**TECNOLOGIE A LIVELLO APPLICATIVO**

Una delle caratteristiche importanti del livello applicativo riguarda la modularità. Infatti la maggior parte delle applicazioni del SI sono modulari, cioè divise in moduli, ovvero componenti autonomi associati a una o più funzionalità.

È molto utile avere applicazioni modulari, perché rende più semplice la manutenzione: si possono modificare solo specifici moduli senza cambiare tutto il SI.

Inoltre, spesso si fa la scelta Make-or-Buy per i singoli moduli: se l’organizzazione deve usare dei moduli che ricoprono funzionalità sfruttate da molte altre aziende, spesso conviene acquistare questi specifici moduli invece che svilupparti. I moduli già esistenti sul mercato sono detto componenti COTS (=commercial-off-the-shelf).

Fra le applicazioni più acquistate, e non sviluppate, abbiamo le ERP (Enterprise Resource Planning) e le CRM (Costumer Relationship Management).

**ERP**

Gli ERP (Enterprise Resource Planning) sono sistemi basati su tre proprietà:

1. Unicità dell’informazione
2. Modularità
3. Prescrittività

Analizziamole nel dettaglio:

*Unicità dell’informazione* significa che esiste una **sola** rappresentazione dei dati su cui il sistema ERP opera, e questa è comune per tutti i moduli del sistema ERP. Quindi tutti i moduli del sistema vedono gli stessi dati.

*Modularità* significa che la suite software è composta da moduli autonomi, che possono essere acquisiti e integrati in modo incrementale. Ogni modulo è composto da funzioni che supportano attività elementari. È possibile aggiungere a un sistema già esistente moduli dello stesso fornitore di quelli già esistenti oppure di un fornitore diverso.

*Prescrittività* significa che il sistema regolamenta la logica di funzionamento del sistema. Cioè il software può stabilire delle regole (ad esempio che si possono ricevere solo prodotti che sono stati ordinati, che si possono fare ordini solo da fornitori autorizzati…) che l’azienda deve seguire.

I moduli di un ERP possono essere divisi in tre categorie: sistemi istituzionali (si occupano di gestione amministrativa, delle risorse umane, della contabilità), moduli settoriali (attività primarie come la pianificazione delle risorse, gli acquisti, le vendite,…) e moduli direzionali (si occupano di pianificazione strategica, controllo del budget, …).

Dal punto di vista architetturale, gli ERP sono sviluppati con tre diversi approcci (dal più “vecchio” al più recente).

* *Architettura client-server*: tutti i moduli funzionali erano installati su uno o più server e gli utenti accedevano alle funzionalità tramite client.
* *Web-based ERP*: Sistemi ERP sviluppati come applicazioni web, raggiungibili tramite un browser. In questo modo, gli utenti possono accedere all’ERP della propria organizzazione anche da remoto.
* *Moduli con architettura SaaS* (=Software as a Service) (Analizzata nel dettaglio più avanti)

L’evoluzione degli ERP nel tempo li ha resi sempre meno costosi nell’acquisizione e gestione. Infatti l’approccio SaaS è molto meno costoso dell’architettura client-server, ad esempio perché sviluppare ERP con la logica a servizi è meno costoso per il produttore e quindi costa meno acquistare il software da parte dell’azienda, e con questa logica l’organizzazione che usa l’ERP paga l’utilizzo e non la licenza+implementazione (si ricorda infatti che gli ERP tendenzialmente sono applicativi COTS, cioè acquistati, non sviluppati ad hoc dalle singole organizzazioni).

Essendo gli ERP applicativi COTS, sono realizzati dal produttore non per una singola azienda, ma per un determinato settore. Esistono quindi suite diverse per diversi settori.   
Un ERP supporta quasi tutte le attività operative, tranne pianificazione e progettazione di rete.  
Non essendo sistemi specifici, al momento dell’acquisto l’organizzazione deve configurare l’ERP per adattarlo alle proprie esigenze. Questo è possibile attraverso la parametrizzazione: esistono parametri nel software ERP che possono essere settati in modo diverso a seconda di quello che serve all’organizzazione. L’organizzazione, volendo, può anche modificare il codice dell’ERP per aggiungere funzionalità ulteriori.

Gli ERP introducono diversi vantaggi. Innanzitutto aumentano l’efficienza: infatti permettono di risparmiare sul tempo e sui costi automatizzando le procedure. Inoltre migliorano l’efficacia: rendono infatti i dati più accessibili e disponibili, grazie all’unicità dei dati e alla capacità di elaborare e veicolare le informazioni. Questo rende i dati maggiormente disponibili quando richiesti per prendere decisioni a livello strategico.

**CRM**  
  
Il CRM (Costumer Relationship Management) è una suite software che supporta le interazioni fra l’organizzazione e i suoi clienti. Ha l’obiettivo di fidelizzare i clienti costruendo relazioni personalizzate con loro. È utile per le aziende di quei settori in cui l’intensità dei rapporti col cliente è maggiore, oppure hanno un numero molto elevato di clienti, oppure i clienti possono comunicare con l’azienda attraverso vari canali (multicanalità).   
Il CRM è diviso in due componenti principali: CRM analitico (back-end, analizza dati dei clienti) e CRM operativo (front-end, supporta le interazioni coi clienti). Esiste anche un terzo modulo, il CRM collaborativo, che calcola alcuni indicatori utili per determinati attori all’interno dell’organizzazione a cui interessa analizzare i dati dei clienti e delle interazioni con loro.   
  
CRM Operativo

Il CRM operativo, come accennato, si occupa dell’interazione fra organizzazione e clienti. È costituisco da diversi componenti:

* *Marketing*: L’obiettivo odierno delle aziende è fidelizzare i clienti. Per fare questo servono buone campagne di marketing. Più si riesce ad acquistare e analizzare dati, migliori saranno le campagne di marketing. Alcuni sistemi CRM che si possono implementare per aumentare la soddisfazione dei clienti sono :
  + Generazione liste clienti: produce liste di clienti adatti a ricevere una specifica comunicazione di marketing;
  + Gestione campagne: è un sistema che guida l’organizzazione nella pianificazione delle campagne di marketing. Rende più automatizzato il processo di progettazione e analisi del risultato della campagna.
  + Cross-selling e upselling: il primo indica vendere un maggior numero di prodotti a un cliente, il secondo indica aumentare il valore dei prodotti venduti a un cliente.
* *Automazione vendite*: è un sistema che supporta tutto il processo di vendita, dall’identificazione del cliente alla conclusione dell’acquisto. I moduli per il supporto alle vendite sono:
  + Gestione vendite: si occupa della selezione dei contatti, aiuta il venditore nella stesura dell’offerta e del contratto, fornisce tutte le funzionalità per supportare le vendite online;
  + Gestione contatti: gestisce i dati dei clienti. Tiene nota delle interazioni precedenti per rendere più personalizzate quelle successive e migliorare il rapporto col cliente. (Si occupa di persone che sono già clienti;
  + Gestione opportunità: cerca nuovi clienti, calcola la probabilità che una vendita vada a buon fine con un certo cliente. (Si occupa di persone che non sono ancora clienti).
* *Servizi ai clienti*: è importante mantenere i contatti coi clienti anche dopo la vendita. Per questo ci sono i servizi ai clienti. Essi si compongono di:
  + Call center: il CRM gestisce tutte le interazioni telefoniche, sia quelle effettuate dai clienti verso l’organizzazione sia viceversa. Indirizza inoltre il cliente verso il giusto operatore;
  + Web based self-service: permette ai clienti di contattare l’azienda o trovare risposte alle proprie domande su internet
  + Call scripting: fornisce agli operatori le informazioni che servono per risolvere i problemi dei clienti che li contattano.

Il CRM analitico invece analizza i dati dei clienti.   
Ha tre funzionalità principali:  
 - Reporting, che aiuta a capire chi sono i clienti, le loro preferenze;  
 - Analysis, che aiuta a dividere i clienti in categorie a secondo di quanto sono attivi come clienti dell’azienda;  
 - Predicting, per predire i desideri dei clienti.

Il CRM può essere acquisito in modalità Make o in modalità Buy, e possono essere gestiti all’interno dell’organizzazione o in outsourcing (cioè la gestione è affidata a terze parti). I benefici del CRM sono una maggiore soddisfazione dei clienti, un aumento del numero di clienti, una semplificazione dei processi di marketing e vendita.

**DATA WAREHOUSING**  
Il Data Warehouse si occupa principalmente di dati a livello strategico. È necessario avere la possibilità di accedere questi dati in tempo reale e di effettuare query complesse. Gli strumenti che permettono questo appartengono alla categoria applicativa *Business Intelligence* accennata in BOAT.   
I primi strumenti di supporto alle decisioni erano i report, documenti testuali con dati analitici; successivamente sono stati sostituiti dai fogli di calcolo, che permette di automatizzare eventuali calcoli, ma ha comunque dei problemi per quanto riguarda l’integrità e correttezza delle analisi, permettendo un accesso individuale non controllato. Per supportare adeguatamente la gestione dei dati per il livello strategico serve quindi un database che abbia le seguenti caratteristiche:

* Può includere anche dati da fonti esterne
* I dati devono essere strutturati in modo che siano facilmente analizzabili
* La struttura del DB deve essere semplice
* I dati devono essere aggiornati
* I tempi di risposta degli strumenti di analisi devono essere brevi

Un database con queste caratteristiche è detto Data Warehouse.  
Un DB data Warehouse è di tipo OLAP.   
I dati del Data Warehouse sono utilizzati per prendere decisioni strategiche; ci accedono i manager; sono principalmente dati storici, d’archivio; sono dati aggregati a partire da dati operazionali; l’accesso è in lettura, non in scrittura.  
Un database di questo tipo è multi-dimensionale, rappresentato in un *ipercubo* (=matrice multidimensionale). Le dimensioni possono essere organizzate gerarchicamente. Gli elementi di una base di dati multidimensionale sono:

* Fatto: è l’elemento dell’*ipercubo* ottenuto specificando un valore per ogni dimensione (cioè un elemento, una casella della matrice).
* Dimensione: sono le coordinate di ciascun elemento
* Misura: valore quantitativo del fatto

Un Data Warehouse comprende anche tutte le operazioni legate all’acquisizione e alla gestione dei dati. Ricordiamo che si tratta di dati a livello strategico e di pianificazione della Piramide di Anthony.

Proprietà del Data Warehouse  
Dato che il Data Warehouse è un sistema OLAP, e dunque i dati sono in sola lettura, non avrà le proprietà tradizionali delle basi di dati che contemplano la possibilità di effettuare modifiche (proprietà ACID) , ma proprietà diverse, dette FASMI. Le proprietà FASMI sono:

* *Fast*: il sistema deve rispondere rapidamente alle richieste
* *Analytical*: deve permettere analisi complesse
* *Shared*: permettere l’accesso a più utenti, con permessi diversi
* *Multidimensional*
* *Informational*: deve contenere tutte le informazioni utili

Deve inoltre essere:

* *Orientato alle entità*, cioè considerare le principali entità di analisi (ad esempio sono entità di analisi le vendite, gli ordini,…)
* *Integrato*: i dati provengono da fonti interne ed esterne
* *Variabile nel tempo*: memorizza dati storici, quindi deve tenere traccia del periodo in cui sono stati acquisiti i diversi dati
* *Persistente*: i dati inseriti non devono essere modificati.

Architettura del Data Warehouse  
All’interno del Data Warehouse (DW) possiamo trovare diverse basi di dati organizzate in modo gerarchico. Queste basi di dati sono:

* *Sorgenti*, cioè fonti da cui devono essere estratti i dati; possono essere interne o esterne. Vengono sottoposte a operazioni dette ETL (Extraction, Trasformation, Loading) che permettono di estrarre, modificare e salvare i dati nel nostro DW.
* *Staging area,* è un database intermedio fra sorgenti e DW in cui sono memorizzati i dati prima di essere salvati nel DW.
* *Data mart,* sono piccoli database tematici che contengono un “pezzo” dell’informazioni contenuta nel DW completo.

ETL  
Abbiamo visto che i processi ETL consistono nell’acquisizione, trasformazione e salvataggio dei dati dalle sorgenti al DW. L’estrazione può essere statica (prende tutti i dati) oppure incrementale, se vengono presi i dati “nuovi” rispetto all’ultima estrazione e li aggiunge a quelli già estratti.  
Per il processo di trasformazione si effettuano:

* *Pulizia* dei dati, cioè se ci sono degli errori vengono corretti;
* *Riconciliazione*: si aggregano i dati relativi allo stesso “oggetto”;
* *Standardizzazione* dei formati: se i dati sono eterogenei, vengono resi omogenei nel formato;
* *Ricerca ed eliminazione dei duplicati*.

I processi ETL sono supportati dai Metadati, cioè una documentazione che fornisce le informazioni sulla struttura del DW, che memorizza le sorgenti e le trasformazioni per ogni dati, per mappare i dati operazionali a quelli caricati nel DW, che tiene statistiche sull’utilizzo del DW.

Modello concettuale del DW

Uno dei vari modelli concettuali usati per rappresentare sistemi multidimensionali è il Dimensional Fact Model (DFM). In questo modello, il fatto è rappresentato come un rettangolo che contiene le misure corrispondenti. Le dimensioni sono cerchi collegati al fatto, come attributi. Possono essere semplici oppure gerarchie. Alcune dimensioni possono essere opzionali.   
  
Modello logico del DW  
Per passare dal modello concettuale a quello logico si deve scegliere un DBMS da utilizzare.   
Le varie possibilità sono:

* Modello *MOLAP* (Multidimensional-OLAP), traduce un modello concettuale multidimensionale in un database multidimensionale che può essere interrogato usando motori appositi. Ha il vantaggio di avere una corrispondenza 1:1 col progetto concettuale. Ha lo svantaggio di “sprecare” tanta memoria per allocare l’ipercubo ed essere difficile da interrogare per un utente medio.
* Modello *ROLAP* (Relational OLAP): traduce il progetto concettuale multidimensionale in un modello relazionale, che usa tabelle e può usare linguaggi basati su SQL. È più semplice da usare per gli utenti e usa meno memoria rispetto al MOLAP. Però è meno fedele del MOLAP al progetto concettuale e le query sono lente.
* Modelli HOLAP (Hybrid OLAP) usa un database relazionale per il DW, e dei database multidimensionali per i data-mart.

Per l’approccio ROLAP, si trasforma il progetto concettuale da multidimensionale a relazionale in due modi: facendo la join di tutte le tabelle per un fatto su un’unica tabella, oppure usando tante tabelle per un solo fatto.

Operatori OLAP  
L’utente interagisce con il DW applicando un operatore al database. I possibili operatori sono:

* *Drill down*: si scende lungo la gerarchia di una dimensione, passando da un livello di aggregazione più basso (quindi entrando nel dettaglio);
* *Roll up*: opposto del drill down;
* *Slice:* permette di focalizzare l’analisi su una porzione di dati, fissando il valore per una delle dimensioni (ad esempio analizzare le vendite di un *singolo* punto vendita);
* *Dice:* permette di focalizzare l’attenzione su una porzione di dati, fissando delle coordinate (quindi i valori di più dimensioni) (Ad esempio estratte dati delle vendite di un prodotto specifico in uno specifico punto vendita).

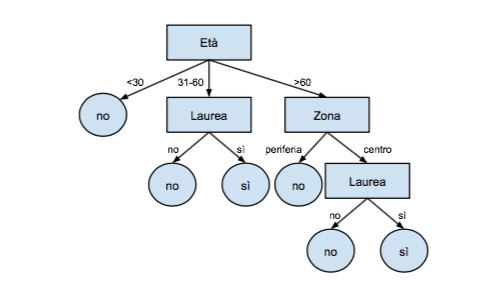
Limiti del DW  
Aggiornare un fatto significa inserire un valore nuovo o aggiornare un valore esistente, e non ha side-effects. Aggiornare una dimensione invece può essere problematico, perché bisogna decidere di situazione in situazione come gestire i valori precedenti, se sovrascriverli, mantenerli o tenere sia quelli “vecchi” sia quelli “nuovi”.  
Inoltre a volte il processo ETL non è completamente automatizzato, richiede intervento umano, e può essere anche molto lento.

**Data Mining**Il data mining consiste nell’estrarre informazioni da database molto grandi, proprio come il DW.   
Il processo di data mining si compone dei seguenti passi:

1. Selezione dei dati di interesse
2. Pulizia di dati per correggere errori
3. Integrazione di dati da fonti diverse
4. Trasformazione dei dati
5. Analisi volta alla ricerca di condizioni notevoli nei dati analizzati
6. Valutazione dei risultati
7. Consegna dei risultati all’utente

I primi 4 passi sono i classici passi già visti per la creazione di un DW.

Le funzioni di data mining si distinguono a seconda delle loro proprietà.   
Innanzitutto si può fare agire istruendo gli algoritmi di data mining con apprendimento supervisionato, cioè si fornisce alla funzione una variabile target (che coincide con l’output desiderato a un determinato input) e l’algoritmo “impara” così che tipo di output deve cercare se gli viene fornito un certo input. Nell’ apprendimento non supervisionato invece non viene fornita una variabile target, l’algoritmo “cerca da solo” dei pattern e delle regole nei dati e da lì trae le conclusioni su come deve agire.   
Inoltre si possono usare tecniche descrittive o predittive. Le prime analizzano i dati e li categorizzano. Le seconde analizzano i dati e attraverso essi fanno previsioni per il futuro.

**Regole Associative**Le regole associative sono una tecnica di data mining che rientra nelle categorie *dell’apprendimento non supervisionato* e delle *tecniche descrittive.* Esse descrivono le relazioni fra vari attributi nel database.  
Il loro scopo è identificare condizioni che spesso si verificano contemporaneamente. Sono solitamente espresse come un’implicazione (A => B). Questa relazione può essere espressa probabilisticamente tramite la probabilità condizionata, cioè il rapporto delle volte in cui A e B si verificano insieme sulle volte che A si verifica.   
Un esempio di uso delle regole associative consiste nell’applicarle alle campagne di marketing: tramite questa tecnica si possono suggerire prodotti da acquistare in base ai precedenti acquisti dei clienti.   
  
**Classificazione**  
La classificazione è una tecnica di data mining che rientra nelle categorie *dell’apprendimento supervisionato* e delle *tecniche predittive.* Essa permette di assegnare in modo automatico gli elementi del database ad alcune classi predefinite.Essendo una tecnica ad apprendimento supervisionato, l’algoritmo parte da elementi di cui la classe è nota (a noi e all’algoritmo), e costruisce un modello per individuare le caratteristiche che fanno sì che un oggetto sia assegnato ad una classe piuttosto che a un’altra. Poi si testa l’algoritmo su elementi sempre di classe nota (a noi, ma non all’algoritmo) per vedere se le categorizza correttamente. Gli elementi usati per istruire l’algoritmo sono detti training set, quelli per testarlo test set. Un esempio di modello di classificazione sono gli alberi di decisione, che hanno per nodi gli attributi del soggetto da classificare, per archi i possibili valori degli attributi, per foglie le classi a cui possiamo assegnare gli elementi.   
Possiamo ad esempio creare un albero di decisione che prenda in esame età, titolo di studio, residenza di possibili clienti e decidere in base a questi se è probabile che acquisteranno un certo prodotto.   
Esempio: 

In questo caso le classi sono due: cliente che acquisterà il prodotto e cliente che non lo acquisterà. Ovviamente la predizione non è perfetta, e un cliente potrebbe essere assegnato alla classe sbagliata.   
Sulla base di errori e successi si stabiliscono due parametri: precision e recall.   
Il parametro *precision* è la percentuale di elementi classificati correttamente per una classe sugli elementi che vengono classificati come appartenenti a quella classe. Il parametro *recall* indica la percentuale di elementi classificati correttamente per una classe su tutti quelli che effettivamente appartengono a quella classe.

**Clustering**  
Il clustering è una tecnica di data mining che rientra nelle categorie *dell’apprendimento non supervisionato* e delle *tecniche descrittive.* A differenza della classificazione, l’algoritmo che opera il clustering non richiede di conoscere a priori le classi di appartenenza del training set, le “trova da solo” identificando dei pattern. Questa tecnica divide gli elementi in cluster, con le seguenti regole:

* La similarità fra elementi di uno stesso cluster è massima
* La similarità fra elementi diversi cluster è minima

Quindi il clustering si basa sul raggruppare elementi che siano il più simili possibili fra loro. Uno degli algoritmi di clustering più utilizzati è il *k-means*; si tratta di un algoritmo che richiedere di conoscere a priori il numero k dei cluster possibili e una “funzione di similarità” che indica la vicinanza tra due elementi. Per definire una funzione di similarità, si calcola la differenza fra i valori degli attributi di due elementi e poi si fa la somma pesata di queste differenze, a seconda della loro rilevanza per la nostra analisi (per due persone, si potrebbe fare la differenza pesata delle età, della zona di residenza, del titolo di studio, ..., poi fare la somma pesata per dare più importanza ai dati più rilevanti e raggruppare così le persone che hanno il maggior grado di similarità, che hanno il maggior numero di caratteristiche – prese con il loro peso – che siano il più simili possibile).   
Tenendo presente che un cluster può essere visto come un insieme contenente vari punti, si può analizzare come procede l’algoritmo.  
Si procede così: si identifica per ogni cluster il punto che ha per coordinate il valor medio degli attributi degli elementi del cluster; questo punto viene denominato *centroide*; gli elementi del set di dati da classificate vengono assegnati ognuno a un cluster a seconda della loro “distanza” (cioè si calcola la funzione di similarità) dal centroide; ora che i cluster sono “popolati”, si ricalcolano le coordinate dei centroidi e si trovano nuovi centroidi; si ripete la divisione degli elementi; si ripetono tutti i passaggi precedenti. L’algoritmo continua così, iterativamente, finché converge, cioè la posizione dei centroidi, una volta ricalcolata, non cambia più rispetto al calcolo precedente.   
  
Gli algoritmi di clustering possono essere divisi in categorie a seconda di come associano gli elementi ai cluster.   
Un algoritmo esclusivo assegna ogni elemento a un solo cluster.  
Un algoritmo sovrapposto può assegnare un elemento a più cluster.   
Un algoritmo fuzzy è un algoritmo sovrapposto in cui all’assegnamento di un elemento a ogni cluster è associato un peso fra 0 e 1.   
Un algoritmo è completo se ogni elemento è almeno in un cluster o parziale se esistono elementi che possono rimanere senza cluster.

**Selezione del software**Come abbiamo visto, molti software sono COTS, cioè vengono comprati, non sviluppati.   
Per scegliere un software COTS da acquistare, bisogna per prima cosa analizzare i requisiti. I requisiti possono essere *funzionali*, se sono costituiti da una lista di funzionalità che il software dovrà fornire, o *non funzionali*, se definiscono le proprietà e i vincoli architetturali.   
I requisiti devono essere valutati attraverso degli indicatori. Essi possono essere:

* *Indicatori funzionali*: sono le funzionalità supportare dal software;   
  fra essi, il valore più importante è il grado di completezza, cioè il numero di funzionalità fornite dal software su quelle richieste. Un altro indicatore funzionale importante è la personalizzabilità, cioè quanto il software può essere adattato alle esigenze dell’azienda.
* *Indicatori architetturali*: sono gli aspetti tecnologici critici, quelli più importanti per la scelta del software. I principali sono:
  + Scalabilità dell’infrastruttura: il software deve poter reggere l’aumento del carico di lavoro o delle funzionalità del SI.
  + Interoperabilità della soluzione: quanto il software è “aperto verso il mondo esterno:
  + Sicurezza del sistema
* *Costi della soluzione*

Alla valutazione delle caratteristiche del software stesso va affiancata la valutazione del fornitore. Esso si valuta secondo i principi di credibilità e capacità del produttore, cioè l’esperienza, la reputazione, l’attenzione al prodotto.