# Sviluppo delle applicazioni software

Docenti: Viviana Bono, Giovanna Petrone, Claudia Picardi, Gianluca Torta

{...}@di.unito.it

- **CFU**: 9
- **Ore**: *40 (teoria) + 50 (laboratorio)*

*Grazie a Simona Bernardi (Universita` di Saragoza, Spagna), che ha messo a disposizione il suo materiale*

# Bibliografia

- Processo Unificato
  - **Larman C.:** "*Applicare UML e i Pattern: Analisi e progettazione orientata agli oggetti*" Pearson-Prentice Hall, terza edizione (testo di riferimento)
  - **Jacobson I., Booch G., Rumbaugh J.**: "*The Unified Software Development Process*" Addison Wesley. (libro degli ideatori di UP)
- Unified Modeling Language (UML)
  - **Fowler M.**: "*UML Distilled*", Terza edizione italiana, Pearson-Addison Wesley. (guida minima all'UML)
  - **Bennet S., Skelton J., Lunn K.:** *"Introduzione a UML",* Collana Schaum's, McGraw-Hill (esercizi su UML)
  - **OMG:** http://www.omg.org (documentazione on-line)
- Design patterns
  - **Gamma, Helm, Johnson, Vlissides**, "*Design Patterns",* Prima edizione italiana, Pearson Education Italia-Addison Wesley.

# Materiale sul corso

- Sito I-learn:

`http://informatica.i-learn.unito.it/course/view.php?id=716`

# Obiettivi di questa parte del corso

- Conoscere una metodologia di sviluppo del software (Unified Process)
- Imparare  ad usare UML nell'ambito di UP.
- Conoscere uno strumento CASE a supporto della notazione UML.
- Imparare  a progettare usando i pattern di progettazione e pattern architetturali.

# Organizzazione

- Ingegneria del Software con metodologie O-O
  - Unified Process
  - UML
- Ingegneria dei requisiti
- Ingegneria della progettazione con pattern (design/architectural pattern)
- Uso dello strumento CASE Virtual Paradigm in lab.
- Progetto di uno studio di caso

# Lezione I

- Introduzione a UP
- Prima fase: Ideazione
- Requisiti evolutivi
- Casi di studio

# Ingegneria del software: problematiche (IEEE)

- Strategie sistematiche, a partire da richieste formulate dal committente, per lo sviluppo, esercizio e manutenzione del software
- Studio ed applicazione di tali strategie

# Standard

- Esistono molte organizzazioni che si preoccupano di stabilire degli standard di processi o di prodotti per l'industria del software
- Lo scopo è migliorare la qualità dei prodotti software e dei processi di produzione
- Standard del software
  - Standard IEEE – metodologie per lo sviluppo del sw
  - Standard OMG per UML, CORBA
  - Standard W3C, per tecnologie WEB
  - ….

# Standard OMG: UML

- Object Management Group è una organizzazione internazionale che raccoglie i principali vendor di sw
- UML è uno dei suoi standard più conosciuti
- UML è una notazione visuale per la specifica del sw
- Un modello UML è costituito da un insieme di diagrammi correlati, ciascuno dei quali descrive una “vista” del sistema
- Non è un processo di sviluppo
- Viene definito mediante un meta-modello

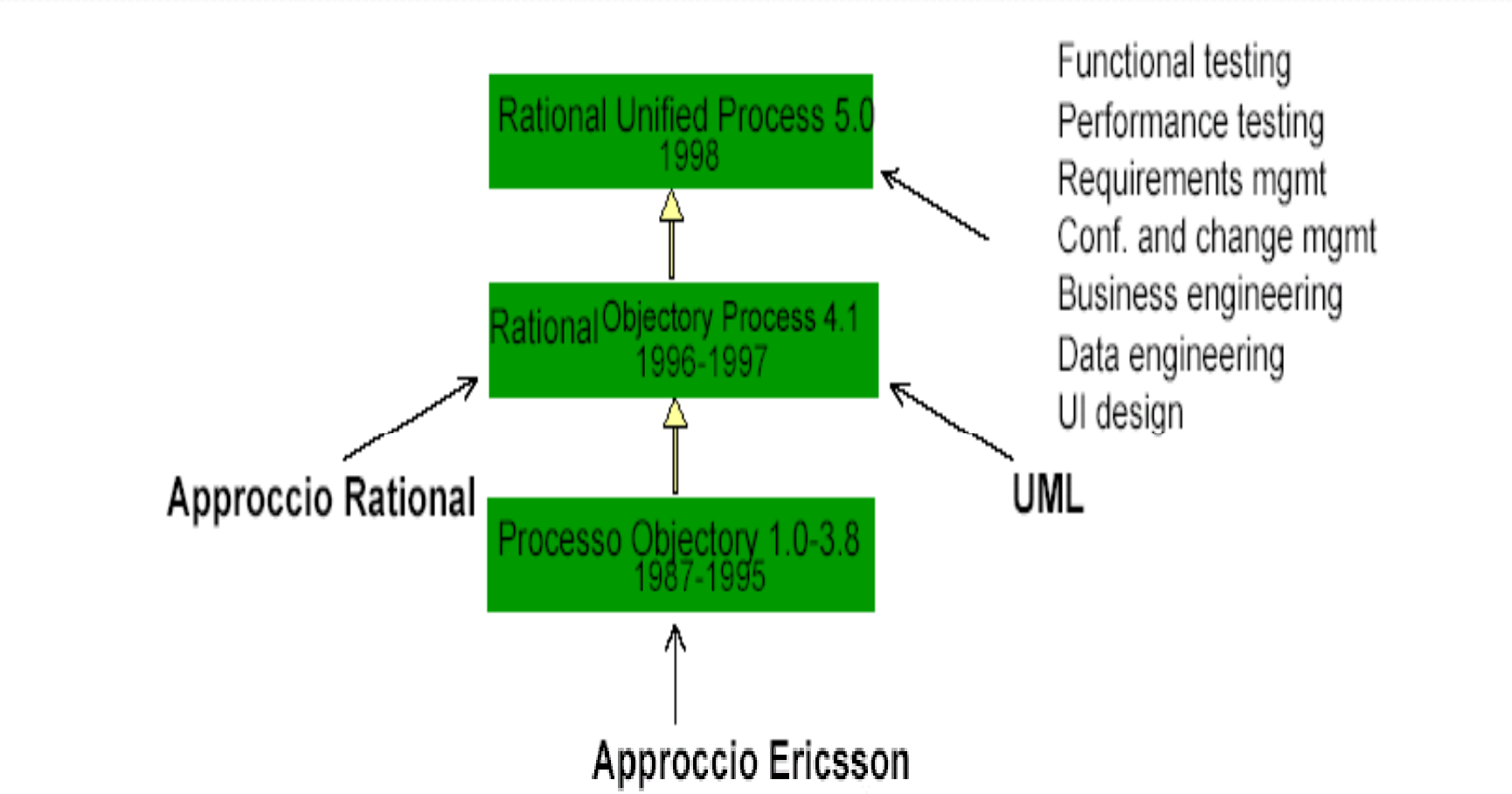
# Origini di UML/(R)UP

- Inizio degli anni ’90 tre metodi di sviluppo del sw
  - Metodo Booch (Grady Booch)
  - OMT (Jim Rumbaugh)
  - Fusion/OOSE (Ivan Jacobson) – ’95 arriva a Rational

’94 creano Rational Sw Co.

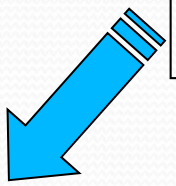
- 1994-95 Unified Modeling Language (v0.8) e RUP
- 1997 UML v1.1 standard OMG
- ….
- 2001 UML v1.4
- 2003 UML v1.5
- 2004 Standard ISO/IEC 19501
- 2007 UML 2.1
- 2009 UML 2.2

# Evoluzione (R)UP



# UML e UP

Booch, Rumbaugh, Jacobson



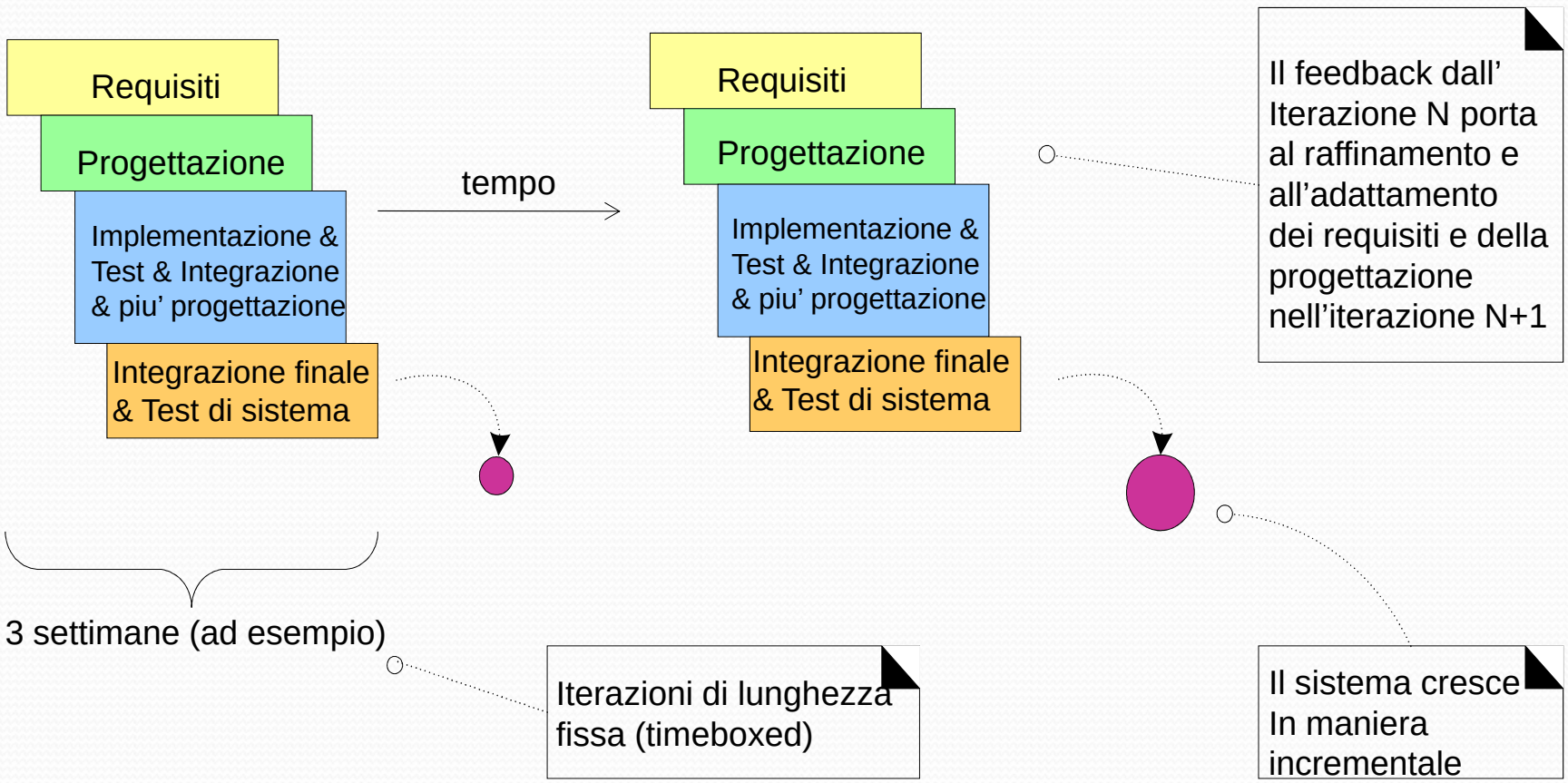
UML
(Unified Modeling Language)
• notazione per specificare un sistema software
• standard OMG (Object Management Group) dal 1997

UP (Unified Process)
• non è uno standard
• è un processo di sviluppo che utilizza UML
• versione commerciale (IBM-Rational): RUP (Rational Unified Process)

# Unified Process

- E` uno schema generale di processo (framework) che deve essere adattato a diversi tipi di progetti.
- Caratteristiche principali
  - Iterativo e incrementale
  - Iterazioni iniziali guidate
    - dal rischio,
    - dal cliente e
    - dall'architettura
  - Flessibile e può essere applicato usando un approccio "agile"

# UP: Iterativo e incrementale

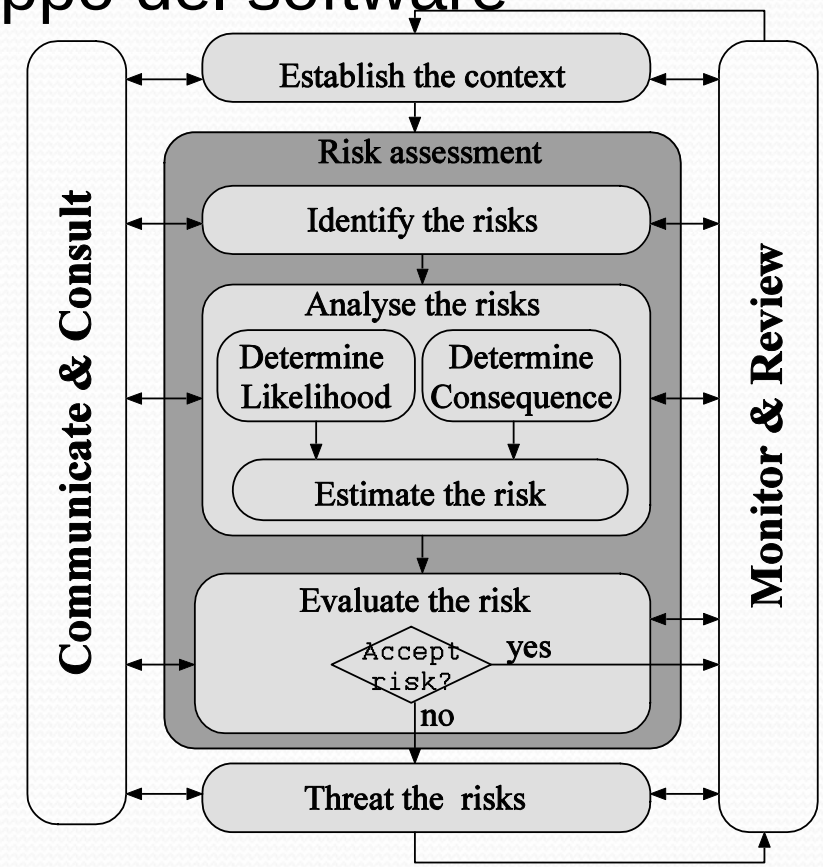


# UP: iterazioni iniziali

- Gli obiettivi principali delle iterazioni iniziali sono scelti per
  - Identificare e ridurre i rischi maggiori
  - Costruire e rendere visibili le caratteristiche piú importanti per il cliente
  - Stabilizzare il nucleo dell’architettura software

# Cos'e' il rischio?

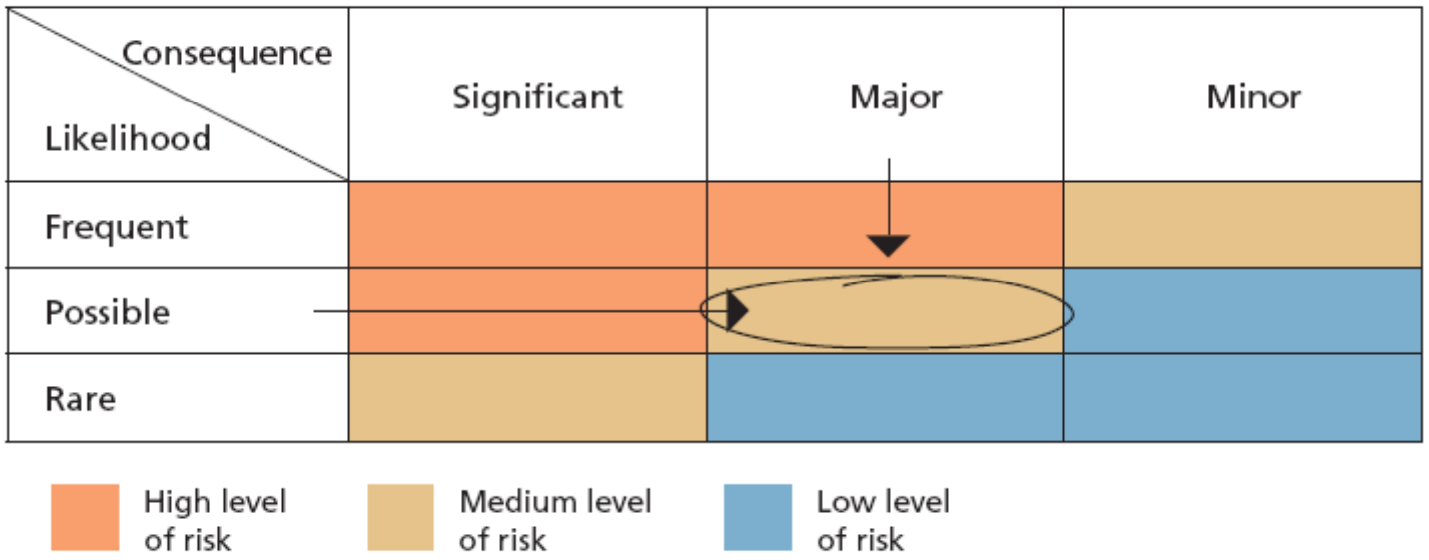
- Esistono ''processi'' di gestione (vari standard) del rischio associato allo sviluppo del software
- AS/NZS 4360 standard



# Criteri di rischio (esempi)

| Risk criterion | Objective |
|---|---|
| Safety | Safety must be upheld at all times. No injuries or fatalities will be accepted |
| Financial impact | Project costs should remain within allocated budget |
| Media exposure | The project must ensure that the reputation of the business is protected from negative media exposure |
| Timing | The project must be completed within the contractual timeframe |
| Staff management | The project must utilise existing staff skills. Where a particular skill set is not available, sub-contracting may be considered |
| Environment | The project must operate within requirements of environmental legislation and be consistent with the business's environmental commitment |

# Valutazione del rischio

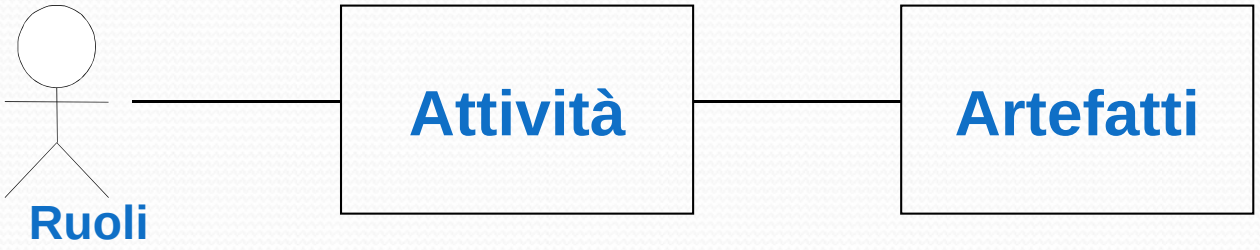


# UP e` agile

- UP incoraggia l'uso di pratiche agili introdotte da altre metodologie:
  - Iterazioni corte e timeboxed
  - Raffinamento di piani, requisiti, progettazione
  - Gruppi di lavoro auto-organizzati che si coordinano in riunioni regolari (Scrum)
  - Programmazione a coppie e sviluppo guidato dai test (XP=eXtremeProgramming)
  - Modellazione agile: l'obiettivo è la comprensione del sw piuttosto che la documentazione dello stesso

# Cosa c’è in UP

- Un’organizzazione del piano di progetto per fasi sequenziali
- Indicazioni sulle attività da svolgere nell’ambito di discipline e sulle loro inter-relazioni
- Un insieme di ruoli predefiniti
- Un insieme di artefatti da produrre



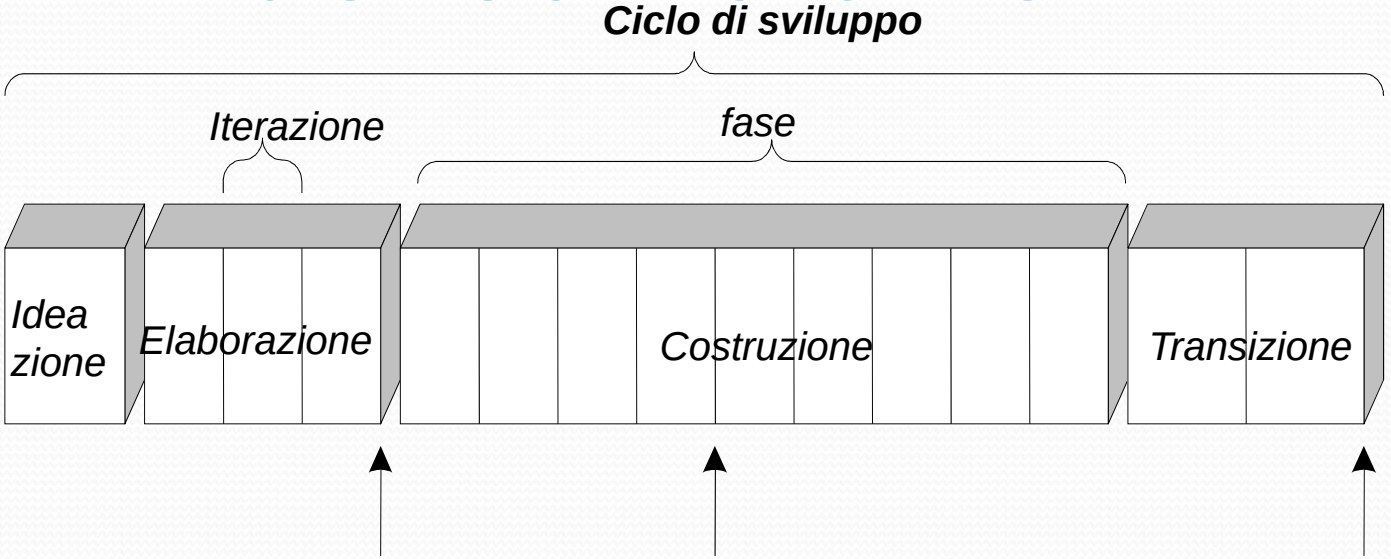
# Fasi di UP (I)

- Ideazione (Inception)
  - Visione approssimata, studio economico, portata, stima approssimativa dei costi e tempi di sviluppo
  - Milestone: **Obiettivi**
- Elaborazione (Elaboration)
  - Visione raffinata, implementazione iterativa del nucleo dell´architettura, risoluzione dei rischi maggiori, identificazione della maggior parte dei requisiti e della portata, stime più realistiche.
  - Milestone: **Architetturale**

# Fasi di UP (II)

- Costruzione (Construction)
  - Implementazione iterativa degli elementi che rimangono, più facili e a rischio minore, preparazione al rilascio
  - Milestone: **Capacità operazionale**
- Transizione (Transition)
  - Beta test, rilascio prodotto, addestramento utenti
  - Milestone: **Rilascio prodotto**

# UP: fasi ed iterazioni




*milestone*

Fine di un´iterazione in cui si verifica una decisione o valutazione significativa

*release*

Sottoinsieme stabile ed eseguibile del prodotto finale.La fine di ogni itera-zione produce una release minore

*incremento*

La differenza (delta) tra le release di 2 iterazioni successive

.

*produzione finale release*

A questo punto il sistema viene rilasciato e consegnato ai clienti

# Discipline UP

- Le attività in UP si eseguono nell´ ambito di discipline (Core Workflow)
- Una disciplina è un insieme di attività ed artefatti (work product, in RUP) – come ad es. schemi di BD, documenti, modelli, codice – in un’area specifica

# Discipline ingegneristiche

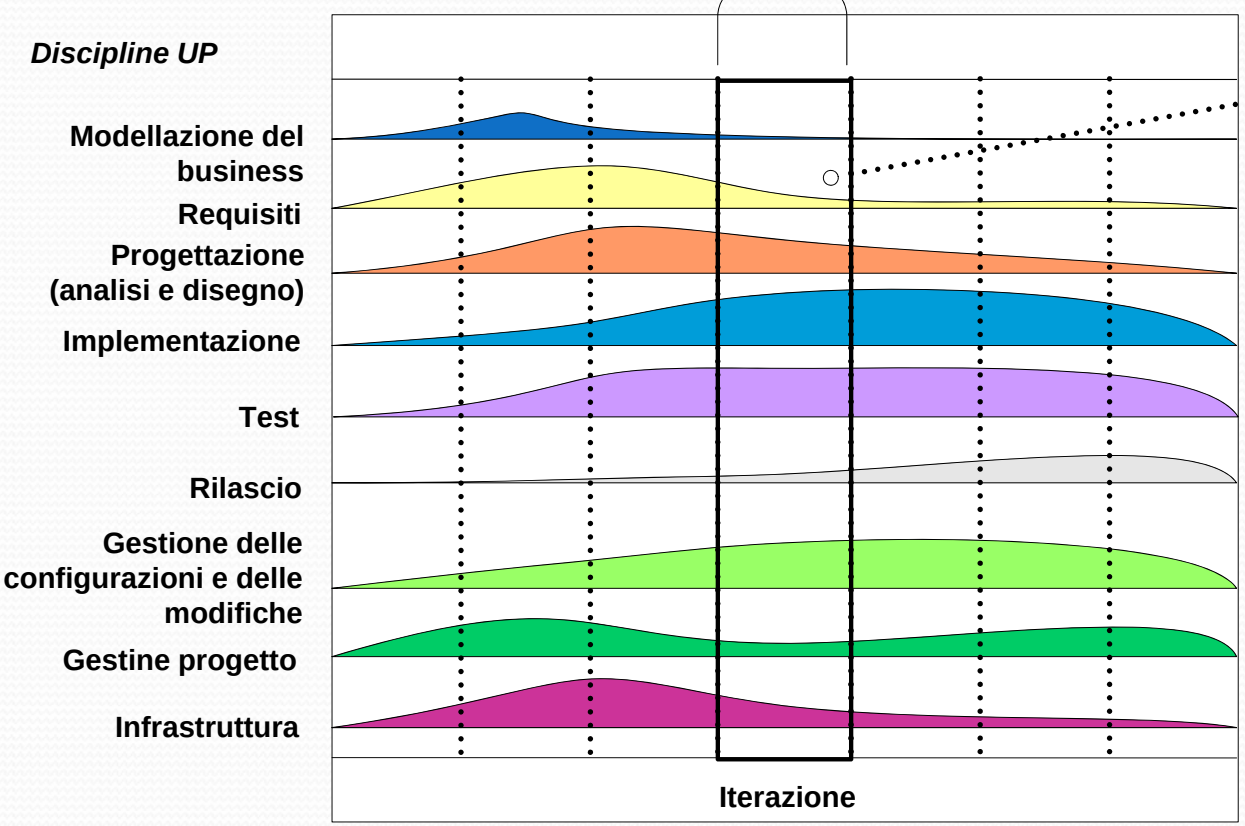
- Modellazione del business
  - Attività che modellano il dominio del problema ed il suo ambito
- Requisiti
  - Attività di raccolta dei requisiti del sistema
- Progettazione (analysis & design)
  - Attività di analisi dei requisiti e progetto architetturale del sistema
- Implementazione
  - Attività di progetto dettagliato e codifica del sistema, test su componenti
- Test
  - Attività di controllo di qualità, test di integrazione e di sistema
- Rilascio (Deployment)
  - Attività di consegna e messa in opera

# Discipline di supporto

- Gestione delle configurazioni e del cambiamento
  - Attività di manutenzione durante il progetto
- Gestione progetto
  - Attività di pianificazione e governo del progetto
- Infrastruttura (Environment)
  - Attività che supportano il team di progetto, riguardo ai processi e strumenti utilizzati

# UP: discipline ed iterazioni

**Un´iterazione di 4 settimane, ad es. su un miniprogetto che comprende il lavoro svolto nelle varie discipline e che termina con un eseguibile stabile.**




**Sebbene L´iterazione includa lavoro nella maggiorparte delle discipline, impegno e l´enfasi cambiano nel tempo. Questo è solo un esempio**

# Fasi e discipline

- Attenzione!
  - Le fasi **sono sequenziali** e la fine di una fase corrisponde ad una milestone
  - Le discipline (tipologie di attività) **non sono sequenziali** e si eseguono nel progetto in ogni iterazione
  - Il numero di iterazioni dipende dalla decisione del manager di progetto e dai rischi del progetto

# Uso di UML in UP

- UP usa solo UML come linguaggio di modellazione (ad esempio, non si usano i Data Flow Diagram)
- I diagrammi UML si usano con variabilità: se un diagramma non è necessario non si usa, però tale scelta si indica esplicitamente. Bisogna personalizzare UP prima di applicarlo.
- I diagrammi si usano in UP seguendo le caratteristiche di iterazione ed incremento (incrementi definiti su uno stesso diagramma)
- UP dice quando usare un diagramma

# Adattamento del processo

- In UP quasi tutto (tra artefatti e pratiche) è opzionale, eccetto che lo sviluppo iterativo e guidato dal rischio, la verifica continua della qualità e naturalmente il codice.
- La scelta delle pratiche e artefatti UP per un progetto si riassume in un documento chiamato Scenario di Sviluppo (artefatto della disciplina Infrastruttura)

# Scenario di sviluppo (esempio)

| Disciplina | Pratica | Artefatto<br>Iterazioni → | Id.<br>(I1) | Elabor.<br>(E1..Ei) | Constr.<br>(C1..Cj) | Trans.<br>(T1..Tk) |
|---|---|---|---|---|---|---|
| Modellazione del business | Modellazione agile, workshop dei requisiti | Modello di dominio | | Inizio | | |
| Requisiti | Workshop dei requisiti, esercizio sulla visione | Modello UC<br>Visione<br>Specifica Suppl.<br>Glossario | In.<br>In.<br>In.<br>In. | Raffin.<br>Raffin.<br>Raffin.<br>Raffin. | | |
| Progettazione | Modellazione agile e sviluppo guidato dai test | Modello progetto<br>Doc. Architett.sw<br>Modello dei dati | | Inizio<br>Inizio<br>Inizio | Raffin.<br><br>Raffin. | |
| Implementazione | Sviluppo guidato dai test, programmazione a coppie, integrazione continua, standard di codifica | ... | | | | |
| Gestione progetto | Gestione progetto agile, riunioni Scrum giornaliere | ... | | | | |
| ... | | ... | | | | |

# Non si è capito lo sviluppo iterativo o UP se....

- Si cerca di definire tutti i requisiti del sw prima di iniziare la progettazione o l`implementazione
- Si dedicano gg o settimane a modellare con UML prima di iniziare a programmare
- Si pensa: ideazione = requisiti, elaborazione = progettazione, costruzione = implementazione (cioè, si adotta l´approccio a “cascata”)
- Si pensa che l’obiettivo dell’elaborazione sia quello di definire in maniera completa e dettagliata i modelli, che verranno tradotti in codice durante la costruzione
- Si pensa che la durata adeguata per una iterazione siano 3 mesi al posto di 3 settimane
- Si cerca di pianificare il progetto nei dettagli dall’inizio fino alla fine, e di prevedere in maniera speculativa tutte le iterazioni e cosa deve accadere in ognuna di esse.

# Lezione I

- Introduzione a UP
- Prima fase: Ideazione
- Requisiti evolutivi
- Casi di studio

# Ideazione (Inception)

- Permette stabilire una visione comune e la portata del progetto (studio di fattibilità)
- Durante la ideazione:
  - Si analizzano circa il 10% dei casi d'uso in dettaglio
  - Si analizzano i requisiti non funzionali più critici
  - Si realizza uno studio economico per stabilire l'ordine di grandezza del progetto e la stima dei costi
  - Si prepara l'ambiente di sviluppo
- Durata: normalmente breve (primo workshop dei requisiti e pianificazione della prima iterazione dell'elaborazione)

# Artefatti nell'ideazione (I)

- Modello dei casi d'uso
  - Requisiti funzionali. Identificazione della maggior parte dei nomi dei casi d'uso, 10% descrizione dettagliata
- Specifiche supplementari
  - Altri requisiti (non funzionali)
- Visione e studio economico
  - Obiettivi e vincoli ad alto livello, studio economico, riassunto del progetto (executive summary)
- Glossario
  - Dizionario dei dati e terminologia del dominio

# Artefatti nell'ideazione (II)

- Lista dei rischi e piano di gestione dei rischi
  - Rischi ed idee per affrontarli
- Prototipi e  proof-of-concept
  - Chiarire la visione e verificare le idee tecniche
- Piano dell'iterazione
  - Cosa fare nella prima iterazione dell'elaborazione
- Piano delle fasi e piano di sviluppo del sw
  - Ipotesi sulla durata e sforzo dell'elaborazione
- Scenario di sviluppo
  - Personalizzazione di UP (passi ed artefatti)

Non sono troppi ?

- Si scelgono quelli che aggiungono valore al progetto
- Si completano parzialmente
- Sono preliminari ed approssimativi

# Lezione I

- Introduzione a UP
- Prima fase: Ideazione
- Requisiti evolutivi
- Casi di studio

# Requisiti evolutivi

- Capacità e condizioni alle quali il sistema deve essere conforme
- UP propone un insieme di best practice per gestire i requisiti che cambiano
- Modello FURPS+
  - Functional: caratteristiche comportamentali, capacità
  - Usability: fattori umani, help, documentazione
  - Reliability: frequenza dei guasti, capacità di riparazione
  - Performance: tempo di risposta, throughput, uso risorse
  - Supportability: adattamento, manutenzione, configurabilità
  - +: requisiti complementari e secondari (risorse limitate, linguaggi e hw, di interfaccia, legali)

# Lezione I

- Introduzione a UP
- Prima fase: Ideazione
- Requisiti evolutivi
- Casi di studio

# NextGen POS System [Larman]

- Un POS (point of sale) richiede lo sviluppo di software utilizzato tra l'altro per registrare vendite ed pagamenti, in negozi e supermercati. Comprende componenti hardware e software. Alcune funzionalità di servizio, ad es. il calcolo delle imposte, sviluppate da terzi e l'inventario devono interagire con il POS. Il sw POS deve essere tollerante ai guasti: ad es. se alcuni servizi non funzionano, il sw deve comunque permettere di registrare i pagamenti in modo che l'attività non si fermi.
- Il sw POS deve essere in grado di usare terminali diversi ed interfacce diverse, browser, PDA, touch screen, interfacce dedicate
- Poiché il sw verrà venduto a diversi clienti, è necessario pensare al grado di personalizzazione

# VolBank System [Bennet&al]

- La VolBank è un´organizzazione senza fini di lucro che si occupa di mettere in contatto volontari con persone e gruppi che hanno bisogno di assistenza di qualsiasi genere. Il nome VolBank nasce dall´ idea che le persone possono ´´depositare´´ il tempo che sono in grado di mettere a disposizione degli altri, così come una lista di capacità ed abilità che desiderano offrire.
- VolBank necessita di un sistema informatico per gestire la registrazione e l' abbinamento di volontari e bisognosi di aiuto, nonché la notifica degli abbinamenti alle varie parti in causa.
- Il sistema sw dovrà anche disporre di un collegamento con il web server di VolBank.