

# Basi di Dati 2

Marco Casu



# Contents

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Contesto Organizzativo . . . . .	3
1.2	Ciclo di Vita del Software . . . . .	3
1.3	Il linguaggio UML . . . . .	5

# 1 Introduzione

Questo corso non è ristretto esclusivamente alla progettazione di basi di dati, bensì fornisce cenni sulla progettazione di software di grandi dimensioni, supportati da basi di dati reali.

Un cliente (committente) fornisce delle specifiche riguardo un progetto che bisogna sviluppare, esso stesso non sa come verrà implementato o quali sono nello specifico tutte le funzionalità, un insieme di ingegneri del software, progettisti, e programmatori si occuperanno di "tirare su" il lavoro completo nel tempo, e varie figure professionali verranno necessariamente coinvolte.

*Tempi per un progetto software complesso :*

- Capire il problema e cosa vuole realmente il cliente : 33% del tempo totale.
- Progettazione, capire come implementare le richieste del cliente : 50% del tempo totale.
- Effettiva realizzazione (sviluppo del codice) : 17% del tempo totale.
- Del tempo extra per i test di verifica e la manutenzione.

## 1.1 Contesto Organizzativo

Le figure professionali *chiave* coinvolte nel progetto sono dette **attori**, generalmente sono :

- Committente ed Esperti del dominio
- Analisti e Progettisti
- Programmatori
- Utenti finali e Manutentori

Qual'è la differenza tra analisti e progettisti? E di cosa si occupano gli esperti del dominio?

Il **dominio** dell'applicazione è l'insieme di informazioni necessarie da conoscere per poter lavorare ad un progetto che fa riferimento ad uno specifico ambito, ad *esempio*, un applicazione che si occupa di registrare e gestire le contravvenzioni stradali, vedrà sicuramente nel suo dominio il codice stradale e le informazioni legislative.

L'esperto del dominio è una figura, appunto esperta, del dominio inerente al progetto in questione, viene pagata dal committente e funge da consulente durante lo sviluppo.

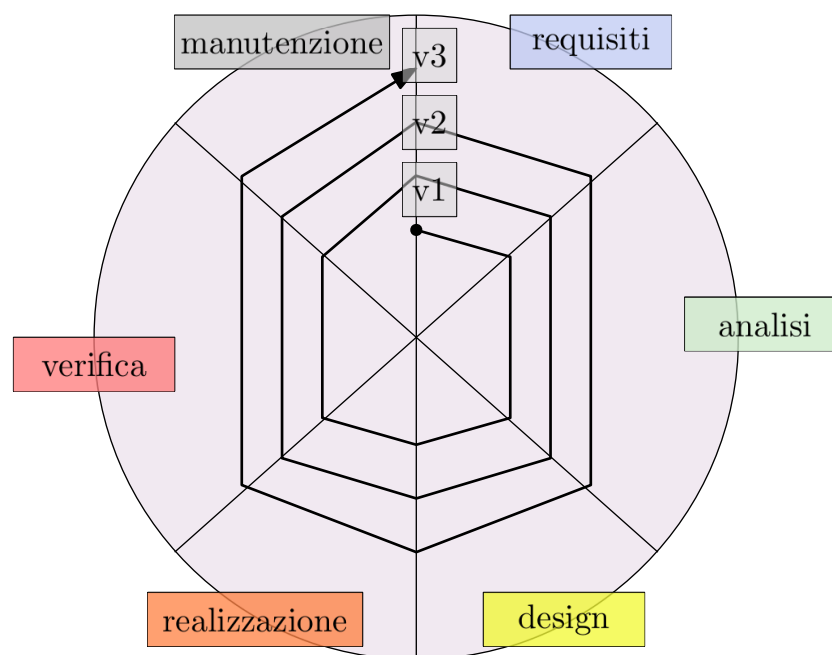
## 1.2 Ciclo di Vita del Software

È possibile suddividere lo sviluppo di un software in macro-fasi principali.

1. **Studio di fattibilità** - Ci si approccia al progetto valutando i costi per realizzarlo ed i benefici, si pianificano le attività e le risorse del progetto, umane ed economiche, e si individua l'ambiente di programmazione hardware e software.
2. **Raccolta dei requisiti** - Bisogna capire *cosa il sistema deve fare*, scrivere in prosa una documentazione che descriva precisamente le usabilità del progetto, sintetizzando i requisiti, che spesso sono contraddittori, trovando i giusti compromessi.

3. **Analisi concettuale dei requisiti** - Sono coinvolti gli analisti, che produrranno uno schema matematico del progetto, dettagliato per filo e per segno, che definirà cosa l'applicazione deve fare indipendentemente dal come. Lo schema prima citato è detto *schema concettuale*, e sarà la base da cui partire per la progettazione.
4. **Progettazione (design) dell'applicazione** - Bisogna capire *come* il sistema realizzerà le sue funzioni, entra in gioco il progettista, che definirà l'architettura volta ad ospitare il software e l'insieme delle tecnologie necessarie.
5. **Realizzazione** - Una volta che si hanno le linee guida per la realizzazione, composte nelle fasi precedenti, si delega la scrittura del codice ai programmatori, che non sono coinvolti nel resto e non devono necessariamente essere a conoscenza di cosa stanno facendo, ma esclusivamente produrre le funzioni richieste.
6. **Verifica, esercizio e manutenzione** - Le diverse componenti dell'applicazione vengono integrate. Una volta che il progetto è realizzato e pronto alla messa in esercizio, si passa da una fase di testing ad una fase di utilizzo effettivo, l'applicazione verrà monitorata durante l'esercizio ed eventuali correzioni verranno prodotte.

Si osservi il seguente diagramma rappresentante il **modello a spirale** di realizzazione :



Tutto il progetto viene costruito in maniera "iterativa", si dice che lo sviluppo del software sia *agile*, si comincia raccogliendo i requisiti strettamente necessari, per poi procedere all'analisi considerando tali requisiti, con l'andare avanti delle fasi portando alla realizzazione di una prima versione del software, pronta ad essere messa in esercizio, implementante esclusivamente le funzionalità di base, tale versione renderà chiare le idee al committente che potrà fornire nuovi requisiti, in modo tale da ricominciare il ciclo.

Nulla vieta alle varie fasi di essere eseguite in parallelo, ad esempio, nel tempo  $t_0$  vengono stilati i requisiti per la prima versione del software, nel tempo  $t_1$  gli analisti iniziano a produrre il modello della versione 1, ma possono essere nel mentre stilati i requisiti della versione 2, al tempo  $t_3$ , com'è di facile intuizione : Si raccolgono i requisiti per la versione 3, si produce il modello della versione 2, si progetta la versione 1.

### 1.3 Il linguaggio UML