

# INGEGNERIA DEL SOFTWARE

Alessandro Pioggia

16 gennaio 2022

# Indice

0.1	Analisi dei requisiti . . . . .	2
0.1.1	Analisi orientata agli oggetti . . . . .	2
0.1.2	Analisi funzionale . . . . .	2
0.1.3	Analisi orientata agli stati . . . . .	2
0.1.4	Astrazione . . . . .	3
0.1.5	Linguaggi utilizzati per la specifica dei requisiti . . . . .	3
0.2	Progettazione . . . . .	4
0.2.1	Modalità . . . . .	4
0.2.2	Approccio . . . . .	4
0.2.3	Il buon progettista . . . . .	4

## 0.1 Analisi dei requisiti

L'analisi dei requisiti è la fase che permette, attraverso la modellazione della realtà, di redigere la specifiche dei requisiti, documento che rappresenterà l'input per le successive fasi di progettazione. Per rendere possibile la creazione un coadiuvato è necessario, da parte dell'analista, intervistare il produttore, cercando di assimilare al meglio tutte le informazioni riguardanti il dominio applicativo.

In questa fase è necessario fare in modo che il documento sia chiaro, non ambiguo, accessibile e privo di contraddizioni.

L'analisi è incrementale e deve comunicare gli aspetti statici, dinamici e funzionali del progetto software.

### 0.1.1 Analisi orientata agli oggetti

In questa tipologia di analisi vengono curati principalmente gli aspetti statici, ovvero vengono definiti gli oggetti e le relazioni presenti fra essi. Definire gli oggetti significa individuare tutte le informazioni e proprietà ad essi connesse, esse tendono a rimanere invariate nel tempo, per questo viene definito un approccio **statico**.

### 0.1.2 Analisi funzionale

L'analisi funzionale si pone il problema di definire le specifiche prendendo come riferimento le funzioni, ovvero viene studiata solo ed esclusivamente la relazione presente fra dati in ingresso e dati in uscita. Vengono presi in considerazione i flussi informativi, che verranno poi modificati da processi.

### 0.1.3 Analisi orientata agli stati

L'analisi orientata agli stati analizza i vari stati evolutivi del prodotto, in funzione di ciò, vengono analizzati i comportamenti e successivamente redatta la documentazione delle specifiche dei requisiti.

### **0.1.4 Astrazione**

I meccanismi di astrazione utilizzati sono:

- classificazione
- ereditarietà (is-a)
- aggregazione (part-of)
- associazione

### **0.1.5 Linguaggi utilizzati per la specifica dei requisiti**

- informale (linguaggio naturale, molto contraddittorio e poco chiaro, sconsigliato)
- semiformale (utilizzo di diagrammi secondo un preciso standard, che però è semplice ed intuitivo (e/r, dfd))
- formale (linguaggio tecnico, difficile da comprendere e inutilmente complicato, indipendentemente dal contesto)

Una volta definito il tipo di linguaggio da utilizzare è necessario effettuare una distinzione fra formalismi dichiarativi e operazionali. I primi definiscono il problema indicando le proprietà che esso deve avere, i secondi ne descrivono il comportamento, nella maggior parte dei casi attraverso un modello. I formalismi operazionali sono i più indicati in questo contesto, in quanto più comprensibili e modellabili.

## **0.2 Progettazione**

Se la fase di analisi vuole sincerarsi sul "che cosa" sviluppare, nella parte di progettazione ci si chiede "come" svilupparlo, dunque si ha una sorta di ponte fra analisi e codifica.

### **0.2.1 Modalità**

Nella progettazione si suddivide il problema iniziale in più sottoproblemi, il più possibile indipendenti fra loro, questo dà la possibilità di fare gestire il lavoro a più team, anche in parallelo. Una fase di progettazione ben realizzata consente un notevole risparmio di risorse, dal momento che un eventuale errore in fase di produzione o peggio, durante la manutenzione comporta un costo molto più elevato per l'azienda.

### **0.2.2 Approccio**

L'approccio varia in base alla strategia adottata, ne conosciamo due:

- generale
- specifico

Il primo permette di mantenere una buona elasticità, può essere modellato anche nelle fasi successive. Il secondo invece è rigido, non contempla modifiche nelle fasi successive, questo semplifica il passaggio dalla progettazione alla codifica.

### **0.2.3 Il buon progettista**

Il buon progettista è colui che ha una buona conoscenza di tutto ciò che riguarda lo sviluppo software, sa anticipare i cambiamenti ed ha un buon grado di esperienza. Inoltre, è colui che si pone come obiettivi della progettazione l'affidabilità, la modificabilità, la comprensibilità e la riusabilità del software.