

## Progettazione – parte I

Leggere sez. 4.1  
Ghezzi et al



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

## Progettazione

- Attività preliminare alla realizzazione di qualsiasi artefatto complesso.
- Scopi:
  - scomporre l'artefatto in parti
  - assegnare a ognuna una funzione
  - definire le loro interazioni
  - verificare che l'artefatto complessivo sia funzionale agli obiettivi

Progettazione 1



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

2

## Progettazione: due accezioni

- Attività che fa da ponte fra i requisiti e implementazione del software (dal “cosa” al “come”)
- Definizione della struttura di tutti gli artefatti coinvolti nella produzione di software
  - Es.: il documento dei requisiti richiede una struttura che lo renda facile da capire e modificare

Progettazione 1

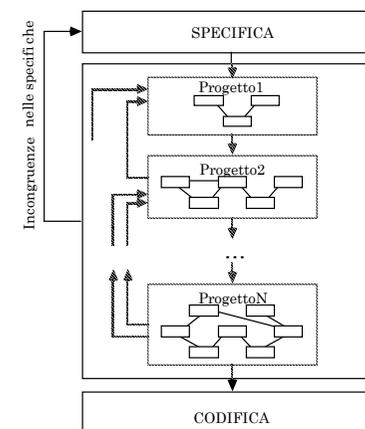


università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

3

## Progettazione del software

- Processo iterativo
- Porta a scoprire incongruenze delle specifiche
- Non c'è un livello di dettaglio definito prima di passare alla codifica
- Si inizia con un'architettura.



Progettazione 1



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

4

## Architettura software

- Mostra la struttura globale e l'organizzazione del sistema
- Definisce
  - i principali componenti del sistema
  - relazioni fra componenti
  - motivazioni della scomposizione scelta
  - vincoli sul progetto dei componenti
- Guida la successiva progettazione

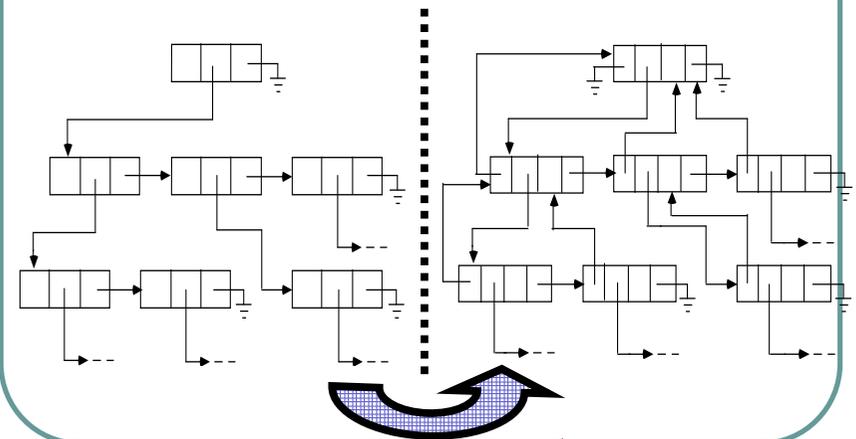
## Due obiettivi dell'attività di progettazione

- Progettazione in vista del cambiamento
  - Prevedere i cambiamenti possibili
  - Soddisfare i requisiti attuali, ma in modo adattabile a nuovi requisiti
- Famiglie di prodotti
  - Progettare insieme diverse versioni di un prodotto permette di risparmiare sulla produzione delle parti comuni

## Possibili cambiamenti

- Algoritmi
  - Possono essere necessari algoritmi diversi a seconda dell'applicazione (compromesso spazio/tempo, hardware disponibile, etc.)
- Rappresentazione dei dati
  - I cambiamenti hanno una ricaduta notevole sui costi di manutenzione
- Progettare in modo da astrarre dai dettagli

## Esempio



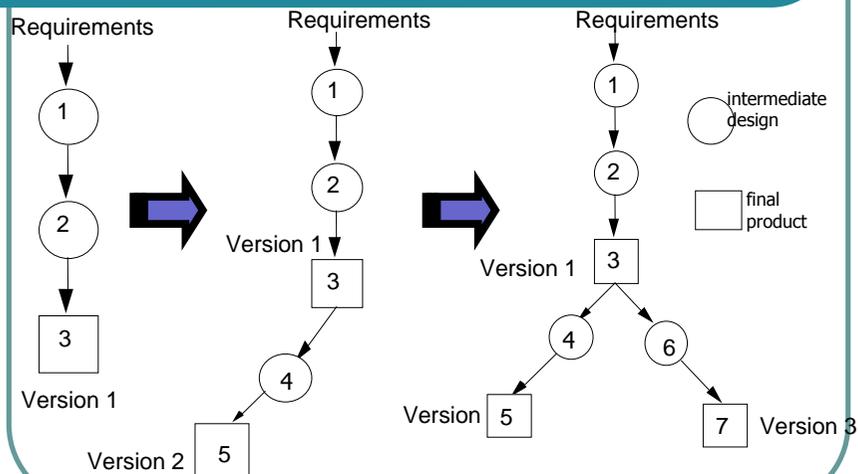
## Possibili cambiamenti

- Macchina astratta
  - Sistema operativo
  - Semantica del linguaggio di programmazione
  - DBMS
- Ambiente sociale (leggi, moneta, etc.)
- Cambiamenti dovuti al processo di sviluppo

## Famiglie di prodotti

- Versioni diverse dello stesso sistema
- Alcuni requisiti sono comuni, altri variano a seconda della versione
  - Es. famiglia di telefoni cellulari
    - Possono differire per standard di rete, lingue supportate, disponibilità di fotocamera, etc.
  - Es. sistema di prenotazioni
    - Hotel e università possono differire per risorse, intervalli, etc.

## Progettazione sequenziale



## Progettazione sequenziale

- Realizzare il primo membro e modificarlo successivamente per ottenere gli altri
- Non soddisfacente:
  - Il progetto della famiglia è condizionato da scelte fatte per il primo membro
  - I progetti intermedi spesso non sono documentati e si procede direttamente sul codice

## Progettazione corretta

- Durante la progettazione del primo membro, identificare
  - le parti comuni
  - quelle che caratterizzano i singoli membri
- Successivamente, riprendere dalla progettazione delle parti caratteristiche
- Questo metodo presuppone la *modularizzazione*

