

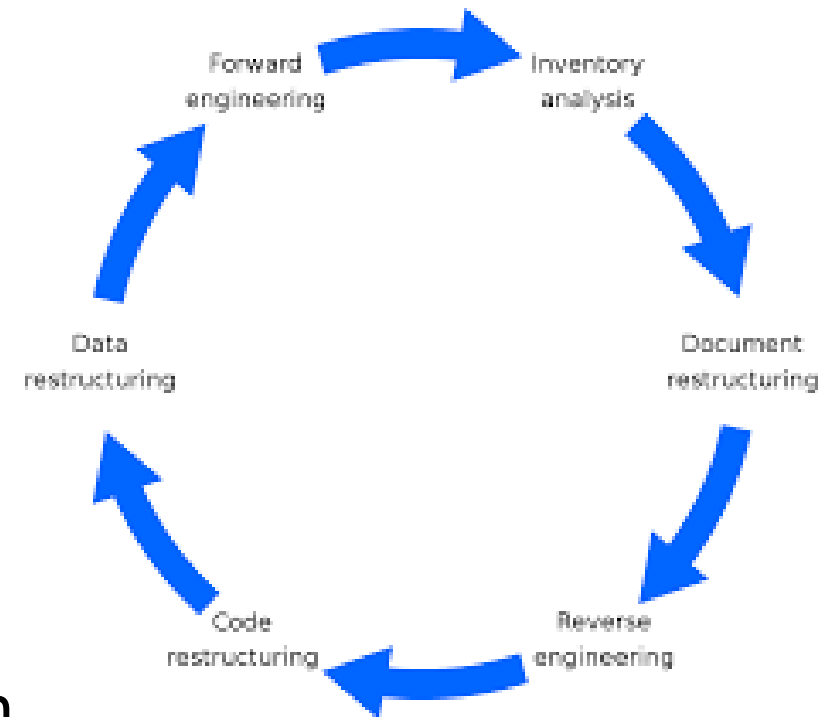
# Analisi e progettazione del Software

*Francesco Pugliese, PhD*

*neural1977@gmail.com*

# Analisi e progettazione del Software

- ✓ **L'Ingegneria del Software** proponeva, anni fa, una distinzione netta tra le attività di analisi e design (progettazione):
- ✓ **Analisi:** lo studio di “**cosa**” deve fare il sistema (punto di vista “**logico**”).
- ✓ **Design:** lo studio di “**come**” è opportuno implementare il sistema (punto di vista “**tecnico**”).
- ✓ Si suggeriva, in particolare, di non iniziare il **design** prima di aver chiarito tutti i requisiti per il nuovo sistema. A partire almeno dagli anni novanta del '900, questa raccomandazione di metodo è entrata in crisi:



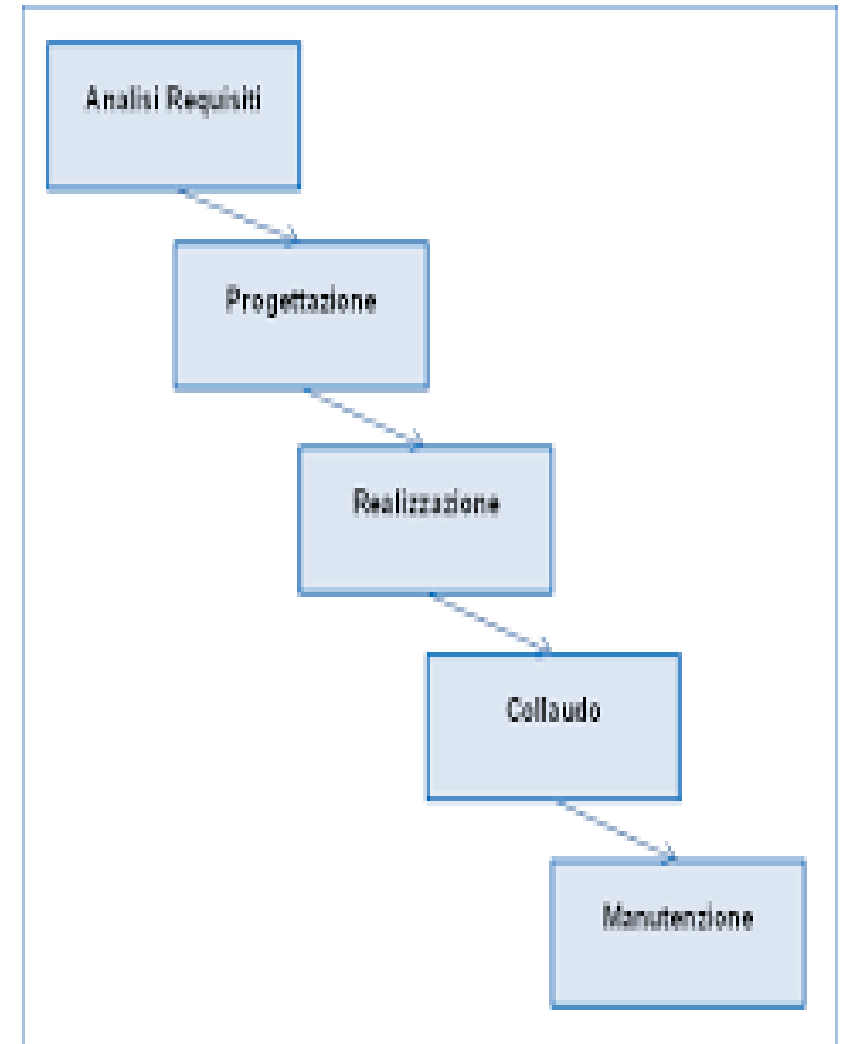
# Analisi e progettazione del Software

- ✓ La maggior parte dei **manuali sull'Ingegneria del Software** raccomanda di non iniziare la **progettazione** fino a quando tutti i requisiti non sono concordati.
- ✓ È un consiglio sbagliato, perché comunque non arriverete **mai a conoscere tutti i requisiti**, e iniziare presto la progettazione vi servirà probabilmente a **scoprire nuovi requisiti.**" (Alan Davis, "Just Enough Requirements Management", 2005)



# Significato di Analisi

- ✓ Il termine "**analisi**", d'altra parte, può essere interpretato e declinato in diversi modi, e in organizzazioni diverse può assumere significati molto distanti tra loro.
- ✓ In **Ingegneria del Software**, l'**analisi** è una **delle fasi del ciclo** di vita del software; essa ha lo scopo generale 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**.



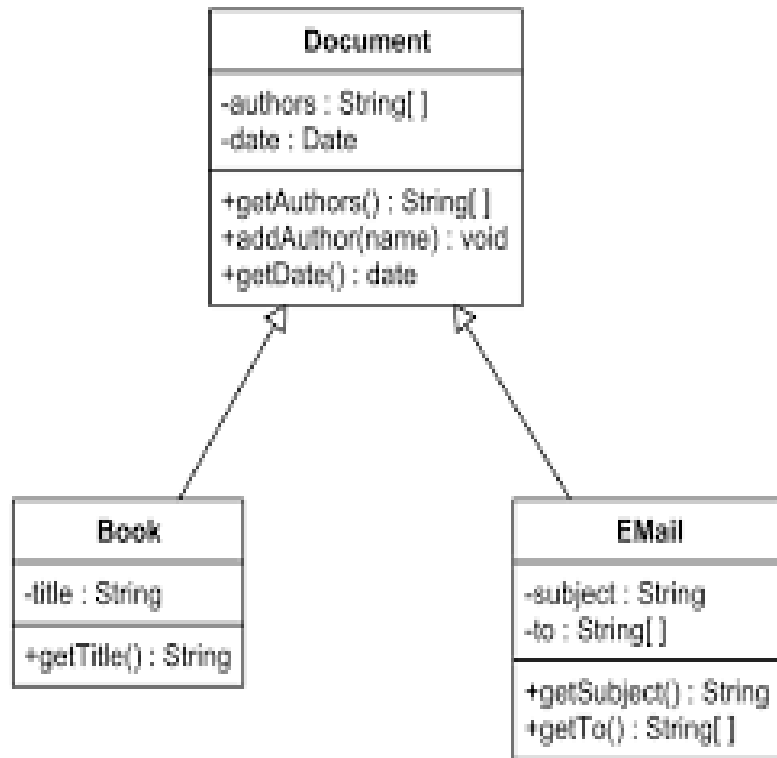
# Significato di Analisi

---

- ✓ Le **informazioni** raccolte nella fase di analisi rappresentano il punto di partenza per la **progettazione** di un prodotto software e per l'intero **processo della sua realizzazione**, validazione e manutenzione.
- ✓ Se l'analisi sia una fase che viene conclusa una volta per tutte all'inizio del processo di realizzazione di un prodotto software, oppure un'attività che viene svolta iterativamente nel processo stesso, dipende dal particolare **modello di ciclo di vita del software** adottato nel progetto.



# Significato di Analisi



- ✓ I **modi** e gli **strumenti** (sia concettuali che tecnici) con cui viene svolta l'analisi dipendono dalla **metodologia** scelta.
- ✓ Nel caso della **analisi object-oriented**, per esempio, gli strumenti possono includere notazioni come **UML**, che consente la descrizione del **dominio applicativo**, nonché di come il sistema deve interagire con tale dominio, in termini di classi, relazioni fra classi, e altri concetti tipici dello sviluppo **orientato agli oggetti**; e gli strumenti possono includere ambienti **CASE (Computer Aided Software Engineering)** specifici come **Rational Rose**.

# Significato di Progettazione

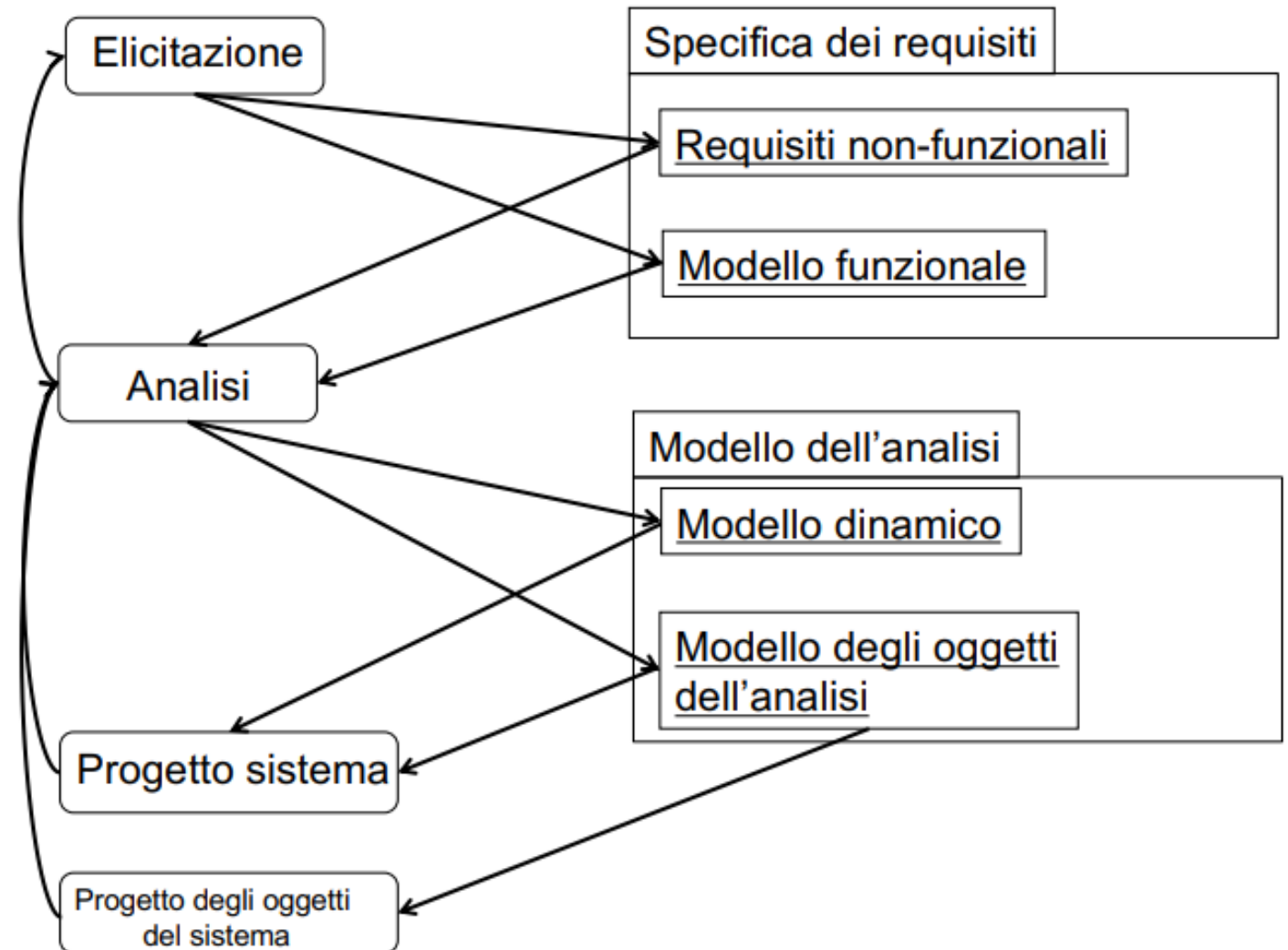
- ✓ **Progettare** = anticipare un futuro artefatto mediante **problem solving** creativo.
- ✓ **L'analisi** si occupa di: Capire il **problema**, e preparare la base per una **soluzione** mediante un **modello di analisi**.





# Significato di Analisi e Progettazione

- ✓ La **progettazione** vera e propria si occupa di **descrivere** (anticipare) una **soluzione** al problema mediante un modello **progettuale** che di solito si ispira ad un qualche **paradigma** progettuale.





# Significato di Analisi e Progettazione

---

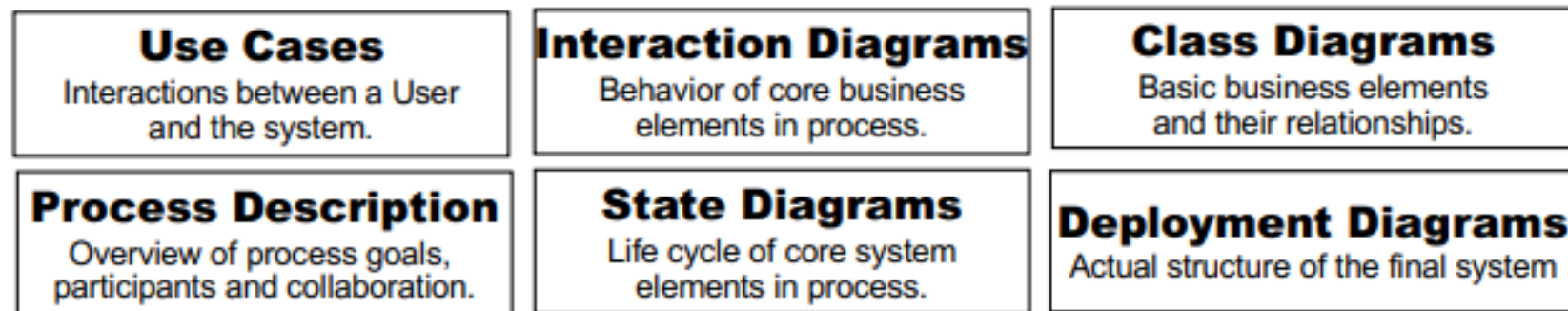
- ✓ Un **modello** è una rappresentazione **semplificata** della realtà che contiene **informazioni** ottenute focalizzando **l'attenzione** su alcuni aspetti cruciali e **ignorando** alcuni dettagli
- ✓ Un **modello di un'applicazione** sw descrive e rappresenta in modo semplificato la sua **struttura** ed il suo **comportamento**



# I modelli del sw richiedono più viste

---

- ✓ Un buon **modello** include almeno quattro viste:
  - Vista di **Contesto** (dell'ambito di uso o scopo)
  - Vista **Funzionale** (compiti in relazione allo scopo)
  - Vista **Strutturale** (elementi e relazioni tra gli oggetti)
  - Vista **Comportamentale** (dinamica e interazioni)



*Funzione*

*Comportamento*

*Struttura*

# I modelli del sw richiedono più viste

---

1. La vista di **contesto** descrive ambito e scopo del sistema da progettare (es. "Vision document")
2. La vista **funzionale** descrive la forma di un sistema in relazione allo scopo (es. "Documento dei Casi d'uso")
3. La vista **comportamentale** modella il comportamento di un sistema in relazione ai possibili input
4. La vista **strutturale** modella i componenti di un sistema e le loro relazioni

Le prime due viste sono "**soggettive**" (descrivono le intenzioni del progettista), invece le seconde sono "**oggettive**" (modellano proprietà future del sistema)

# Bibliografia

---

<https://analisi-disegno.com/analisi/analisi-design/>

<http://www.cs.unibo.it/~cianca/wwwpages/ids/7.pdf>