniche del project management.

Problem solving: gestione della situazione e risoluzione del problema. Una volta definito con chiarezza il problema in tutti i suoi aspetti, si può passare alla fase del problem solving. Il problem solving riesce se è preceduto da un corretto problem setting.

3. Tecniche di problem solving

- Programmare il problem solving in sei mosse
 - Definizione del problema
 - Analisi del problema
 - Design dell'algoritmo
 - Approccio top-down
 - Pseudo codice
 - Flowchart
 - Codifica del programma
 - Testing e debugging del programma
 - Documentazione del programma

Definizione del problema: descrizione del problema in maniera chiara e precisa. Si deve conoscere esattamente cosa fare.

Analisi del problema: comprendere completamente i dati in input richiesti e l'output prodotto dal programma. Quali sono i dati di input? Quali sono i loro tipi di dati? Quali sono i dati di output? Quali sono i formati di output?

Design dell'algoritmo: l'algoritmo è la definizione formale di un metodo per risolvere il problema.

Affinamento progressivo: suddividere il problema in sotto problemi. Ogni sotto problema è ulteriormente suddiviso in sotto problemi più piccoli. Il processo è ripetuto finchè ogni sotto problema è facile da risolvere. Il processo è conosciuto come «Stepwise refinement».

Top-down design: Se la scomposizione inizia al livello più alto e poi procede verso il basso la tecnica è detta **top-down**. Il task identificato ad ogni livello della suddivisione è detto **modulo**.

Pseudocodice: Sequenza di istruzioni English-like che rappresentano un algoritmo. L'algoritmo in questa fase viene scritto utilizzando un linguaggio simile ad un linguaggio di programmazione.

Flowchart: utilizzati per illustrare la struttura logica. Sono rappresentazioni grafiche di sequenze, selezioni e cicli.

Codifica del programma: scrittura del programma in un linguaggio di programmazione: **coding**. Scrittura della struttura del programma in moduli. Un buon programma deve essere facile da leggere e deve utilizzare un approccio modulare.

Testing e debugging: debugging è la fase con la quale si testa il programma e lo si libera dai bug. Esistono tre tipi di errori di programmazione. Sintattici: regole grammatiche del linguaggio di programmazione. Run-time: errori che si verificano quando il programma è in esecuzione (divisione by zero). Logici: algoritmo incorretto oppure implementazione incorretta di un algoritmo che produce risultati inaspettati.

Documentazione del programma: contiene una descrizione scritta del programma. È utilizzata per rendere il programma comprensibile anche ad altri.