

# **Business Analysis II**

---

Programmazione di Progetto

Prof. Thimoty Barbieri

Università degli Studi di Pavia



# Obiettivi della lezione

---

- Presentare alcuni criteri standard per stimare le dimensioni di uno sviluppo
- Presentare alcune tecniche per organizzare nel tempo lo sviluppo, suddividendolo in milestone e dividendo le attività tra le risorse



# Il Conteggio Funzionale

---

- Le tecniche di conteggio funzionale misurano la dimensione del software quantificando le funzionalità fornite verso l'esterno, basandosi principalmente sul disegno logico del sistema. Tenendo conto di tale fatto si può ricordare che gli obiettivi del conteggio sono:
  - Valutare ciò che viene chiesto dall'utente e ciò che gli viene fornito
  - Misurare un software indipendente dalla tecnologia utilizzata per lo sviluppo
  - Fornire una metrica dimensionale tale da supportare studi di qualità e di produttività
  - Fornire una base per studi relativi alla preventivazione e stima
  - Fornire un fattore di normalizzazione che permetta il raffronto tra sistemi di organizzazioni diverse
- Per soddisfare i requisiti sopra citati, il processo di misurazione dovrebbe essere:
  - Abbastanza semplice, per diminuire il sovraccarico del processo di misurazione
  - Conciso, per garantire la congruenza tra tempo, progetti e gestione del personale applicativo.



# Conteggio Funzionale FP

---

- L'analisi a Function Point è un metodo standard per misurare lo sviluppo del software *dal punto di vista dell'utente*, a valle dell'analisi a Use Case.
- Essa quantifica le funzionalità del software basandosi principalmente sulla progettazione logica, con l'obiettivo di:
  - misurare le funzionalità che l'utente richiede e riceve
  - misurare la manutenzione e lo sviluppo del software in maniera *indipendente dal tipo di tecnologia utilizzata* (host monolitico, CS, Web...)
- Il conteggio deve essere coerente tra tutti i progetti e tra tutte le organizzazioni che utilizzano questa metodologia ( in modo da consentire la creazione di repositories di conteggio tra categorie, raffronti tra le prestazioni di varie aziende, etc.)



# Metodo Generale FP

---

- La metrica dei FP misura un'applicazione basandosi su due aree di valutazione. La prima è basata sul valore degli Unadjusted Function Point (UFP), che riflette le funzionalità fornite all'utente dell'applicazione; la seconda valutazione fornisce il valore dei Fattori di Aggiustamento, che valorizza la complessità delle funzionalità generali fornite all'utente. Affinando il conteggio ottenuto dalla prima fase con il Fattore del Valore di Aggiustamento ottenuto dalla seconda fase, viene prodotto il conteggio dei Function Point finale.
- La procedura generale da seguire per il conteggio è la seguente:
  - 1. Determinare il *Tipo di Conteggio*
  - 2. Identificare l'*Ambito di Conteggio* e i *Confini Applicativi*
  - 3. Identificare tutte le *funzioni dati*
  - 4. Identificare tutte le *funzioni transazionali*
  - 5. Eseguire il conteggio *unadjusted*
  - 6. Determinare il valore del fattore correttivo *Caratteristiche Generali Sistema*
  - 7. Finalizzare il calcolo usando il fattore correttivo



# Metodo Generale FP

---

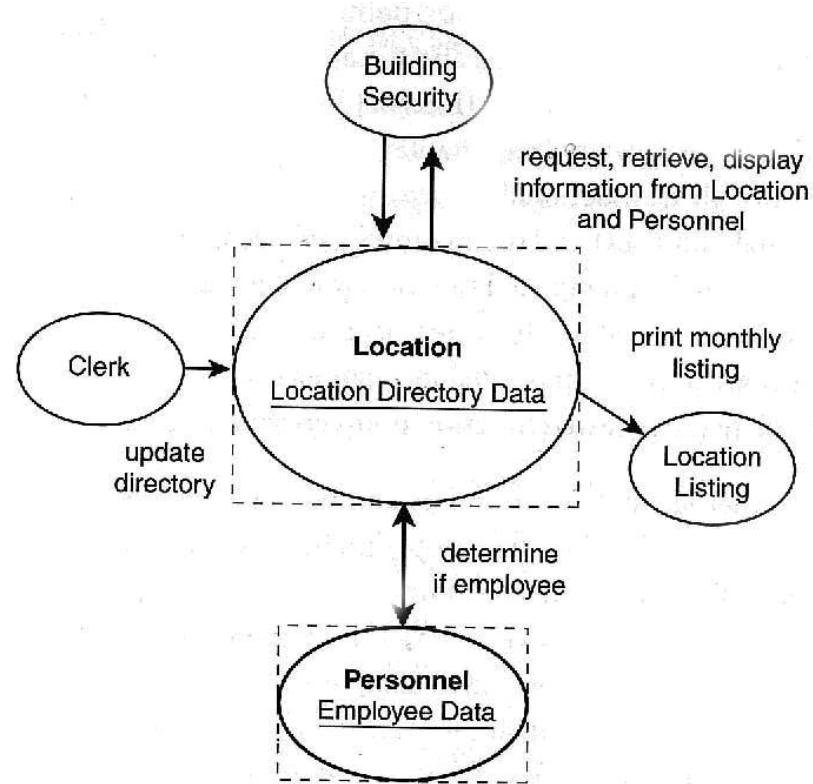
- Nel conteggio degli UFP le funzionalità vengono valorizzate considerando il “cosa” viene consegnato all’utente e non il “come”; vengono, infatti, conteggiati solo i componenti richiesti dall’utente e a questo visibili. Tali componenti vengono distinti in Function Type, tali Function Type possono essere Dati o Transazioni.
- Dati:
  - Internal Logical File (ILF) – aggregati logici di dati gestiti internamente al sistema
  - External Interface File (EIF) – aggregati logici di dati scambiati o condivisi dall’applicazione con altre applicazioni
- Transazioni:
  - External Input (EI) – informazioni distinte fornite dall’utente e usate dal sistema come dati in ingresso
  - External Output (EO) – output distinti che il sistema ritorna all’utente come risultato delle proprie elaborazioni
  - External Inquiry (EQ) – interrogazioni in linea che producono una risposta immediata dal sistema
- La complessità di ogni Function Type è valutata su una scala nominale di tre punti: Basso, Medio, Alto. Alla complessità sono assegnati dei punteggi. La somma dei punteggi dà un valore che esprime la dimensione della funzione analizzata (*Misura in FP*).



# Esempio

---

- Nell'applicazione per la gestione del database delle località:
- Funzioni Dati
- ILF: il database delle località, EIF: il database del personale
- Funzioni Transazionali
- EI: aggiornamento dell'impiegato del database località
- EO: Lista stampata mensile delle località
- EQ: Richiesta caratteristiche località dal rep. Sicurezza; sincronizzazione del database del personale



# Calcolo dei FP

- Il calcolo dei Function Point avviene seguendo tre passi:
- valorizzare gli Unadjusted Function Point (UFP), contando le funzioni dati e transazionali secondo la tabella fornita.
- valorizzare le caratteristiche generali (GCS) ed ottenere il TDI (Total Degree of Influence). Calcolare il VAF (Value Adjustment Factor) utilizzando la formula:
  - $VAF = (TDI \times 0.01) + 0.65$
- valorizzare i Function Point (FP) utilizzando ad es. la formula :
- $FP = UFP * VAF$  (per applicazioni conteggio iniziale, per altri tipi di conto vedi oltre)
- Tabella del conteggio UFP

Componente	Basso	Medio	Alto
EI	X 3	X 4	X 6
EO	X 4	X 5	X 7
EQ	X 3	X 4	X 6
ILF	X 7	X 10	X 15
EIF	X 5	X 7	X 10



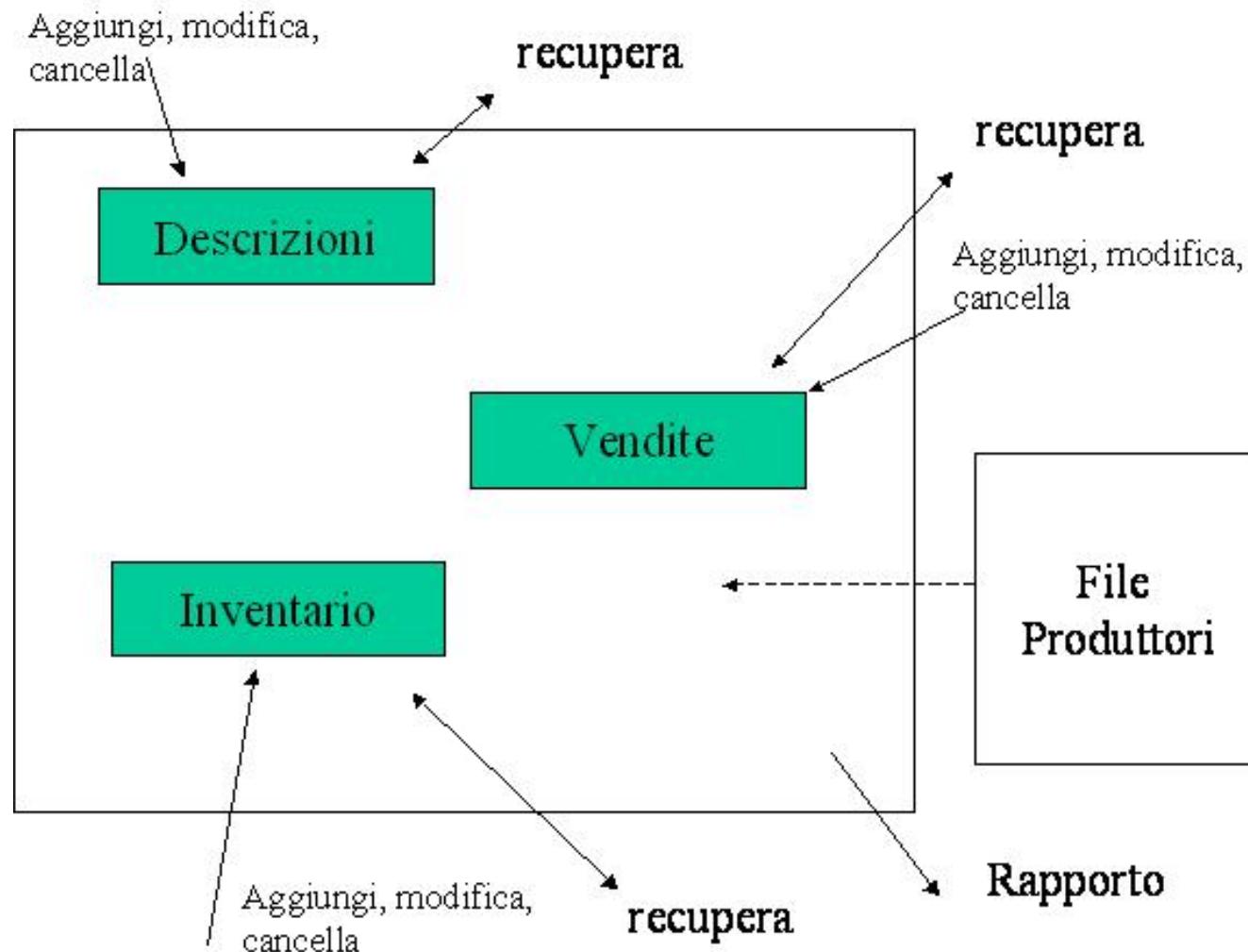
# Esempio

---

- Si consideri lo sviluppo iniziale di un'applicazione di Catalogo Commerciale. Esso consente di inserire, modificare, cancellare articoli, memorizzati in una tabella articoli interna all'applicazione. La tabella articoli dispone di una sola chiave e di oltre 20 campi per ciascun articolo. E' anche possibile per l'utente recuperare a video le descrizioni.
- L'applicazione dispone inoltre di una tabella inventario, con una chiave e oltre 20 campi per ciascuna voce, per cui è possibile inserire, modificare, cancellare voci, e recuperare a video le voci.
- Un'ulteriore tabella Vendite, con una chiave e 20-50 campi, consente di aggiungere, modificare, cancellare vendite, e recuperare a video le voci.
- A fine mese viene generato un rapporto calcolato partendo dai dati delle tre tabelle, ed appoggiandosi ai dati contenuti in un file esterno all'applicazione, che contiene gli indirizzi dei produttori degli articoli commercializzati, e che serve per completezza dei dati di rapporto.
- Disegnare lo schema dell'applicazione ed eseguire il conteggio.



# Schema dell'Esempio



Descrizione	Tipo	DET	RET/FTR	Compless.
Descrizioni	ILF	30	1	B
Descr: agg.	EI	>15	1	M
Descr: mod	EI	>15	1	M
Descr: del	EI	< 5 (confine!)	1	B
Descr: rec	EQ	> 19	1	M
Inventario	EIF	> 15	1	B
Inv: agg.	EI	>15	1	M
Inv: mod	EI	>15	1	M
Inv: del	EI	< 5 (confine!)	1	B
Inv: rec	EQ	> 19	1	M
Vendite	EIF	> 15	1	B
Vend: agg.	EI	>15	1	M
Vend: mod	EI	>15	1	M
Vend: del	EI	< 5 (confine!)	1	B
Vend: rec	EQ	> 19	1	M
Rapporto	EO	> 19	> 1	A
Produttori	EIF			B



# Conteggio FP Totali

Componente	Basso	Medio	Alto
EI	<b>3 X 3</b>	<b>6 X 4</b>	X 6
EO	X 4	X 5	<b>1 X 7</b>
EQ	X 3	<b>3 X 4</b>	X 6
ILF	<b>3 X 7</b>	X 10	X 15
EIF	<b>1 X 5</b>	X 7	X 10

- Totale UFP = **78**
- GSC considerate: Data Comm = 4, Distribuzione dell'elaborazione = 0, Prestazioni = 3, Utilizzo intensivo configurazione = 2, Transaction Rate = 3, Online Data Entry = 5, End User Efficiency = 4, Online Update = 3, Complex Processing = 1, Reusability = 0, Installation Ease = 0, Operational Ease = 0, Multiple Sites = 1, Facilitate Change = 2. **TDI = 31**
- **VAF = (TDI x 0.01) + 0.65 = 0.96**
- **FP APPL INIZ = UFP x VAF = 74.88 = 75**



# Conteggio in OOP

---

- L'Analisi a FP nasce per cercare di essere il più possibile svincolata dalla tecnica implementativa. Tuttavia la sua impostazione tradisce un retaggio legato ad applicazioni "mainframe", che ragionano in termini di tracciati e di flussi.
- Per poter applicare il conteggio FP a paradigmi più complessi quali OOP, occorre una "mappatura" convenzionale tra i concetti FP ed i concetti OOP. Tuttavia, allo stato della specifica (4.1.1) non esiste una mappatura ufficiale. Tutto si riconduce comunque a tentare di prescindere il più possibile dall'implementazione, per mantenere un punto di vista logico a livello utente.
- Il punto di vista base per il conteggio, oltre alle specifiche, è il class diagram UML.



# Conteggio FP applicato a OOP

FP	OOP
ILF	Classe logica <sup>[1]</sup> interna all'applicazione. Attributi modificabili
EIF	Classe logica esterna all'applicazione. Attributi/Stato non modificabili
DET	Attributi logici <sup>[2]</sup> della classe
RET	Attributi utilizzati in modo polimorfico per raccogliere valori di diversi sottotipi; Presenza di ereditarietà (la classe contata è sottoclasse); Presenza di specializzazione unicamente dal punto di vista logico (la classe contata è superclasse)
Processo Elementare	Non è detto che coincida con l'invocazione di un singolo metodo: riferirsi a sequence diagram di alto livello e agli use case.
EI	Metodo invocato che cambia lo stato di una classe interna
EQ	Metodo che presenta all'esterno del boundary stati di una o più classi interne
EO	Metodo che presenta all'esterno stati di una o più classi interne modificandole, o per calcolo, o per alterazione dello stato generale del sistema.

[1] La cui presenza cioè non è data da motivi strettamente implementativi, ma dall'analisi dei requisiti

[2] Attributi cioè la cui presenza non è data da motivi implementativi



# Conteggio di Applicazioni Web

---

- Anche nel conteggio FP di una web application, il problema è dato dall'impostazione “legacy” dell'analisi. In più, la realizzazione di una web application è un frutto di una integrazione estremamente complessa di numerose tecnologie in stato di continua evoluzione (HTML, DHTML, JavaScript, XML, Flash, XLST, J2EE, ASP, PHP, ASP.NET ... ), e la ripetibilità dei processi produttivi in questo contesto è piuttosto scarsa. Anche in questo caso si può tuttavia tentare un mappaggio tra concetti Web e concetti FP.



FP	Web
ILF/EIF, DET/RET	<p>Legate alle basi dati su cui si appoggia la web application, oppure ai file di sorgente dati da cui attinge per il funzionamento. Distinguere tra dati che vengono effettivamente modificati dalla web application, e tra dati che sono solo usati come lettura (possibilmente da link esterni - EIF).</p> <p>La scrittura di dati in sessione nelle web appl (ASP/JSP/PHP/.NET) è da considerarsi trattamento con ILF.</p>
EI/EO/EQ	Contate sulle maschere HTML o generate dinamicamente con scripting serverside che compongono l'interfaccia applicativa verso l'utente internet. Un salvataggio da una form dentro il db è una EI. Un invio di dati via email all'utente che ne fa richiesta è una EQ se i dati sono solamente "pescati", una EO se sono risultato di un calcolo a partire dai dati interni alla web application.
Confine Applicativo	Dato dal dominio del sito, e dalla particolare funzionalità di web appl che si analizza. Le funzionalità del browser sono da scartare. Motori di ricerca non sviluppati nell'applicazioni non sono da includere.
Punto di vista utente	Nel caso di una web application, considerare l'utenza del particolare servizio. Nel caso di un intero sito, considerare gli utilizzatori della home page.
Processo Elementare	Se occorrono più pagine HTML/ASP/JSP/PHP per inserire dati, attraverso più form, ma logicamente si tratta di un inserimento unico, occorre considerarlo come un unico processo elementare.



# COCOMO

---

- I FP sono una stima della dimensione del software, ma non dello sforzo e della durata (e quindi del costo) che il software potrà avere. Nella determinazione di questi fattori entrano solitamente in gioco altri parametri che FP non considera, quali la percentuale di software riusato e la facilità con cui riusarlo, la coesione del team di sviluppo, la dimestichezza con la tematica di cui tratta il software o con la tecnologia, la maturità del processo di sviluppo aziendale (si usano standard per la progettazione?), il talento degli analisti e dei programmatore, e via dicendo.
- COCOMO II (Constructive Cost Model 2° Edizione), redatto nel 2000, propone un modello guida per la stima di sforzo e durata del progetto, a partire da una misura della dimensione del progetto (ad es. in FP). Il modello è strutturato in modo da dover essere calibrato sugli effettivi parametri vitali e di funzionamento dell'azienda. Esso propone anche una calibrazione “base” iniziale eseguita su una media di circa 200 progetti.
- Sono disponibili due formule, PM (per calcolare i mesi uomo), e TDEV, per calcolare la durata in giorni. Entrambe si basano su KSLOC (migliaia di righe di codice stimate nel software).



# Formule di Base del COCOMO II

---

- La programmazione nominale in mese uomo è stimata dalla formula
- **$PM = A * (KSLOC ^ E) * \text{product}(EM[i])$**
- dove KSLOC è la dimensione del codice in migliaia di righe di codice sorgente,e
- **$E = B + 0.01 * \text{sum}(SF[i])$**
- A e B sono costanti che derivano dalla calibrazione del modello.
- EM[i] sono gli Effort Multipliers che per un progetto con caratteristiche nominali sono tutti ad 1, dando una produttoria 1.0
- SF[i] sono gli Scale Factors che per un progetto con caratteristiche nominali danno una sommatoria di 19
- Il tempo di sviluppo in mesi, TDEV:
- **$TDEV = C * PM ^ F$**
- Dove  **$F = D + 0.2 * 0.01 * \text{sum}(SF[i]) = D + 0.2 * (E - B)$**
- C e D sono altre due costanti derivate dalla calibrazione del modello
- Le costanti COCOMO II.2000 ottenute su 160 campioni di modello sono:
- **$A = 2.94, B = 0.91, C = 3.67, e D = 0.28$**
- Risolvendo TDEV in funzione di KSLOC:
- $TDEV = 3.67 * (2.94 * KSLOC ^ 1.1) ^ .318$
- Invertendo per trovare KSLOC in funzione di TDEV:
- $KSLOC = (.34 * (.272 * TDEV) ^ 3.14) ^ .909$
- Il numero di persone necessarie nominalmente nel team di ottiene con:
- **$N = PM / TDEV$**



## Esempio COCOMO

---

- Applicazione della formula base nel caso di un progetto stimato a 100 KSLOC. Consideriamo i moltiplicatori di sforzo superiori al valore nominale di 1.0 per modellare uno sviluppo di un grosso progetto
- $$\begin{aligned} \mathbf{PM} &= A * (\mathbf{KSLOC} ^ E) * \mathbf{product(EM[i])} = 2.94 (100) ^ 1.15 \\ &= 586.51 \text{ Mesi Uomo} \end{aligned}$$
- Deriviamo da questo la durata prevista
- $$\begin{aligned} \mathbf{TDEV} &= C * \mathbf{PM} ^ F = 3.67 (586.6) ^ {0.28 + 0.2 \times (1.15 - 0.91)} = 3.67 (586.6) ^ {0.328} = 29.7 \text{ Mesi} \end{aligned}$$
- Da cui si evince che il numero di persone medio necessario al progetto:
- $$\mathbf{PM} / \mathbf{TDEV} = 19.75 = 20$$



# Da FP a KSLOC

---

- Il mappaggio da FP a KSLOC avviene attraverso una tabella, ottenuta mediante lo studio comparato delle due misurazioni. La tabella può essere calcolata internamente sullo storico produttivo aziendale, oppure estratta dalla letteratura.

Linguaggio	SLOC/UFP
C	128
Basic	64
C++	55
Java	53
Visual Basic	29
Visual C++	34
Cobol	91
Assembly	320



# Riconfigurazione del modello COCOMO

---

- In COCOMO II è possibile applicare fattori correttivi per adattare la dimensione in KSLLOC sui seguenti parametri:
- SU : Software Understanding
- AA : Assessment and Assimilation (possibilità di usare in modo rapido porzioni riusabili)
- UNFM: Unfamiliarity dei programmatore con la tecnologia
- REVL: Requirements Unstability and Volatility
- Alcuni degli Effort Multipliers e degli Scale Factors sono invece:
- - PREC (Precedenteness), FLEX (Development Flexibility), TEAM (Team Cohesion), PMAT (Process Maturity), RELY (Software Reliability), DATA (Database Size), RUSE (Develop for Reusability), CPLX (Product Complexity), DOCU (Documentation), TIME (Execution Time Constraint), ACAP (Analyst Capability), PCAP (Programmers Capability), PCON (Personnel Continuity), APEX (Application Experience), PLEX (Platform Experience), LTEX (Language and Tool Experience), TOOL (Use of software Tools).



# Programmazione di Progetto

---



# La gestione del progetto

---

- Un progetto inteso come processo temporaneo, non ripetitivo, con inizio e fine ben determinati, con l'obiettivo di realizzare un input, deve essere gestito mediante un'opportuna strutturazione di tre elementi: persone (organizzazione del personale), prodotto (organizzazione del prodotto), processo (organizzazione delle attività). In generale la "progettazione del progetto" avviene mediante *Breakdown Structures*.
- **Organization Breakdown Structure (OBS)**
  - Individua i componenti organizzativi, cioè le figure, i modi e le strutture che partecipano alla direzione ed alla esecuzione del progetto.
- **Activity Breakdown Structure (ABS)**
  - Definisce i tipi generici di attività direzionali ed esecutive svolte nel progetto, come pianificazione, controllo, progettazione base dati, programmazione, ecc.



# PBS e WBS

---

## ■ Product Breakdown Structure (PBS)

- Descrive la struttura del sistema prodotto del progetto (infrastruttura di rete, hardware, software di supporto, moduli GUI del software applicativo, backend basi dati, ecc.)

## ■ Work Breakdown Structure (WBS)

- Incrocia OBS x ABS x PBS, definisce i pacchetti di lavoro specificando di chi sono competenza (OBS), in quale passo del processo si collocano (ABS) e per quale parte del prodotto sono necessari (PBS). Si tratta dell'unità base di temporizzazione e allocazione costi.



# Introduzione a Project 2000

---

- Microsoft Project 2000 assiste il Project manager (PM) nel lavoro con le attività, le risorse, le assegnazioni, i costi, e le scadenze. Il PM può elaborare piani, previsioni e controllare l'andamento del progetto a mano a mano che è eseguito, apportando correzioni al piano ed essendo in grado di prevedere l'impatto di queste correzioni in termini di tempi e costi.

Il nuovo Project 2000 assiste anche in altri aspetti:

- Schedulazione secondo calendari di attività
- Allarmi sulle scadenze
- Calcolo dei percorsi critici sul progetto
- Gestione delle risorse Lavoro e delle risorse Materiali
- Impostazione delle priorità tra attività, controllo delle risorse sovra-assegnate, strumento di livellazione delle risorse
- Gestione di Gantt e Reticolare a diverso livello di dettaglio, con ricalcolo automatico
- Condivisione dei Gantt e dei piani via rete in modo collaborativo
- Coordinamento delle risorse tra più progetti



# La pianificazione

---

- La teoria generale della gestione del progetto identifica 4 fasi
  - Definire il progetto
    - Impostare gli obiettivi, l'ambito, rilevare i vincoli e fare ipotesi di lavoro
    - Ambito del progetto, Budget e Tempi sono tre aspetti legati tra loro nel "Triangolo del Progetto". Il progetto consiste nel trovare il tradeoff fattibile tra di essi.
  - Creare e raffinare il piano
    - Individuare le attività e schedularle, metterle in sequenza, definire delle milestone (attività cardine), elencare le risorse, le apparecchiature, programmarne i costi, assegnare alle attività. Raccogliere questi dati direttamente o dalle parti in causa
  - Controllare ed aggiornare il piano
    - Confrontare con il piano originale, rischedulare, eventualmente modificare (taglio dell'ambito o dei costi..)
    - Rischedulare e valutarne le conseguenze, riassegnare le risorse per onorare le scadenze.
  - Chiudere il progetto
    - Raccogliere i dati di utilizzo e delle prestazioni e usarli come base per il piano del progetto successivo



# Definizione di Attività

- La lista di attività per la pianificazione si compila seguendo i pacchetti di lavoro individuati. Le attività possono essere di vario tipo (semplici, ricorrenti, cardine), e raggruppate in fasi per consentire un migliore controllo. Le attività cardine (milestones) sono punti di controllo temporali in cui si valuta l'andamento del progetto aspettandosi un risultato (deliverable) intermedio.



# Durate

---

- La durata dell'attività va stimata in base a metriche, esperienze precedenti, incontri con le risorse specializzate. Un'attività ha:
  - inizio e fine determinati
  - personale, materiale, macchinari assegnati
  - un risultato finale misurabile
- L'attività cardine (milestone):
  - va posta alla fine di una serie di attività per evidenziare l'importanza della loro esecuzione
  - solitamente è concordata con il committente
- Le durate possono essere espresse in ore, giorni, settimane o mesi. Si possono impostare le abbreviazioni nelle preferenze del progetto.
- Le durate sono importanti perché le attività possono essere collegate in sequenza. Le durate e le sequenze sono utilizzate per stimare la data finale del progetto. Questo calcolo è preciso tanto quanto la stima delle durate di ciascuna attività è precisa. Si può utilizzare una approssimazione di durata: *ottimistica, pessimistica o attesa*.
- Note:
  - Differenza tra Durata e Durata Stimata (contrassegnata da ?). In questo modo è facile capire se la durata è ancora in fase di stima incerta.
  - Differenza tra Durata e Durata Trascorsa (g contro gt): i gt comprendono i giorni/ora fuori dal calendario lavorativo.



# Analisi PERT

---

- Una **Analisi PERT** aumenta il grado di sicurezza della stima delle durate delle attività, utilizzando delle durate a stima calcolata.
- Solitamente è meglio non inserire una data precisa ma una durata relativa – in questo modo la riprogrammazione delle attività risulta più semplice.
- L'Analisi PERT calcola una durata stimata basandosi su durate ottimistiche, attese e pessimistiche. A ciascuna durata è assegnato un valore. A ciascuna classe di durata è assegnato un peso (default  $1+4+1=6$ ). I pesi possono essere modificati ma la somma deve sempre valere 6. Con l'analisi di PERT, le durate bilanciate compaiono nel Gantt. E' anche possibile visualizzare la programmazione con un Gantt pessimistico, atteso, ed ottimistico.



# Mettere in sequenza le attività

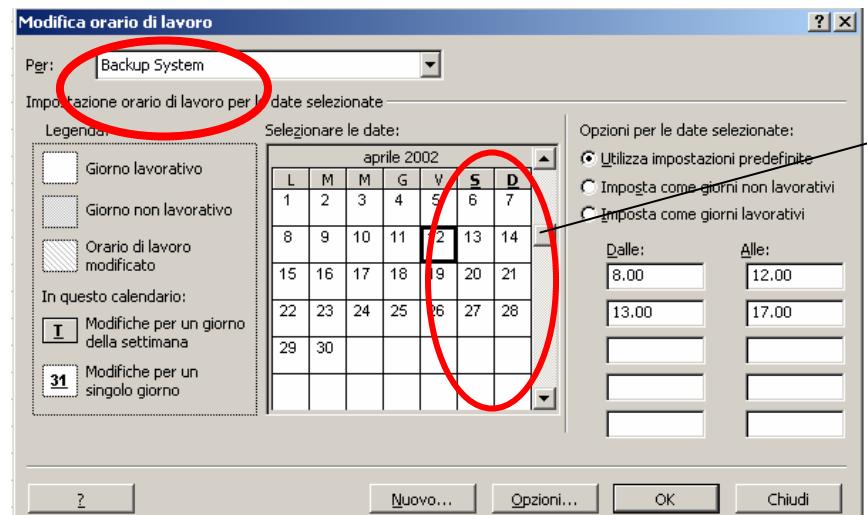
---

- La relazione tra le attività viene usata per riprogrammare in modo flessibile il piano, quando le attività slittano nel tempo. Le attività possono essere *predecessori* o *successori*. Una relazione tra due attività è rappresentata da una freccia. Le relazioni possono essere specificate solo in modo diretto (predecessore o successore diretto). Ci sono diversi tipi di relazioni: Fine-Inizio (FI), Inizio-Inizio (II), Fine-Fine (FF), Inizio-Fine (IF). E' possibile fare doppio click su una relazione esistente, oppure trascinare le barre nel diagramma, oppure utilizzare la finestra Info Attività, per gestire la relazione.
- Ritardo (lag): tra la fine di una attività e l'inizio del successore (un numero positivo o % nel campo ritardo della finestra Relazione)
- Anticipo (lead): può specificare una sovrapposizione temporale tra predecessore e successore (numero negativo o %).
- L'utilizzo di anticipo e ritardo dà ulteriori flessibilità nel rescheduling.



# Calendari di Attività e Risorsa

- Project funziona con il concetto di Calendario. Un Calendario è una particolare configurazione di orari di lavoro e di riposo, e relative date. Il progetto funziona utilizzando il calendario di progetto standard, ma è possibile creare nuovi calendari, ed assegnarli a particolari attività o risorse.
- Un esempio di calendario di risorse può essere quello di una risorsa assegnato ad un turno notturno, oppure di un macchinario che funziona nottetempo.



*Il sistema di backup funziona anche nel weekend*



# Vincoli di attività

---

- Le attività sono rischedulate all'interno del progetto quando necessario. Tuttavia, talvolta occorre imporre dei vincoli temporali su una certa data di inizio o di fine.
- I vincoli di data su una attività consentono di specificare dei limiti temporali, ma consentendo un certo grado di flessibilità nel ricalcolare la programmazione. I vincoli sono di tre categorie:
  - Vincoli Flessibili
    - Consentono di cambiare le date delle attività in conseguenza alla riprogrammazione. "Al più presto possibile" e "Il più tardi possibile" sono due esempi. Il primo si usa per programmare da una data di inizio, il secondo da una data di fine.
  - Vincoli Moderati
    - Forniscono dei riferimenti temporali fissi ma possono essere programmati in un modo più flessibile rispetto ai vincoli non flessibili (es. Finire non oltre il)
  - Vincoli Non Flessibili
    - Legano una data fissa all'inizio o alla fine di una attività. L'attività non viene anticipata o ritardata quando si verifica una riprogrammazione (usare con moderazione)



# Assegnazione di Risorse

---

- Le risorse sono persone, apparecchiature e materiali necessari per portare a termine un'attività; possono essere:
  - Lavoro: persone e apparecchiature che eseguono lavoro in una attività
  - Materiali: materiali di consumo che sono utilizzati (e consumati) durante l'esecuzione dell'attività
- L'aggiunta dell'elenco delle risorse al progetto, con i loro costi e disponibilità consente di migliorare il controllo sul progetto, rendendo possibile:
  - Vedere chi lavora quando e su quale attività
  - Ottimizzare la durata delle attività misurando le risorse da assegnare ad una attività
  - Identificare personale sotto o sovra occupato e livellare il carico di lavoro
  - Tenere traccia del costo di ciascuna risorsa, e quindi di ciascuna attività e dell'intero progetto
  - Mantenere il giusto equilibrio tra ambito, tempo e budget
- Ciascuna risorsa può avere il proprio particolare Calendario.



# Creare la Lista delle Risorse

---

- La Lista delle Risorse comprende il nome della persona, il programma di lavoro, i costi, le ferie, le abilità, ecc. Le risorse possono essere raggruppate secondo un criterio (stesse mansioni, etc.). Una risorsa chiamata come un gruppo può rappresentare più di una persona (es. "Imbianchini" o "Programmatori").
- L'Elenco Risorse consente di inserire la lista delle risorse. Nel campo Tipo è possibile specificare se la risorsa è di tipo Lavoro o Materiale. Se è Materiale, Etichetta specifica l'unità di misura che serve a tenere traccia del consumo. Se si gestiscono anche i costi, è possibile specificarli per ogni risorsa.
- La Lista Risorse può essere utilizzata in comune con altri progetti usando un terzo file Project che viene chiamato Pool di Risorse. (v. oltre)



# Disponibilità delle Risorse

---

- Specifica quale è la percentuale di tempo che una risorsa può dedicare al progetto, in relazione alla disponibilità a tempo pieno. Questa quantità di tempo viene chiamata Unità Massima. Il calendario che specifica i giorni in cui la risorsa può lavorare viene usato per determinare la disponibilità della risorsa.
- Il valore nominale di una Max Unit per una risorsa è 100% - significa che la risorsa è assegnata a tempo pieno al progetto. Quanto sia effettivamente questo tempo, dipende dal calendario assegnato alla risorse, o dal calendario standard del progetto.
- Se la risorsa rappresenta un gruppo (es. "Programmatori"), le MaxUnit possono essere assegnate ad es. al 300% (questo significa tre programmatori che lavorano a tempo pieno sull'attività, con il calendario di progetto o di attività).
- Una risorsa part time ha una Max Unit % di es. 50% o 75% del tempo di lavoro settimanale.
- Questo aspetto può essere collegato ad un "calendario delle disponibilità".



# Le Unità Massime di Assegnazione

---

**Resource Information**

General	Working Time	Costs	Notes
Resource name: <input type="text" value="Carpenter"/>	Initials: <input type="text" value="Car"/>		
Email: <input type="text"/>	Group: <input type="text"/>		
Workgroup: <input type="text" value="Default"/>	Code: <input type="text"/>		
Resource type: <input type="text" value="Work"/>	Material label: <input type="text"/>		
<a href="#">Windows Account...</a>			

Resource Availability

	01/04/2001		
	Available From	Available To	Units
	15/02/2001	01/04/2001	300%
	02/04/2001	09/04/2001	50%

**Buttons:** Help, Details..., OK, Cancel



# Impostare l'Orario di lavoro e i Calendari

---

- E' possibile specificare i giorni lavorativi e non lavorativi per determinate risorse, e che non seguono il normale calendario di progetto. In questo modo si evita di programmare l'uso di risorse quando queste non sono disponibili.
- Innanzitutto si crea una nuovo calendario, poi lo si assegna alla risorsa usando la scheda Orario di Lavoro nella finestra Informazioni risorsa. Le attività assegnate a questa risorsa vengono rischedulate di conseguenza.
- Le Unità Massime sono sempre specificate in relazione all'Orario di lavoro specificato per quella risorsa.
- Questo è particolarmente utile per modellare il lavoro eseguito da un macchinario, che non segue i normali orari di lavoro delle persone.



# Assegnare Risorse alle Attività

---

- Quando si assegna una risorsa ad una attività, è necessario specificare quanta parte di una risorsa è assegnata al task – questa quantità è l'Unità di Assegnamento.
- Per le Risorse Lavoro, è una percentuale del calendario di lavoro. Si riferisce ad una particolare attività, mentre le MaxUnit si riferiscono in generale al progetto. Se le unità di assegnamento sono maggiori delle maxunit della risorsa, allora la risorsa è *sovrallocata*.
- Per le Risorse Materiali, si assegnano dei *tassi di consumo*:
  - fisso: la quantità di materiale consumato è costante, anche se la durata dell'attività cambia
  - variabile: il consumo si basa sull'effettiva durata (tempo) – si specifica con un tasso es. 10/set o 5/g
- Il comportamento della riprogrammazione quando si assegna per la prima volta una risorsa ad un'attività, e quando la si assegna per la seconda volta, è diverso, a causa della politica di programmazione basata sulle risorse.



# Esempio Assegnazione MaxUnit

---

- Esempio:
- La Risorsa A ha 100% MaxUnit su un calendario di lavoro di 40h a settimana.
- Se viene assegnata all'attività B con 100% unità assegnazioni, A lavora 8h per 5 giorni alla settimana fino al completamento dell'attività. Se viene assegnata al 50%, A lavora 4 ore al giorno, e le altre 4 ore sono disponibili per essere assegnate ad altre attività.
- L'assegnamento dipende dal calendario di lavoro.
- Per le risorse che rappresentano gruppi di persone, ad es. con 400% di maxunit, 200% unità di assegnazione significano che 2 su 4 persone del gruppo sono assegnate a tempo pieno nel calendario di lavoro corrente (oppure 4 a metà tempo).



# Fattori che influenzano la programmazione

---

- Durata: il tempo stimato per una attività. Cambiare la durata impatta sulla data finale dell'attività.
- Dipendenze tra attività: il tipo di link tra predecessore e successore (FS, FF, SS...)
- Sovrapposizioni e Ritardi: il tempo di anticipo e ritardo tra le dipendenze di attività
- Vincoli: inflessibili o moderati. ASAP e ALAP consentono una programmazione più flessibile.
- Risorse Assegnate Inizialmente: Servono a conteggiare lo sforzo in ore a partire dalla durata programmata inizialmente.
- Risorse Aggiuntive: Successive assegnazioni diminuiscono proporzionalmente la durata (principio dell'assegnazione basata sulle risorse). Lo sforzo rimane invariato.



# Fattori che influenzano la programmazione

---

- Unità di Assegnamento: una risorsa assegnata parzialmente completa l'attività in un tempo maggiore.
- Lavoro: a parità di assegnamento aumentare il lavoro corrisponde ad aumentare la durata.
- Calendario di lavoro: includendo orari straordinari, la programmazione si accorcia.
- Programmazione basata sulle risorse: Aggiungendo risorse, la durata si comprime. Togliendone, si allunga.
- Tipo di Attività: a unità fisse, a lavoro fisso, a durata fissa: questo influenza il comportamento della riprogrammazione quando si cambia il lavoro, le unità o la durata.



# Programmazione Basata sulle Risorse

---

- Quando una risorsa viene assegnata ad una attività, Project calcola la quantità di lavoro assegnata alla risorsa. Questo succede al momento del primo assegnamento di una risorsa all’attività (prima di adesso, all’attività era stata assegnata solamente una durata).
- **Il calcolo iniziale del lavoro rimane invariato successivamente, indipendentemente dalle risorse che sono successivamente aggiunte.** La durata viene invece ricalcolata in questo modo:

$$\text{Durata} = \text{Lavoro} / \text{Unità}$$

- La Programmazione Basata sulle Risorse, dunque, mantiene il lavoro costante e riprogramma la durata in relazione alle unità di assegnamento.
- Se questo metodo non è appropriato (ad es. quando si aggiunge una risorsa di supervisione oppure una macchina utensile all’attività), questo metodo di programmazione può essere disattivato.
- La programmazione basata sulle risorse entra in causa solo sugli assegnamenti successivi, non al primo assegnamento.  
Il primo assegnamento viene usato per calcolare la quantità di lavoro da collegare all’attività.



# Tipi di Attività

---

- Indica se occorre tenere costanti la durata, il lavoro o le unità nella relazione  $D=L/U$ . Uno dei termini è fisso, uno è il parametro dato, e l'altro viene ricalcolato.
- Quando uno di questi termini viene modificato, uno degli altri due viene ricalcolato secondo il tipo di attività.
- Con la programmazione basata sulle risorse, il lavoro rimane costante indipendentemente dal tipo di attività.

Modify	Fixed Unit	Fixed Duration	Fixed Work
Units	Duration	Work	Duration
Duration	Work	Work	Units
Work	Duration	Units	Duration



# Informazioni Effettive

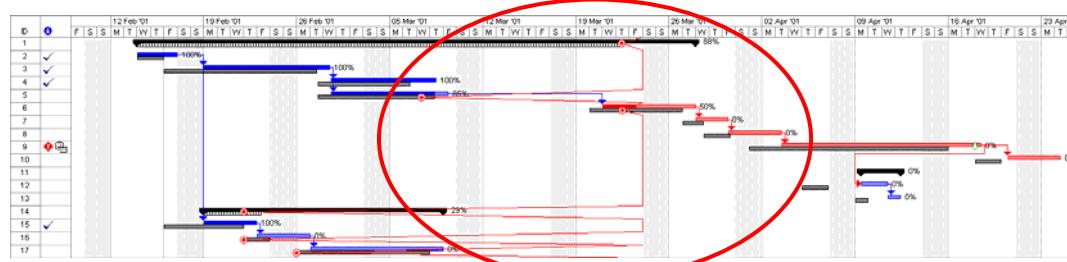
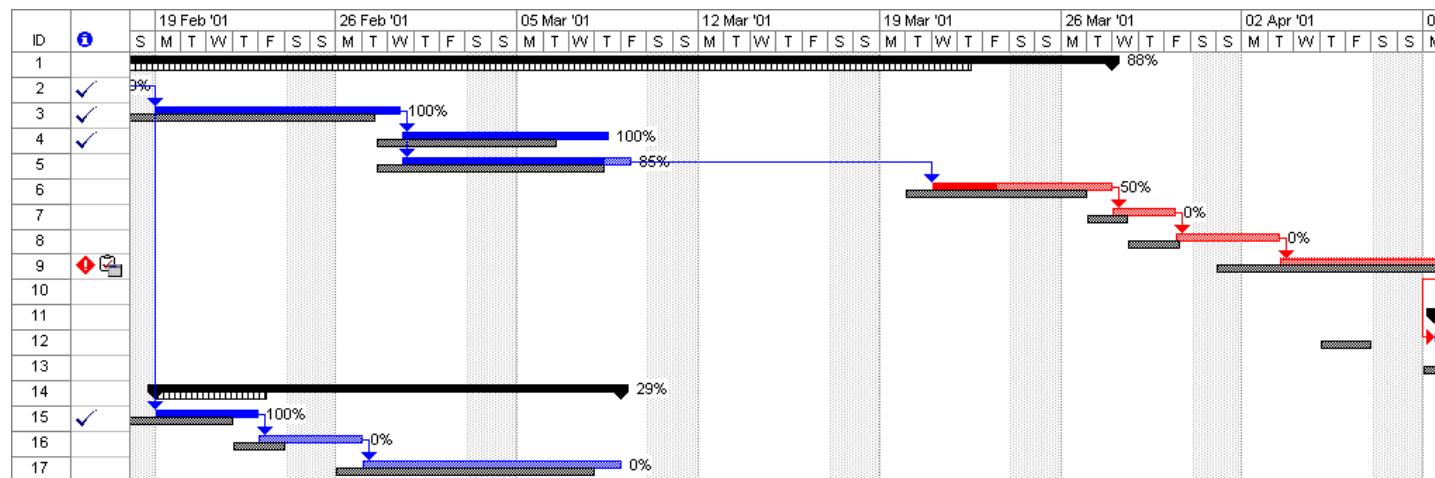
---

- Sono raccolte ed aggiornate le seguenti informazioni:
  - Date effettive di inizio e di fine delle attività
  - Percentuale effettiva di completamento di ciascuna attività
  - Durata effettiva dell'attività
  - Lavoro effettivo dell'attività
  - Costo effettivo dell'attività
- Le informazioni sono raccolte dal manager, oppure da rapporti scritti dai membri del team, attraverso ispezioni, controlli, rapportini di lavoro. Le funzionalità di Workgroup di Project aiutano nella fase di raccolta (e-mail e Project Central).
- I dati dovrebbero essere basati su fatti certi e su risultati, non solamente sul tempo trascorso. Come misurare oggettivamente l'avanzamento dell'attività software?



# Analisi delle Varianze

- Il Gantt consente di confrontare la programmazione corrente con la baseline e le linee di avanzamento:



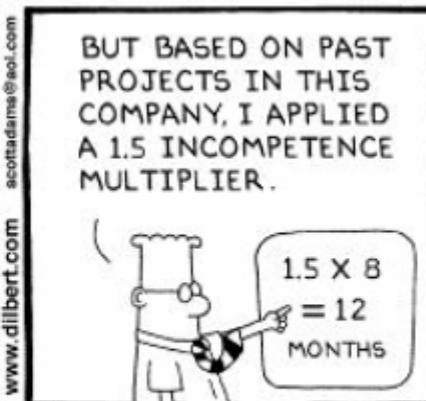
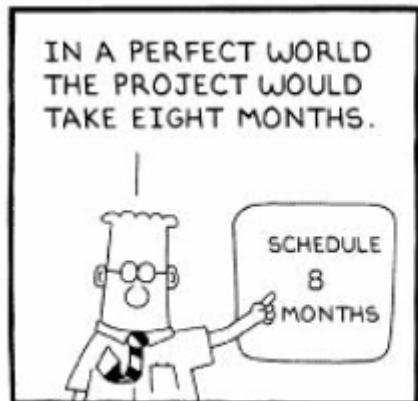
# Varianze sulla baseline

---

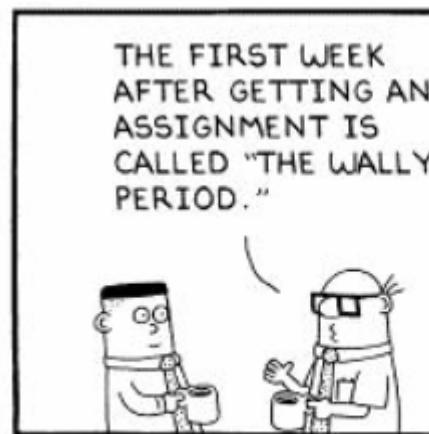
- Una *Varianza* è la differenza tra i dati pianificati e i dati di baseline. L'analisi delle varianze si può fare solo quando si ha una baseline. L'analisi comprende il controllo di:
  - Attività che partono o finiscono in ritardo
  - Attività che richiedono più o meno lavoro di quanto pianificato
  - Attività che stanno procedendo più lentamente di quanto pianificato
  - Attività che costano più di quanto pianificato rispetto all'effettivo lavoro compiuto
  - Risorse che non stanno lavorando il numero di ore programmato
- Le linee di avanzamento sono un modo visivo per determinare quali attività sono in anticipo o in ritardo rispetto alla programmazione.
- La finestra di Statistiche fornisce un rapido riepilogo sullo stato del progetto.
- Nella vista Gantt Verifica, con la Tabella delle Varianze, dei Costi o del Lavoro, è possibile confrontare visivamente i valori tra previsione e programma nella colonna della Varianza.



# Per chiudere con ironia...



Copyright © 2002 United Feature Syndicate, Inc.



Copyright © 2002 United Feature Syndicate, Inc.



# Struttura Deliverable 2

---

## ■ Deliverable 2: Dalla Schedulazione alla Implementazione

- Valutazione degli effort sui casi d'uso da implementare
  - Breve stima dei casi d'uso, utilizzando un criterio a scelta (es. FP, gg/uomo...)
- GANTT di progetto sulle risorse del gruppo
  - Gantt indicante attività, milestone, risorse assegnate, previsione conclusione progetto
  - Indicare anche attività delle ABS collaterali, non solo sviluppo
- Implementazione di uno (o più) casi d'uso a scelta mediante J2EE / jBPM, codice (perlopiù) funzionante.
  - Presentare codice sorgente in formato .zip
  - La demo di come gira il codice sarà presentata durante l'esame



# Bibliografia di Project Management

---

- Roger Pressman, *Principi di Ingegneria del Software*, McGrawHill
- F. Brooks, *The Mythical Man Month*, Prentice Hall
- Ann Lowery, *Managing Projects with Project 2000*, John Wiley
- Alberto Nepi, *Introduzione al Project Management*, Guerini e Ass.
- Bracchi - Motta, *Progetto di Sistemi Informativi*, ETAS



# Bibliografia del Corso

---

- Testi di studio della sintassi UML:
  - Scott, K. – *UML Explained*
  - Cap. 3 *UML Notation Guide*: <http://www.omg.org/uml>
- Testi di riferimento:
  - Maciaszek, L. – *Sviluppo di sistemi informativi con UML (Analisi dei Requisiti e Progetto di Sistema)*, Addison-Wesley
  - Eriksson, M., Penker – *Business Modeling with UML* – Addison-Wesley
  - Booch G., Rumbaugh J., Jacobson I. – *The Unified Modeling Language User Guide* – Addison-Wesley
  - Jacobson I. – *Object Oriented Software Engineering. A Use Case Driven Approach.* – Addison-Wesley
  - Watkins, J. - *Testing IT Software Testing Process*, Cambridge
  - Garmus, D. – *Function Point Analysis* – Addison-Wesley

