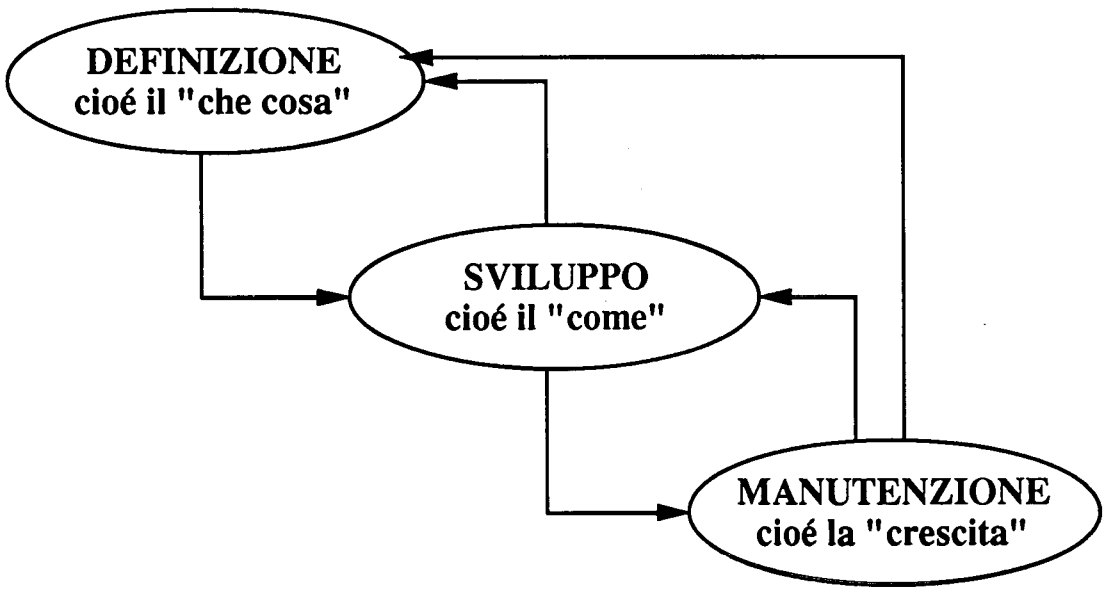
LE FASI DELLA PRODUZIONE DEL SW

Indipendentemente dal paradigma prescelto, il processo di sviluppo si articola in tre fasi generali, quali che siano le dimensioni del progetto, l'area applicativa, la complessità:



1) La DEFINIZIONE (il "che cosa")

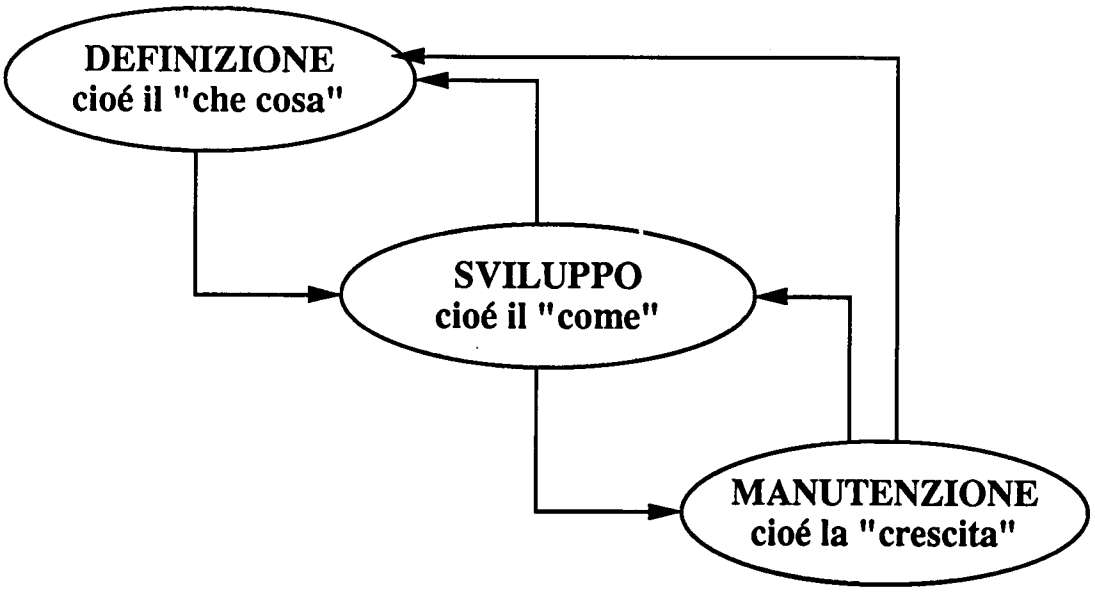
a) analisi del sistema (ruolo del software nell'intero sistema informativo)

b) pianificazione del progetto (risorse, costi, piani di lavoro)

c) analisi dei requisiti (definizione dettagliata della funzione del software e della informazione da gestire, interfacce, ...)

LE FASI DELLA PRODUZIONE DEL SW

(continuaz.)



2) Lo SVILUPPO (il "come")

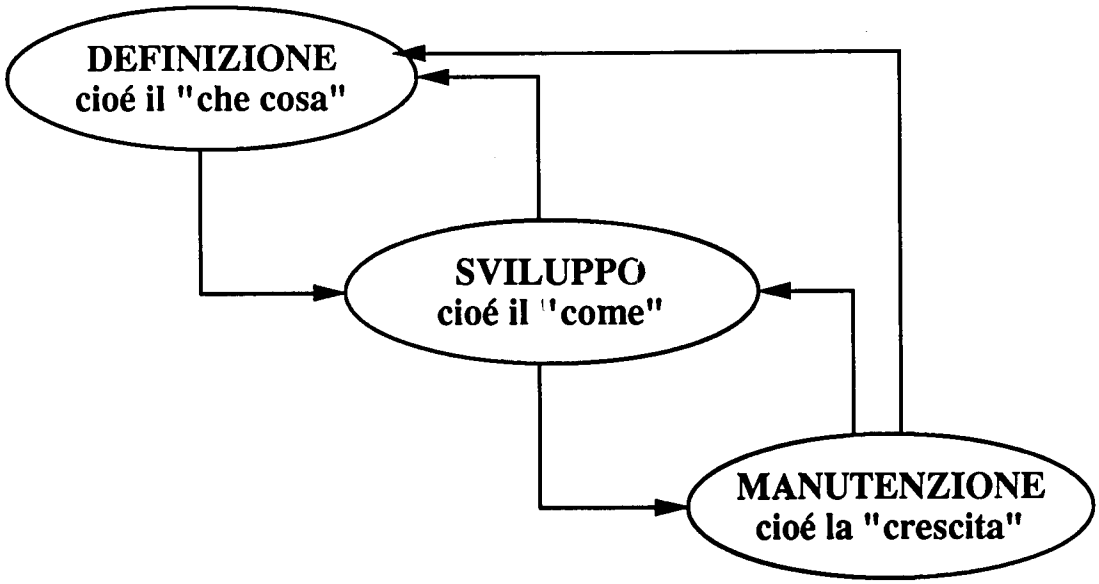
a) progettazione, cioé traduzione dei requisiti in rappresentazioni formali (grafici, tabelle, testi, ...) atte a descrivere l'architettura, le strutture-dati, gli algoritmi, ...

b) codifica cioé traduzione delle rappresentazioni formali in linguaggio procedurale o non e quindi in moduli eseguibili

c) testing cioé definizione di procedure di verifica per l'individuazione di errori logici o di codice)

LE FASI DELLA PRODUZIONE DEL SW

(continuaz.)



3) La MANUTENZIONE (la "crescita")

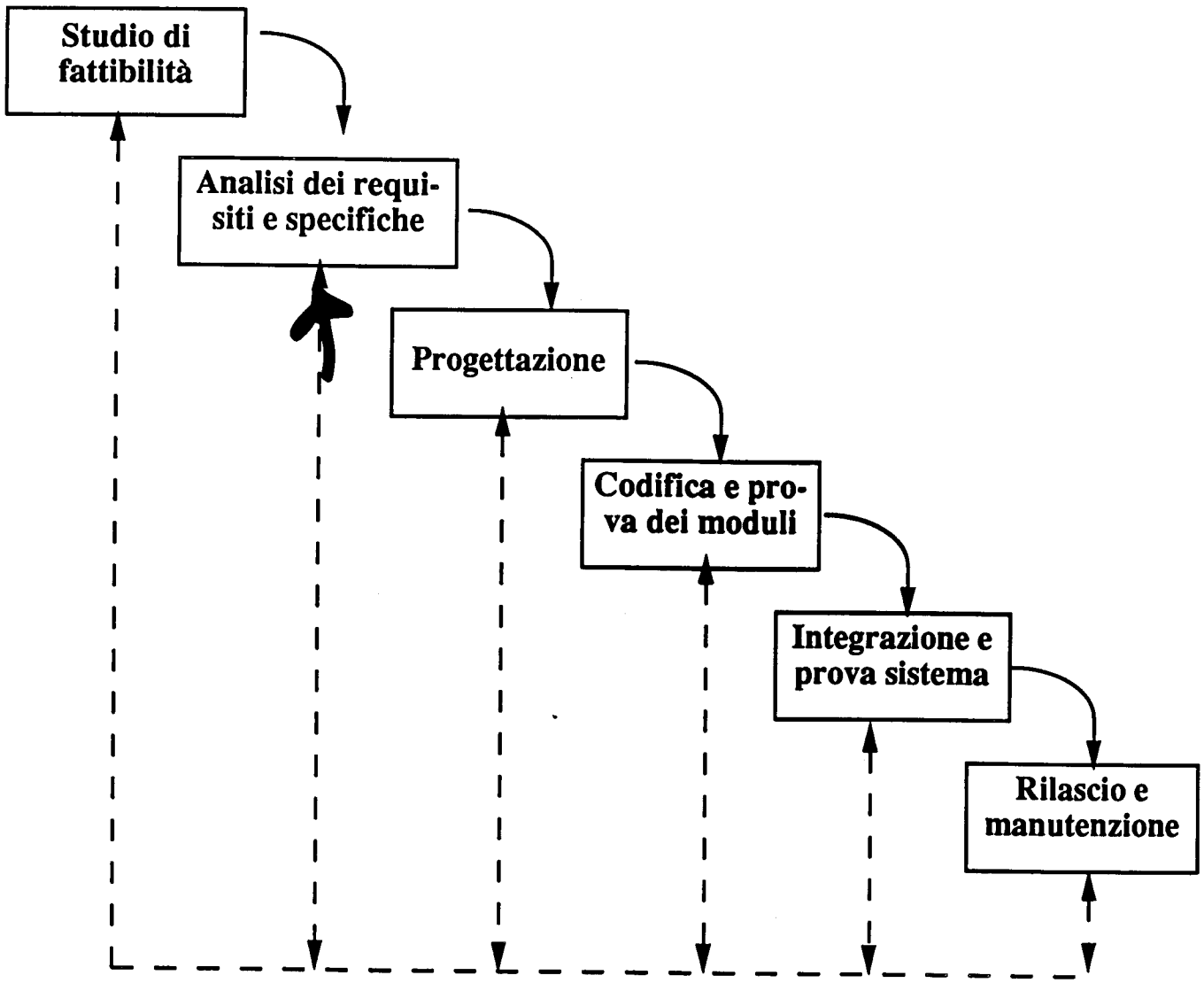
a) correzione degli errori (anche nel tempo)

b) adattamenti per modifiche dell'hardware

c) miglioramenti nel tempo (nuove funzioni, nuovi requisiti, ...)

I PARADIGMI DELL'INGEGNERIA DEL SW

1) Il ciclo di vita



PROBLEMI:

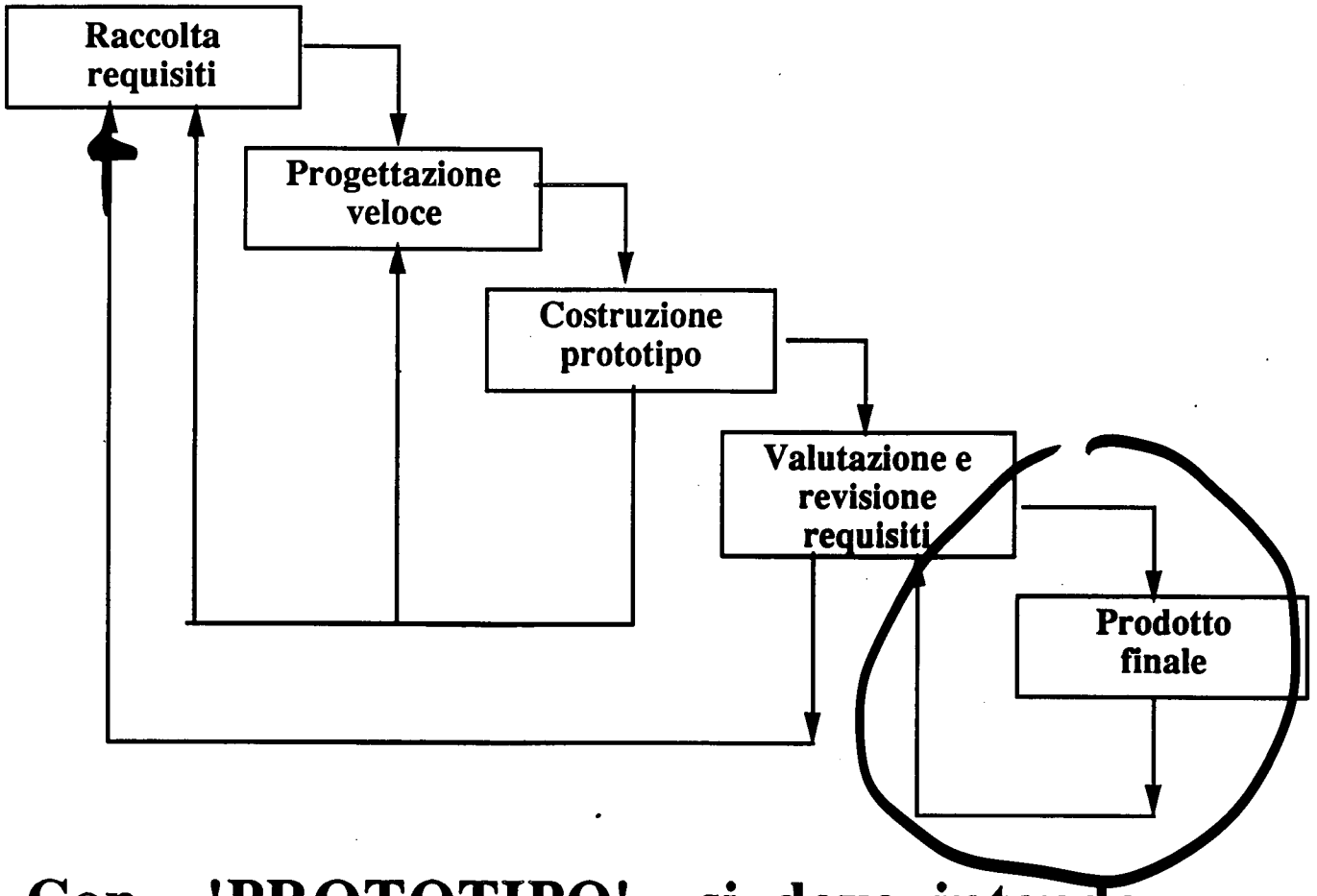
1) Il processo reale non é mai sequenziale

2) All'inizio la definizione data dal cliente non é mai completa

3) Una prima versione si può avere solo alla fine, dopo molto tempo e gli errori scoperti a questo punto sono un guaio.

I PARADIGMI DELL'INGEGNERIA DEL SW

2) La prototipizzazione

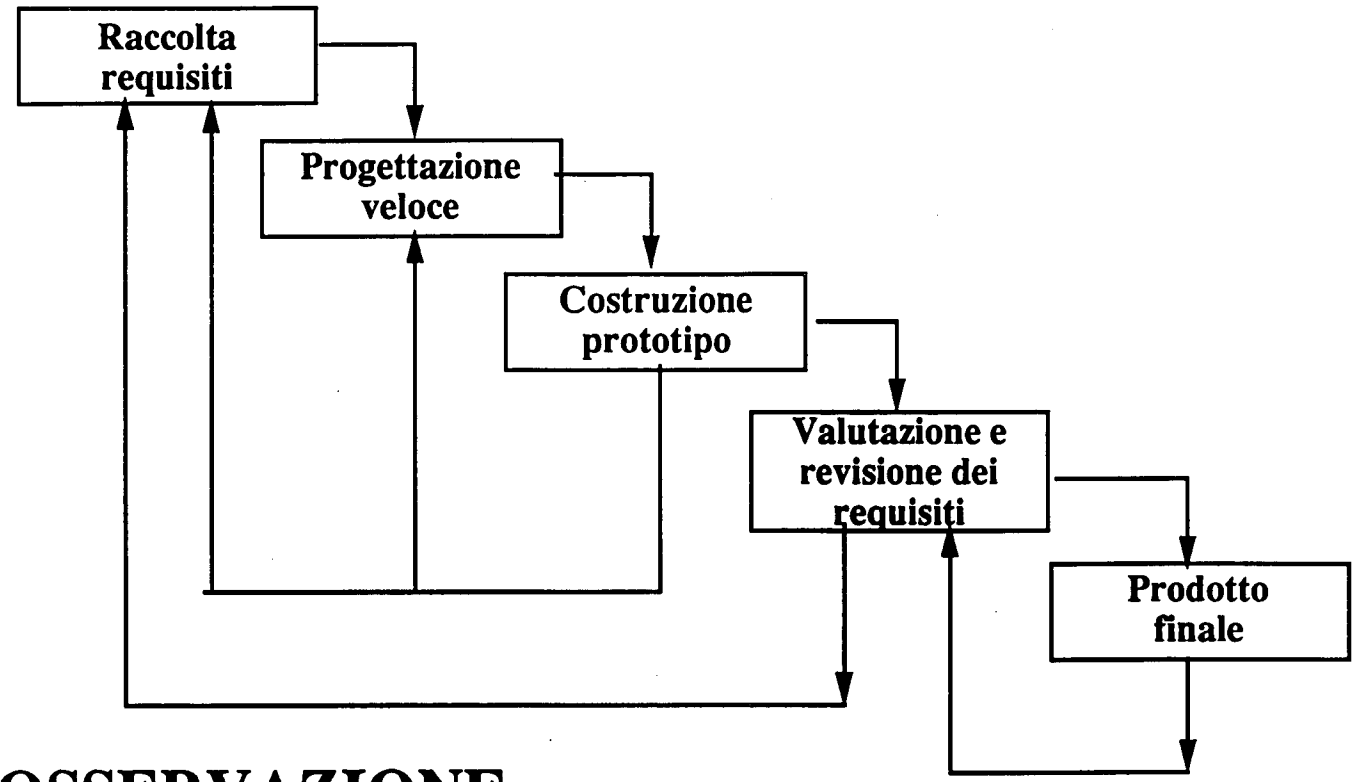


Con 'PROTOTIPO' si deve intendere un modello del software in una delle seguenti forme:

1) un testo che faccia capire all'utente in modo puntuale l'interazione uomo-macchina in ogni possibile situazione
2) un software funzionante che realizzi parte delle funzioni richieste in modo da costituire un esempio significativo
3) un software già esistente che svolga funzioni analoghe e possa essere considerato un punto di partenza per individuare e definire nuove funzioni e miglioramenti

I PARADIGMI DELL'INGEGNERIA DEL SW

2) La prototipizzazione *(continuaz.)*



OSSERVAZIONE:

- **Il prototipo software certamente non sarà riutilizzabile (al più qualche modulo) per la costruzione del prodotto finale**

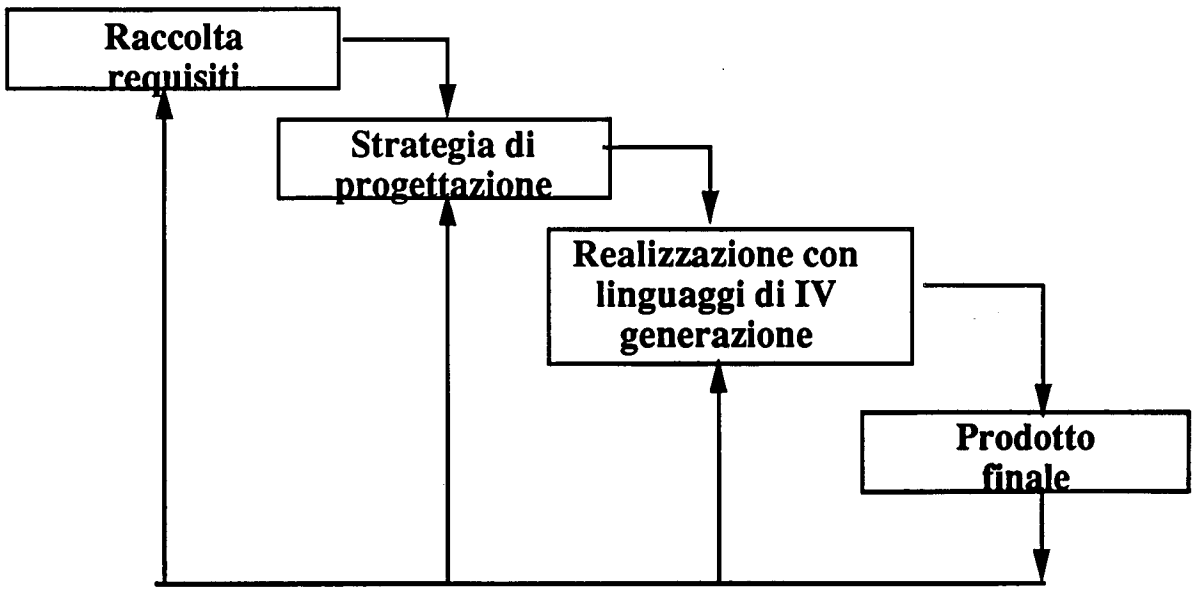
RISCHI:

1) **Il cliente "si affeziona" al prototipo, non lo vuol vedere gettato via e chiede solo dei ritocchi. Se il progettista acconsente, viene fuori un prodotto scadente**

2) **Il progettista fa delle scelte "comode" per accelerare la produzione del prototipo (linguaggio più comodo, algoritmo poco efficiente, ...) e alla fine tali scelte finiscono per essere definitive o condizionanti.**

I PARADIGMI DELL'INGEGNERIA DEL SW

3) Le tecniche di quarta generazione



Con 'LINGUAGGI DI IV GENERAZIONE' si deve intendere un insieme di strumenti software atti a

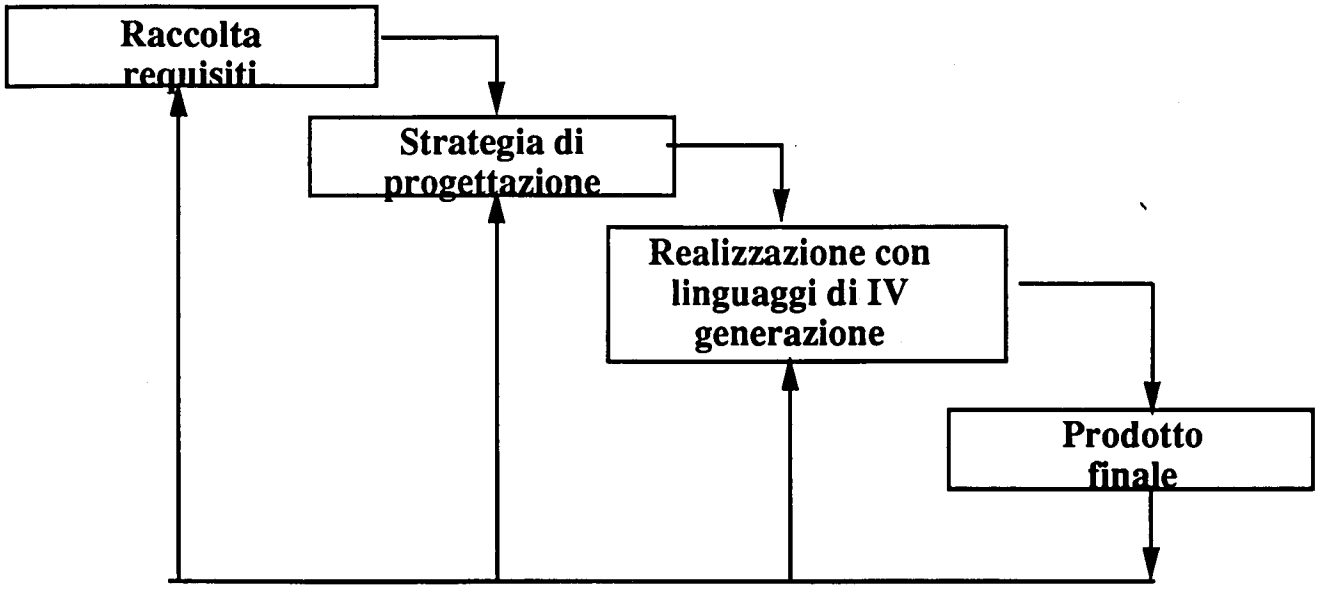
- **specificare le caratteristiche di alto livello con un linguaggio vicino a quello naturale**
- **tradurre automaticamente le specifiche in codice eseguibile**

Un ambiente di IV generazione generalmente comprende:

1) **generatore di interfaccia interattiva con capacità grafiche**
2) **un linguaggio per l'interrogazione di data-base**
3) **un linguaggio per il trattamento dei testi**
4) **un generatore di codice**

I PARADIGMI DELL'INGEGNERIA DEL SW

3) Le tecniche di quarta generazione *(contin.)*



PROBLEMI:

1) Gli ambienti sono ancora non molto sofisticati e sono al più adatti a piccole applicazioni

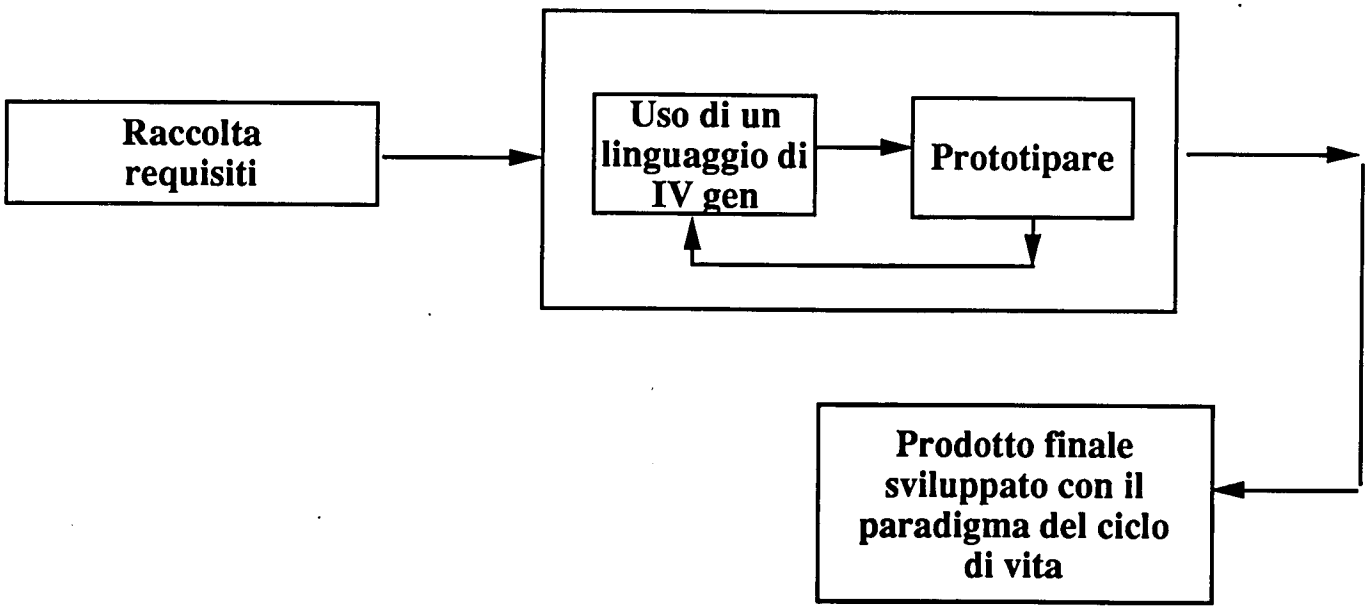
2) La formalizzazione del dialogo cliente-sviluppatore e la conseguente specifica del SW é ancora un punto debole e si continua ad usare l'approccio degli altri paradigmi

3) Il codice prodotto non sempre é efficiente e vi sono problemi per la manutenzione di grandi applicazioni

4) Per grandi applicazioni, attualmente, vengono introdotte delle rigidità di progetto che vanificano il vantaggio della produzione automatica del codice

I PARADIGMI DELL'INGEGNERIA DEL SW

4) Un paradigma misto



OSSERVAZIONI:

- **non é il caso di essere dogmatici nel dire che un paradigma é meglio di un altro.**

- **in genere il paradigma del ciclo di vita é più adatto al caso in cui i requisiti (funzioni e prestazioni) siano sufficientemente chiari.**

- **la prototipazione, invece, é più opportuna quando si cercano elementi chiarificatori (pesante interazione uomo-macchina, interfacce amichevoli, ...).**

- **nel caso di necessità di protipazione, poi, l'uso di un linguaggio di IV generazione può accelerare la costruzione dei prototipi.**

E' quindi chiaro che l'uso di un qualche paradigma misto può spesso essere la scelta migliore.