

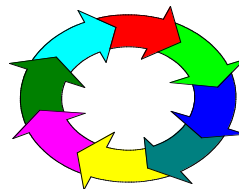
Il ciclo di vita dei sistemi informatici

1

1

Il ciclo di vita

- Comprende le attività svolte durante il periodo di esistenza di un sistema informatico
 - ⇒ definizione strategica
 - ⇒ pianificazione
 - ⇒ controllo di qualità
 - ⇒ analisi dei requisiti
 - ⇒ progettazione
 - del sistema
 - esecutiva
 - ⇒ realizzazione e collaudo in fabbrica
 - ⇒ certificazione
 - ⇒ installazione
 - ⇒ collaudo del sistema installato
 - ⇒ esercizio
 - ⇒ diagnosi e manutenzione
 - ⇒ evoluzione
 - ⇒ messa fuori servizio



2

Definizione strategica

In questa fase vengono prese decisioni strategiche su quale area aziendale deve essere automatizzata. L'obiettivo è definire una direzione generale per il progetto.

- ⇒ Vengono prese decisioni sull'area aziendale che deve essere oggetto di automazione

Pianificazione

Si identificano obiettivi, fabbisogni e si conduce uno studio di fattibilità. Vengono calcolati i costi, i benefici e i tempi per valutare diverse strategie di attuazione.

- ⇒ Vengono definiti gli obiettivi, evidenziati i fabbisogni e viene condotto uno studio di fattibilità per individuare possibili strategie di attuazione e avere una prima idea dei costi, dei benefici e dei tempi.

Controllo di qualità

Si predispose un piano di controllo qualità per assicurarsi che il sistema sia conforme alle specifiche e funzioni come previsto.

- ⇒ Viene predisposto un piano di controllo di qualità per il progetto, allo scopo di garantire il rispetto delle specifiche e di controllare che il sistema realizzato si comporti come previsto

Analisi dei requisiti

I requisiti del sistema vengono formalizzati con tecniche di modellazione. Si producono macro-specifiche che guideranno le fasi successive.

- ⇒ Formalizza i requisiti avvalendosi di tecniche di modellazione della realtà e produce macro-specifiche per la fase di progettazione

Progettazione del sistema

I requisiti sono tradotti in una soluzione architeturale di alto livello, indipendente dai particolari strumenti tecnologici. Si producono specifiche che descrivono come il sistema sarà costruito, senza entrare nei dettagli implementativi.

- ⇒ Interpreta i requisiti in una soluzione architeturale di massima. Produce specifiche indipendenti dai particolari strumenti che saranno usati per la costruzione del sistema

Progettazione esecutiva

La progettazione si concentra sui dettagli della struttura e del comportamento dei componenti del sistema. Si utilizzano strumenti di sviluppo per creare un prodotto funzionante.

- ⇒ Vengono descritti struttura e comportamento dei componenti dell'architettura, producendo specifiche che possano dar luogo, attraverso il ricorso a strumenti di sviluppo opportuni, a un prodotto funzionante

Realizzazione e collaudo in fabbrica

Il sistema viene implementato sulla piattaforma scelta e testato con casi di prova definiti in precedenza (test alpha).

- ⇒ Il sistema viene implementato sulla piattaforma prescelta e viene testato internamente (α -test) sulla base dei casi prova definiti durante la fase di analisi

Stabilire che cosa il sistema farà.

Stabilire come ogni componente sarà implementato e integrato.

3

Trasformare le specifiche tecniche, definite nella progettazione esecutiva, in codice sorgente, configurazioni, e componenti funzionanti.

- Errori bloccanti: malfunzionamenti che impediscono il proseguimento del collaudo.
- Errori non bloccanti: malfunzionamenti che, pur presenti, non ostacolano il collaudo.
- Problemi di operatività: funzionalità che non operano come previsto.
- Problemi funzionali: funzionalità richieste ma non implementate.

Certificazione

L'obiettivo è assicurare che il prodotto soddisfi gli standard di qualità e i requisiti definiti.

- ⇒ L'attività di certificazione del software ha lo scopo di verificare che esso sia stato sviluppato secondo i criteri previsti dal metodo tecnico di progetto, in conformità alle specifiche di sistema e a tutta la documentazione di progetto

Installazione

Questa fase può includere il recupero e la migrazione di dati preesistenti, garantendo la continuità operativa.

- ⇒ Il sistema viene installato e configurato, e vengono recuperati gli eventuali dati pregressi

Collaudo del sistema installato

per identificare eventuali problemi

- ⇒ Gli utenti testano "in vitro" il prodotto installato (β -test). Si possono evidenziare errori bloccanti (malfunzionamenti che pregiudicano l'attività di collaudo), errori non bloccanti (malfunzionamenti che non pregiudicano l'attività di collaudo), problemi di operatività (una funzionalità richiesta non viene attuata adeguatamente) e funzionali (una funzionalità richiesta non è implementata)

Esercizio

- ⇒ Quando il collaudo dà esito positivo il sistema viene avviato ("messo in produzione"), inizialmente affiancando e poi sostituendo gradualmente l'eventuale sistema preesistente

Diagnosi

- ⇒ Durante l'esercizio gli utenti rilevano eventuali errori

Manutenzione

- ⇒ Gli errori che si manifestano durante il funzionamento vengono segnalati e corretti (manutenzione correttiva). Può inoltre essere necessario intervenire sul software per adattarlo ai cambiamenti del dominio applicativo (manutenzione adattativa)

Evoluzione

- ⇒ Si valutano le possibilità di far evolvere il sistema incorporando nuove funzionalità o migliorandone l'operatività (manutenzione evolutiva o perfettiva)

4

2

Analisi dei requisiti

Formalizza Requisiti e Crea Macro-Specifiche per la fase di Progettazione

Obiettivo: Produrre un documento di specifica dei requisiti, che:

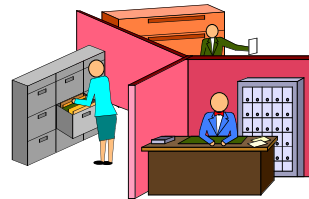
- Descriva cosa il sistema deve fare.
- Serva come riferimento condiviso tra clienti, utenti, analisti e sviluppatori.
- Eviti ambiguità o incomprensioni che potrebbero causare errori nelle fasi successive.

Scopo dell'analisi è produrre, utilizzando tecniche di modellazione della realtà, un documento di **specifica dei requisiti** che diventi l'input per le successive fasi di progettazione e realizzazione

Oggetto dell'analisi è l'organizzazione ^(cioè l'azienda) nel suo complesso:

per comprendere il contesto e identificare le necessità

- ⇒ sottosistemi aziendali Ogni sottosistema ha requisiti operativi particolari da tenere in considerazione.
- ⇒ risorse
- ⇒ processi Ossia le attività eseguite per raggiungere determinati obiettivi.
- ⇒ flussi informativi Analisi del movimento di informazioni all'interno dell'organizzazione.
- ⇒ ...



5

Specifica dei requisiti

rappresenta il punto di incontro tra il cliente (utente finale) e il fornitore (team di sviluppo). L'obiettivo è creare un documento che descriva con precisione le funzionalità richieste dal software e che diventi il riferimento ufficiale per tutte le fasi successive.

- La specifica dei requisiti è un ^{formale} accordo tra il produttore di un servizio e il suo consumatore
- In questa fase del ciclo di vita, attraverso la specifica dei requisiti, l'utente finale e il progettista si accordano sulle funzionalità messe a disposizione dal software
- La difficoltà per questo tipo di ^(fase del ciclo di vita) specifica è data dalla diversità dei linguaggi usati dalle due parti

Gli utenti finali tendono a descrivere le esigenze in termini di processi aziendali o bisogni pratici. I progettisti (o sviluppatori) usano un linguaggio tecnico. La diversità nei modi di comunicare può portare ad Ambiguità.

- Qualità per la specifica dei requisiti: essa deve soddisfare alcune caratteristiche fondamentali:

Ogni requisito deve essere descritto in modo semplice e diretto, evitando tecnicismi inutili.

⇒ **Chiarezza**: ogni specifica deve indicare quanto più chiaramente possibile le operazioni e i soggetti del processo che descrive

⇒ **Non ambiguità**: il processo descritto dalla specifica deve essere definito in modo completo e dettagliato in modo tale da non poter essere interpretato in più modi.

⇒ **Consistenza**: le specifiche non devono contenere punti contraddittori

6

Importanza dei Requisiti

Stadio	Costo relativo per la correzione
Requisiti	0.1 - 0.2
Progettazione	0.5
Codifica	1
Test	2
Accettazione	5
Manutenzione	20

Più tardi viene scoperto un errore nel ciclo di sviluppo del software, maggiore è il costo di riparazione.

Analisi: alcune definizioni

- ❑ **De Marco:** Questa definizione sottolinea l'importanza di comprendere il problema in modo approfondito, senza saltare direttamente alla progettazione o alla realizzazione.

⇒ l'analisi è lo studio di un problema, *prima* di intraprendere qualche azione

- ❑ **Coad:**

⇒ l'analisi è lo studio del dominio di un problema, che porta a una specifica del comportamento *esternamente osservabile*; una descrizione *completa, coerente e fattibile* di ciò che occorre realizzare; una trattazione quantitativa delle caratteristiche operazionali (cioè *affidabilità, disponibilità, prestazioni*)

Misurazione delle caratteristiche operative come:
Affidabilità: Frequenza di errori o malfunzionamenti.
Disponibilità: Percentuale di tempo in cui il sistema è funzionante.
Prestazioni: Tempo di risposta, velocità di esecuzione.

- ❑ **Davis:** L'analisi del problema consiste nel definire i confini e le caratteristiche principali del software. L'analisi implica anche il confronto e la selezione di una soluzione adeguata

⇒ l'analisi del problema è il momento in cui viene definito lo spazio del prodotto; la descrizione del prodotto comporta la scelta di una soluzione e l'esplicitazione del comportamento esterno del prodotto dimostrando che *esso soddisfa i requisiti*

(dominio)

Cosa e come modellare

come è organizzata una determinata entità (un'azienda, un'istituzione, un progetto, ecc.).

- Il processo di analisi è **incrementale** e porta per passi successivi alla stesura di un insieme di documenti in grado di **rappresentare un modello dell'organizzazione** e comunicare, in modo **non ambiguo**, una descrizione **esauriente, coerente e realizzabile** dei vari aspetti **statici, dinamici e funzionali** di un sistema informatico

Metodi di analisi

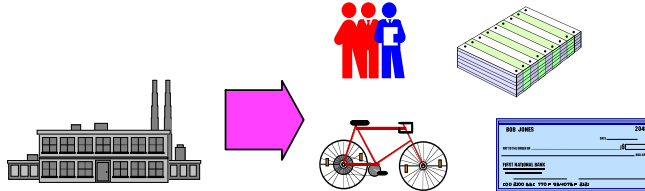
Differenti problemi richiedono differenti approcci e differenti strumenti di analisi

In generale, il metodo scelto dipende dal tipo di sistema da analizzare e dagli obiettivi specifici della progettazione. Spesso, si possono anche combinare più metodi per ottenere una visione completa del sistema.

- 1 □ Analisi orientata agli **oggetti**
- 2 □ Analisi orientata alle **funzioni**
- 3 □ Analisi orientata agli **stati**

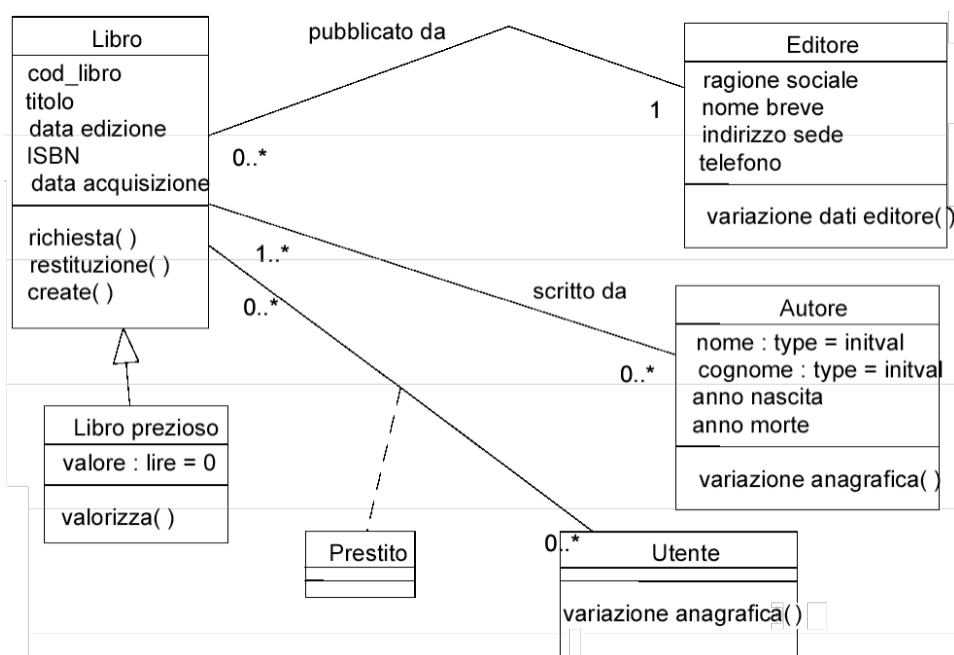
1 Analisi orientata agli oggetti

- L' enfasi è posta:
 - ⇒ sull' identificazione degli **oggetti**
 - ⇒ sulle **interrelazioni** tra oggetti



- *Nel tempo le proprietà strutturali degli oggetti osservati restano abbastanza stabili, mentre l'uso che degli oggetti si fa può mutare in modo sensibile*

Analisi orientata agli oggetti

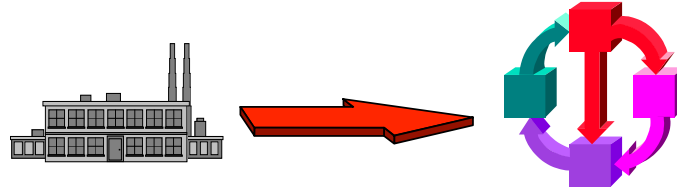


2 Analisi orientata alle funzioni

- L'obiettivo è rappresentare un sistema come:

⇒ un insieme di **flussi informativi** Un flusso informativo è un movimento di dati.

⇒ una rete di **processi** che trasformano flussi informativi Un processo è una funzione o un'attività che trasforma un input in un output.



- Ciò corrisponde alla progressiva costruzione di una gerarchia funzionale

Gerarchia Funzionale: è una rappresentazione organizzata dei vari processi del sistema, in cui i processi più generali vengono scomposti in processi più dettagliati. Man mano che si scende nella gerarchia, ogni funzione principale viene suddivisa in sottofunzioni più specifiche che rappresentano attività più dettagliate.

Analisi orientata alle funzioni

1 : Creazione_Vendita_Produzione

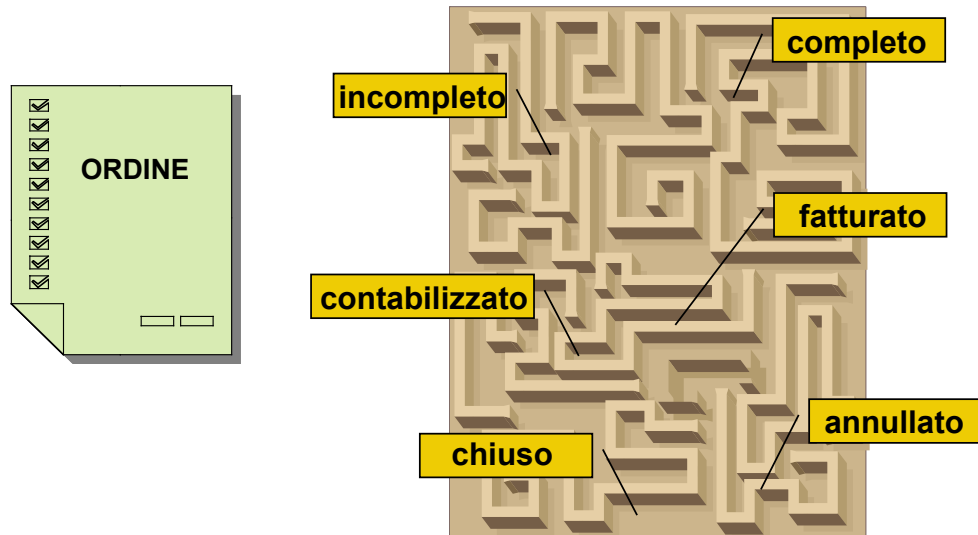
Esempio di Gerarchia Funzionale

- 201 : Progettazione campionario
 - 3011 : Indagine di mercato
 - 3012 : Creazione stilistica
 - 3013 : Creazione prototipi
 - 40131 : Creazione cartamodello
 - 40132 : Realizzazione prototipo
 - 3014 : Definizione modelli campionario
 - 40141 : Scelta prototipi
 - 501411 : Riunione prima selezione
 - 501412 : Evidenziazione difficoltà realizzative
 - 501413 : Produzione schizzo
 - 40142 : Definizione tecnica modelli
 - 40143 : Codifica modelli
 - 3015 : Produzione campionario
 - 40151 : Lancio in produzione del campionario
 - 40152 : Produzione campionario
 - 40153 : Rientro e controllo
- 202 : Pianificazione operativa
 - 3021 : Definizione obiettivi
 - 3022 : Elaborazione delle previsioni di vendita
 - 3023 : Stesura piano operativo
- 203 : Vendita
 - 3031 : Presentazione campionario
 - 3032 : Acquisizione ordini e controllo campagna vendita
 - 3023 : Revisione previsioni di vendita
- 204 : Produzione
 - 3041 : Approvvigionamento materie prime e prodotti finiti
 - 3042 : Ciclo di produzione
 - 40421 : Programmazione produzione
 - 40422 : Confezione
 - 40423 : Rientro e controllo della produzione

3

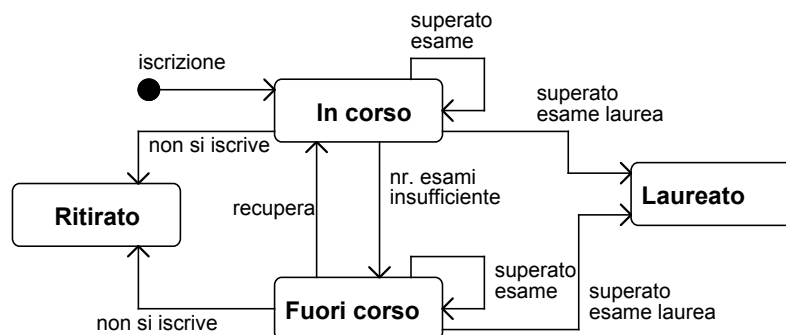
Analisi orientata agli stati

- Per alcune categorie di applicazioni, può essere utile pensare fin dall'inizio in termini di **stati** operativi, in cui si può trovare il sistema allo studio, e **transizioni di stato**



15

Analisi orientata agli stati



16

Ogni tipo di applicazione ha caratteristiche che richiedono approcci specifici, ma l'integrazione dei vari metodi consente di ottenere una visione più completa e accurata del sistema.

Uso dei metodi d'analisi

L'obiettivo è quello di scegliere il metodo o la combinazione di metodi più adatti al tipo di sistema che si sta analizzando, garantendo così una comprensione chiara e una progettazione efficace.

❑ La tendenza attuale è integrare metodi dei tre tipi, tenendo però conto della tipologia di applicazione

- ⇒ **Applicazioni orientate agli oggetti** si concentrano sulle informazioni e sui dati che il sistema gestisce.
 - l'aspetto più significativo è costituito dalle informazioni, le funzioni svolte sono relativamente semplici
 - ⇒ **Applicazioni orientate alle funzioni** si concentrano sulle operazioni che devono essere eseguite sui dati.
 - la complessità risiede nel tipo di trasformazione input-output operata
 - ⇒ **Applicazioni orientate al controllo** (stati) l'aspetto più significativo è la sincronizzazione tra attività che devono cooperare.
 - l'aspetto più significativo da modellare è la sincronizzazione fra diverse attività cooperanti nel sistema
- Tali applicazioni sono spesso sistemi concorrenti o sistemi distribuiti, dove più processi o attività devono operare in parallelo o interagire in modo sincronizzato.

L'astrazione consente di semplificare e organizzare la complessità di un sistema attraverso la riduzione del dettaglio e la focalizzazione sugli aspetti fondamentali.

Il ruolo dell'astrazione

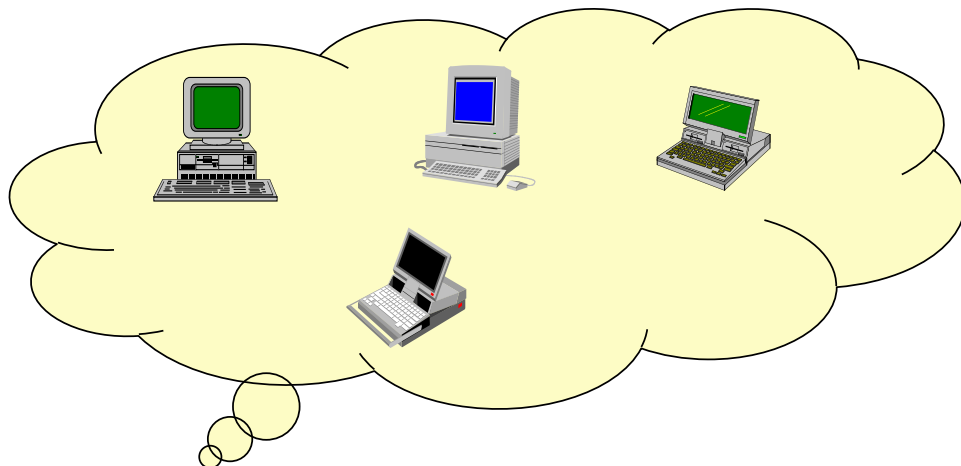
- ### ❑ Molteplici sono le relazioni in gioco fra oggetti, funzioni e stati e molteplici i livelli di possibile dettaglio:
- ⇒ Gli **oggetti** possono essere descritti a partire da termini molto generici (*edificio, strada*) fino ad arrivare a livello di dettaglio specifici (*la torre degli Asinelli, via Saragozza*)
 - ⇒ Le **funzioni** possono essere espresse inizialmente in modo vago (*controllare il livello di gas nocivi nell'aria*) e successivamente precisate (*la programmazione del livello di soglia per l'allarme della centralina viene attivata premendo il pulsante P*)
 - ⇒ Gli **stati** possono essere descritti a un elevato livello di astrazione (*la centralina è in stato di errore*) o specificati in maggior dettaglio (*è acceso il segnalatore di errore nel sensore S*)

Meccanismi di Astrazione

- I principali meccanismi di astrazione usati durante il processo di analisi per costruire una base di conoscenza sul problema sono ¹classificazione, ²generalizzazione, ³aggregazione e ⁴associazione

1 Classificazione

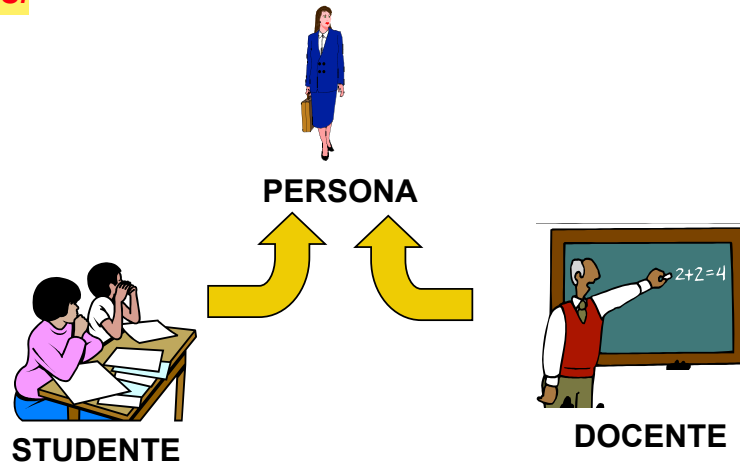
- La classificazione consente di raggruppare in classi oggetti, funzioni, o stati in base alle loro proprietà



la classe dei computer

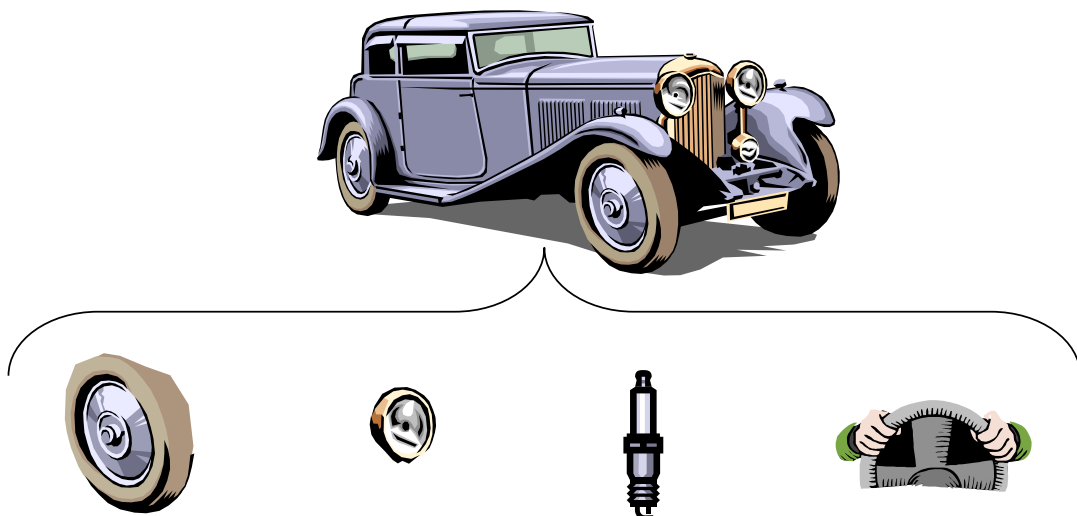
2 Generalizzazione

- La generalizzazione cattura le relazioni **è-un** ovvero permette di astrarre le **caratteristiche comuni** fra più classi definendo **superclassi**



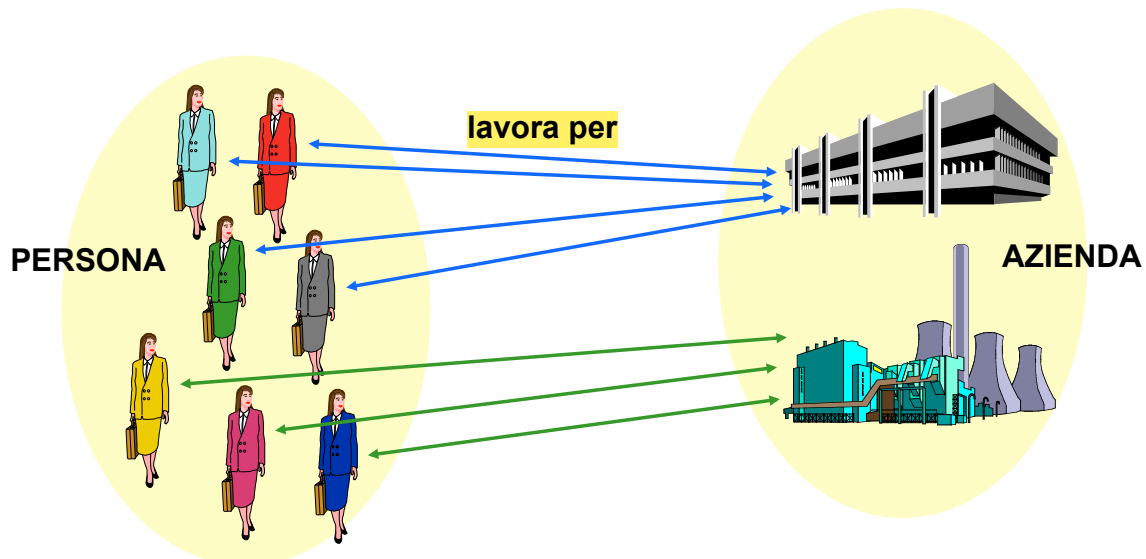
3 Aggregazione

- L'aggregazione esprime le relazioni **parte-di** che sussistono tra oggetti, tra funzioni, tra stati



4 Associazioni

- ❑ Oltre ai meccanismi citati, è importante modellare le associazioni che sussistono fra le varie classi



23

Linguaggi per la specifica dei requisiti

- ❑ **Linguaggi informali**
 - ⇒ Il linguaggio naturale, alla base della comunicazione durante le interviste tra analista e utente, non può essere adottato come unico mezzo per produrre documenti di specifica per le innumerevoli ambiguità di significato
- ❑ **Linguaggi semiformali** combinano il linguaggio naturale con notazioni grafiche o diagrammi che introducono una maggiore struttura e precisione, riducendo le ambiguità.
 - ⇒ notazione grafica, che presenta una semantica sfumata, accoppiata con descrizioni in linguaggio naturale (esempi : E/R, DFD) → poiché dipendono ancora da una certa interpretazione soggettiva del diagramma o delle descrizioni associate.
- ❑ **Linguaggi formali** utilizzano una notazione rigorosa e matematica, con l'obiettivo di eliminare ogni ambiguità e garantire che i requisiti siano espressi in modo preciso e verificabile.
 - ⇒ linguaggi di specifica basati sulla logica dei predicati
 - ⇒ linguaggi di specifica algebrici
 - ⇒ linguaggi concettuali per basi di dati

24

Linguaggi per la specifica dei requisiti

↪ i requisiti di un sistema possono essere molto vari e avere obiettivi differenti, e per questo motivo è necessario usare tipi diversi di notazioni per rappresentare le informazioni relative a questi requisiti in modo adeguato.

La diversità degli obiettivi posti dalla specifica dei requisiti implica l'utilizzo di notazioni diverse per la rappresentazione delle informazioni

↪ è utile quando si vuole descrivere come il sistema deve comportarsi

Formalismi operazionali: definiscono il sistema descrivendone il comportamento (normalmente mediante un modello)

"E è il percorso che si ottiene muovendosi in modo che la somma delle distanze tra due punti fissi p_1 e p_2 rimanga invariata"

↪ si concentrano sulle proprietà che un sistema deve avere, piuttosto che su come il sistema raggiunge quelle proprietà.

Formalismi dichiarativi: definiscono il sistema dichiarando le proprietà che esso deve avere

$$ax^2+by^2+c=0$$

L'approccio operativo fornisce una rappresentazione più semplice poiché più simile al modo di ragionare della mente umana:

- Facilità di acquisizione è spesso intuitivo e più simile al modo in cui la mente umana pensa ai problemi
- Facilità di verifica della correttezza è più semplice verificare se il sistema si comporta correttamente durante lo sviluppo o attraverso i test.
- Facilità di comprensione da parte dei programmatori I formalismi operazionali sono spesso più vicini al linguaggio di programmazione

L'approccio dichiarativo fornisce una rappresentazione che non si presta ad ambiguità

3

Progettazione

- ❑ Riguarda tutte quelle attività che permettono di passare dalla raccolta ed elaborazione dei requisiti di un sistema software alla sua effettiva realizzazione, pertanto fa da ponte tra la fase di specifica e la fase di codifica
- ❑ Durante la fase di progettazione si decidono le modalità di passaggio da "che cosa" deve essere realizzato (specifica dei requisiti) a "come" la realizzazione deve avere luogo
- ❑ Il sistema complessivo viene suddiviso in più sottosistemi

Vantaggi:

- ⇒ la complessità delle singole parti è minore della complessità totale originaria
- ⇒ i sottosistemi ottenuti possono essere realizzati ed analizzati da gruppi diversi di programmatori in modo il più possibile indipendente

Progettazione

□ Due esigenze contrastanti:

per capire se il progetto rispetta i requisiti e le aspettative, ma senza entrare troppo nei dettagli tecnici.

⇒ progetto sufficientemente astratto per poter essere agevolmente confrontato con le specifiche da cui viene derivato

⇒ progetto sufficientemente dettagliato in modo tale che la codifica possa avvenire senza ulteriori necessità di chiarire le operazioni che devono essere realizzate

alla fine della fase di progettazione, il progetto deve contenere tutte le istruzioni precise per i programmatori

□ A seconda della tecnica impiegata per la progettazione, la realizzazione del sistema può risultare più o meno naturale ed immediata

Alcuni approcci progettuali sono più orientati alla facilità di implementazione, mentre altri favoriscono una visione più astratta del sistema.

⇒ Ad esempio:

progettazione orientata a oggetti



realizzazione in un linguaggio a oggetti

Progettazione

➤ Non esiste un metodo generale per la progettazione del software

⇒ Occorre considerare le tipologie di software (software sequenziale, concorrente ed in tempo reale)

□ A una stessa specifica possono corrispondere più progetti, ossia più metodi di soluzione diversi

□ Le scelte di progetto devono poter cambiare in risposta a mutate esigenze di vario tipo senza che per questo tutto il progetto e perciò tutto il software prodotto debba essere modificato radicalmente

Il progetto deve essere flessibile e modificabile in risposta a mutate esigenze o cambiamenti nel corso dello sviluppo.

➤ Il progetto di un sistema software è perciò un'attività altamente creativa, che richiede un insieme di abilità a coloro che vi sono preposti

↑ Obiettivi della progettazione

- 1 □ Produrre software con le caratteristiche di qualità che sono state dettagliate nella fase di analisi e specifica dei requisiti. Ad esempio:
 - ⇒ affidabilità
 - ⇒ modificabilità
 - ⇒ comprensibilità
 - ⇒ riusabilità
- 2 □ ...obiettivi che si possono riassumere nella diminuzione dei *costi* e *tempi* di produzione e nell'aumento della *qualità* del software

I costi maggiori riguardano la fase di manutenzione del software



⇒ capacità di poter far fronte a modifiche da effettuare senza che l'intera struttura dell'applicazione già costruita debba essere messa nuovamente in discussione ed elaborata