

# RUP e OpenUP

Giuseppe Calavaro

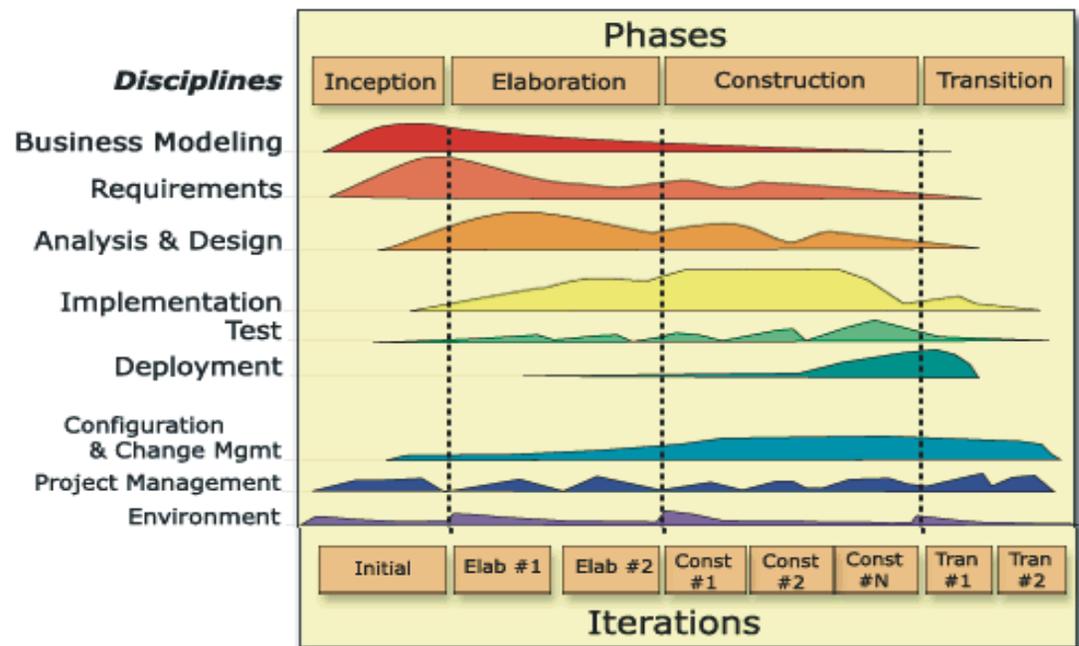
## II RUP – Rational Unified Process



- E' la sintesi metodologica dei metodi che hanno portato alla definizione dell'UML
- Ha gli stessi autori di UML
- Implementa le Best Practices dell'ingegneria del software
- E' un framework di processo che dovrebbe essere "tailored" dai team di sviluppo sulla base delle proprie esigenze

# Implementare le Best Practice

- Attraverso un processo iterativo completo e definito nelle sue fondamenta
- Con guide per le attività e gli elaborati da produrre
- Focalizzandosi sull'architettura
- Con Casi d'Uso che guidano la progettazione e l'implementazione
- Modelli che rappresentano le diverse caratteristiche salienti del sistema

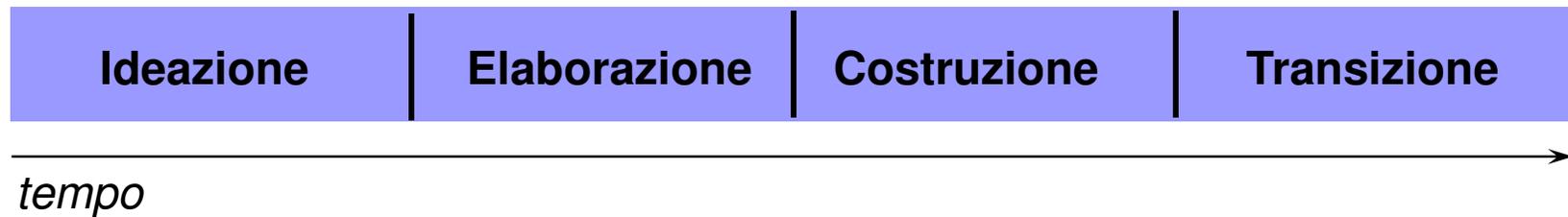


# Processo di sviluppo nell'Ingegneria del Software

- Un processo definisce CHI, fa COSA, COME e QUANDO, nell'esecuzione delle attività per raggiungere un obiettivo prefissato.



# Struttura del Processo UP – Le fasi del ciclo di vita



Lo Unified Process ha 4 fasi:

- **Ideazione**

- Si definisce l'ambito e portata del progetto

- **Elaborazione**

- Si specificano le caratteristiche (features) e la baseline dell'architettura

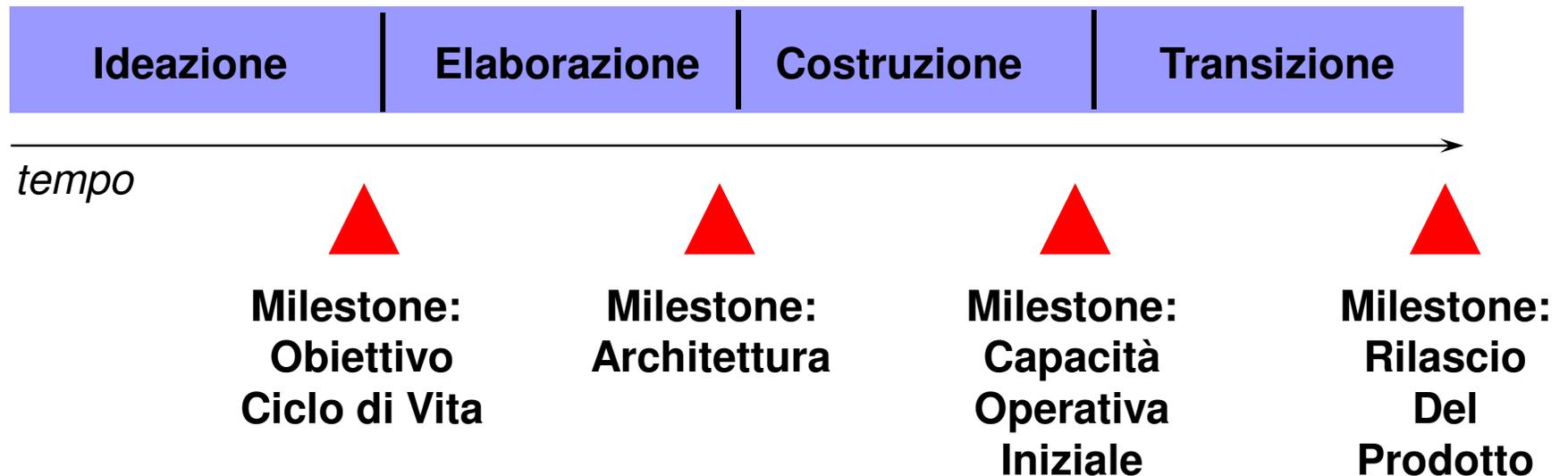
- **Costruzione**

- Si realizza il prodotto

- **Transizione**

- Si eseguono le attività necessarie per passare il prodotto agli utenti finali

# Milestone al termine delle fasi



- Milestone: Obiettivo Ciclo di Vita
  - Accordo su cosa è dentro e cosa è fuori dal progetto e sulla stima dei tempi e costi del progetto.
- Milestone: Architettura
  - Stabilità della vision del prodotto e dell'architettura
- Milestone: Capacità Operativa Iniziale
  - Maturità del prodotto per poter iniziare rilascio
- Milestone: Rilascio del prodotto
  - Conclusione del rilascio. Accettazione dal parte del cliente del prodotto rilasciato

# Fasi ed Iterazioni

| Ideazione              | Elaborazione             |                          | Costruzione         |                     |                     | Transizione            |                        |
|------------------------|--------------------------|--------------------------|---------------------|---------------------|---------------------|------------------------|------------------------|
|                        |                          |                          |                     |                     |                     |                        |                        |
| Iterazione Preliminare | Iterazione Architettural | Iterazione Architettural | Iterazione Sviluppo | Iterazione Sviluppo | Iterazione Sviluppo | Iterazione Transizione | Iterazione Transizione |

## Milestone Secondarie: Versioni Intermedie

- Le fasi sono divise in diverse iterazioni (eccetto la fase di Ideazione)
- Una **ITERAZIONE** è una sequenza di attività basate su un piano di sviluppo e dei criteri di valutazione che risultano in una nuova versione **eseguibile** (interna o esterna).

# Le Iterazioni

- Una iterazione è un miniprogetto durante il quale si passa attraverso tutte le discipline
  - Una iterazione produce sempre una nuova versione **eseguibile** del prodotto
- Una iterazione ha una durata fissa e costante (time-boxed)
  - Di conseguenza il numero di iterazioni per ogni fase dipende dalla durata delle fasi e quindi da quando si riesce a superare le rispettive milestone
- L'ammontare di requisiti da sviluppare in una iterazione cambia nel tempo, con l'esperienza del team ed il grado di maturazione del sistema che si sta sviluppando
  - Le prime iterazioni, tipicamente hanno pochi requisiti da implementare e servono anche a calibrare la cosiddetta velocità dell'iterazione
- Al termine di una iterazione si analizzano i feedback dal test dell'iterazione, i risultati ottenuti rispetto agli obiettivi generali di progetto, le richieste di cambiamento pervenute durante l'iterazione e si aggiornano il piano di progetto e definisce il piano dell'iterazione successiva
  - Il piano di una iterazione, definito all'inizio di questa, specifica i requisiti che saranno implementati

# Fase di Ideazione

## Obiettivi Principali

- Accordo tra tutte le parti interessate (stakeholder) su:
  - obiettivi di progetto per questo ciclo di vita,
  - il budget e
  - i tempi

## Risultati Attesi

- Un documento di visione
- Studio generale del modello dei casi d'uso
- Studio economico con tempi, costi, rischi e criteri di successo
- Un piano di progetto

## Milestone: Obiettivo Ciclo di Vita

- Gli stakeholder sono **d'accordo sull'ambito e portata del sistema, cosa è dentro e cosa è fuori, e sulla stima dei tempi e costi del progetto.**
- La comprensione dei requisiti è presentata dal modello dei casi d'uso con la specifica dei casi d'uso principali a livello di sommario (outline).
- Viene valutata la credibilità delle stime dei costi e tempi, le priorità assegnate, rischi, processo di sviluppo stabilito e di eventuali architetture di riferimento identificate.

# Fase di Elaborazione

## Obiettivi Principali

- Definire, convalidare e creare una baseline dell'architettura eseguibile
- Aggiornare la baseline del documento di visione
- Dimostrare che l'architettura identificata supporterà la visione di progetto con tempi e costi ragionevoli

## Risultati Attesi

- Una descrizione dell'architettura software validata dall'esecuzione del software realizzato
- Un modello dei casi d'uso che si ritiene abbastanza completa (all'80%)
- I requisiti supplementari descritti e compresi
- Lista dei rischi e piano di sviluppo

## Milestone: Architettura

- Si valuta la **stabilità della vision del prodotto e dell'architettura**.
- L'**architettura eseguibile** deve dimostrare che l'architettura sarà in grado di supportare le funzionalità del sistema ed esibire un comportamento corretto in termini di prestazioni, scalabilità e costi.
- Risolti i principali rischi di progetto, pianificazione e stime su cui tutti gli stakeholder si trovano ragionevolmente d'accordo.
- Gli stakeholder accettano la visione di prodotto ed il piano di progetto, inclusi i livelli di spesa.

### Architettura Eseguita

**Implementazione parziale del sistema che include le componenti più significative del sistema dal punto di vista architetturale**

# Fase di Costruzione

## Obiettivi Principali

- Completare versioni eseguibili del prodotto software
- Raggiungere una qualità adeguata il più rapidamente e realisticamente possibile
- Gestire e controllare le risorse, i tempi e costi di sviluppo

## Risultati Attesi

- Il prodotto software integrato sulle piattaforme stabilite
- Descrizione del sistema ed eventuali manuali utente

## Milestone:

### Capacità Operativa Iniziale

- Maturità del prodotto per **poter iniziare rilascio**
  - Per progetti su commessa, un sistema che si ritenga possa essere accettato dal committente
  - Per i prodotti venduti sul mercato, un sistema presentabile come versione Beta ai clienti esistenti
- Le parti interessate sono pronte alla transizione verso la comunità degli utenti

# Fase di Transizione

## Obiettivi Principali

- Preparare per la messa in esercizio del sistema
- Fare in modo che gli utenti siano non solo in grado di utilizzare il sistema
- Assicurarsi che l'intera comunità degli utenti conosca il nuovo sistema e ne apprezzi il valore

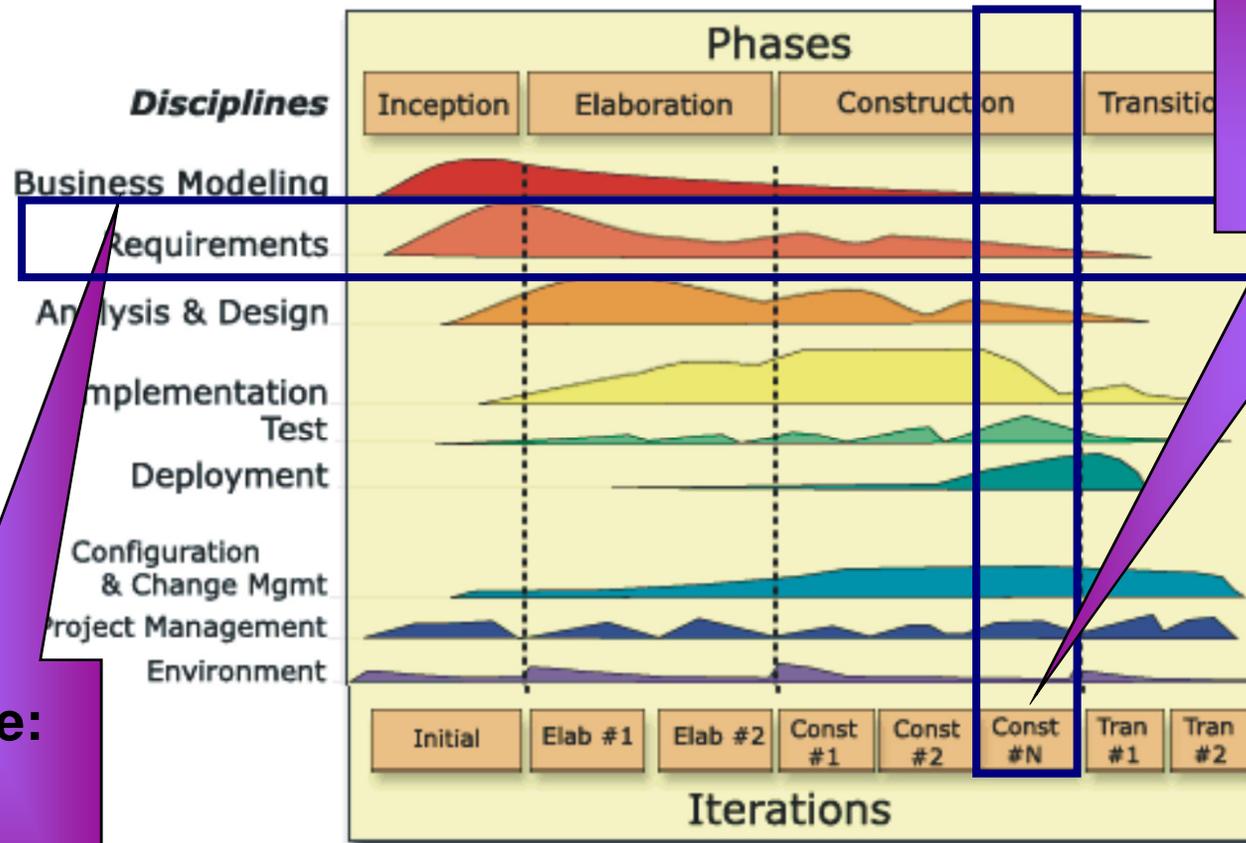
## Risultati Attesi

- Il sistema è pronto per entrare in esercizio ufficialmente
- Compiere eventuali migrazioni di dati dai sistemi precedenti
- Avviare le attività di Marketing

## Milestone: Rilascio del prodotto

- **Pronti per la messa in esercizio** ufficiale del sistema
- Eventuale accettazione ufficiale da parte del cliente del prodotto rilasciato

# Fasi e Discipline: Il processo in un colpo d'occhio



In una iterazione si eseguono tutte le discipline.

Discipline: gruppi di attività logicamente correlate.

# Le discipline in dettaglio

## Sei discipline principali

1. Business Modeling
2. Requisiti
3. Analisi e Progettazione
4. Implementazione
5. Test
6. Rilascio

## Tre discipline di supporto

1. Controllo Configurazione e Cambiamenti
2. Gestione di Progetto
3. Gestione degli Ambienti e Strumenti di sviluppo e Run-Time

# Business Modeling

## ■ Obiettivi

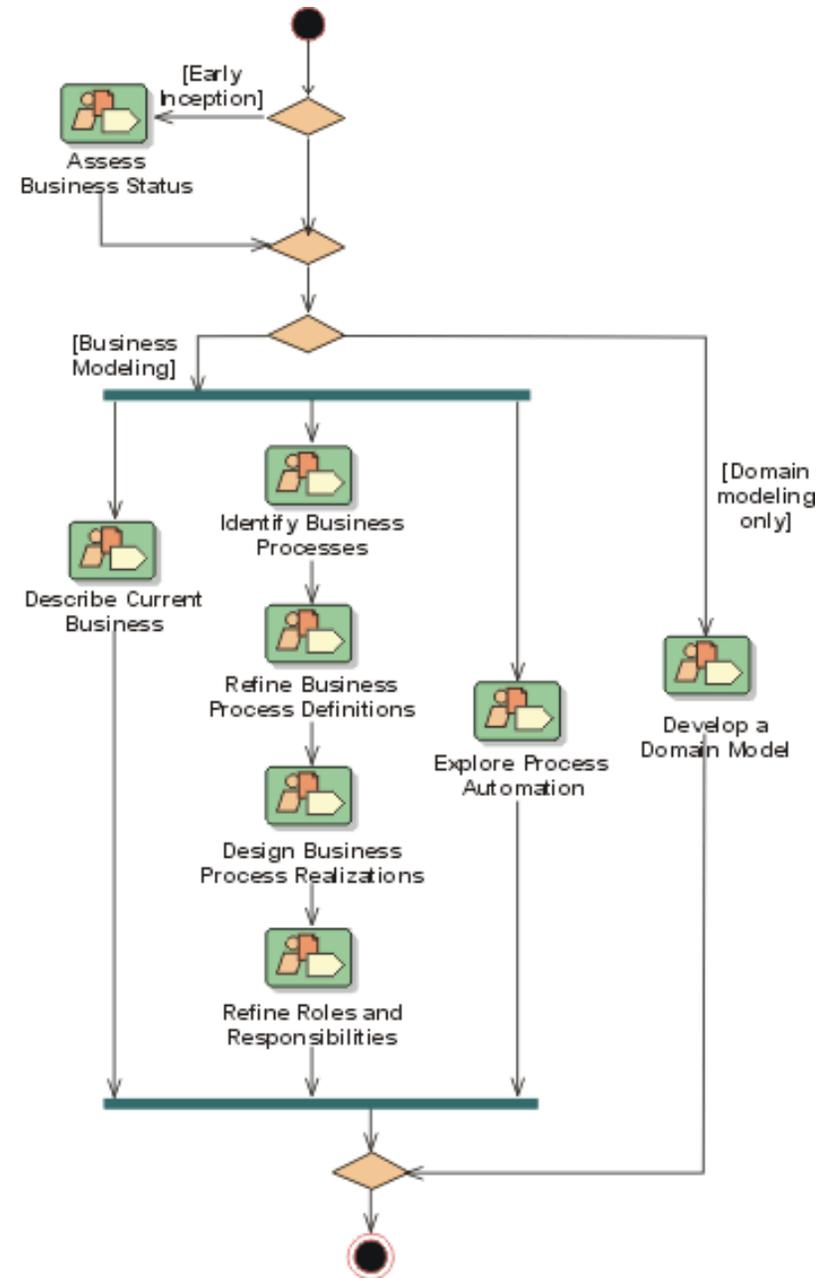
- Comprendere la struttura e la dinamica dell'organizzazione in cui si inserisce il sistema software
- Derivare i requisiti del sistema a sostegno degli obiettivi di business

## ■ Ruoli Principali

- Analista di business
  - Definisce il modello di business
- Progettista di business
  - Descrive i vari processi di business

## ■ Elaborati Principali

- Modello dei casi d'uso di business
- Modello degli oggetti di business
- Specifiche supplementari
- Glossario



# Disciplina: Requisiti

## ■ Obiettivi

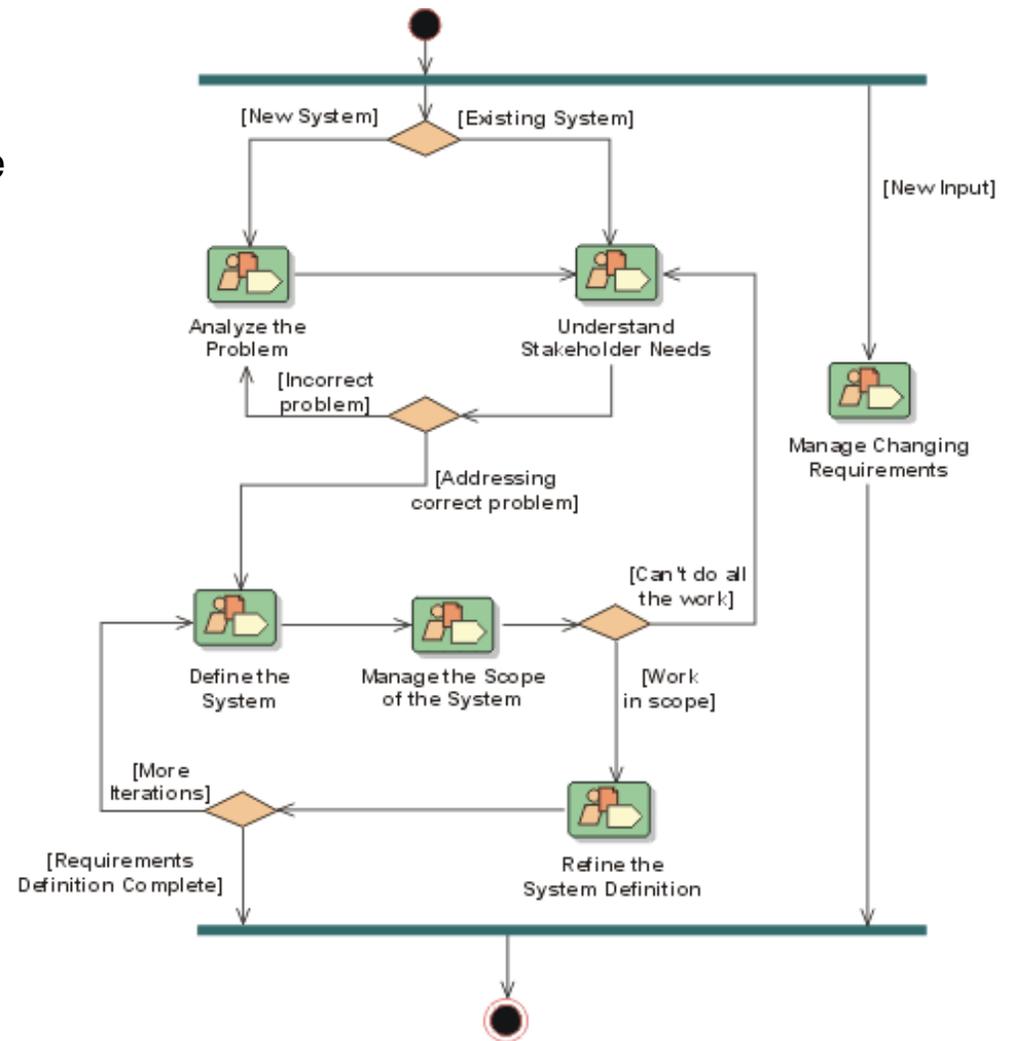
- Comprensione e comunicazione dei requisiti del sistema, sia funzionali che non funzionali
- Gestione dei requisiti e dei loro cambiamenti
- Supporto alla pianificazione dello sviluppo per ogni iterazione

## ■ Ruoli Principali

- Analista e Progettista di sistema
  - Definiscono il modello dei requisiti e sviluppano le specifiche degli elementi del modello
- Architetto, Grafico/Progettista UI

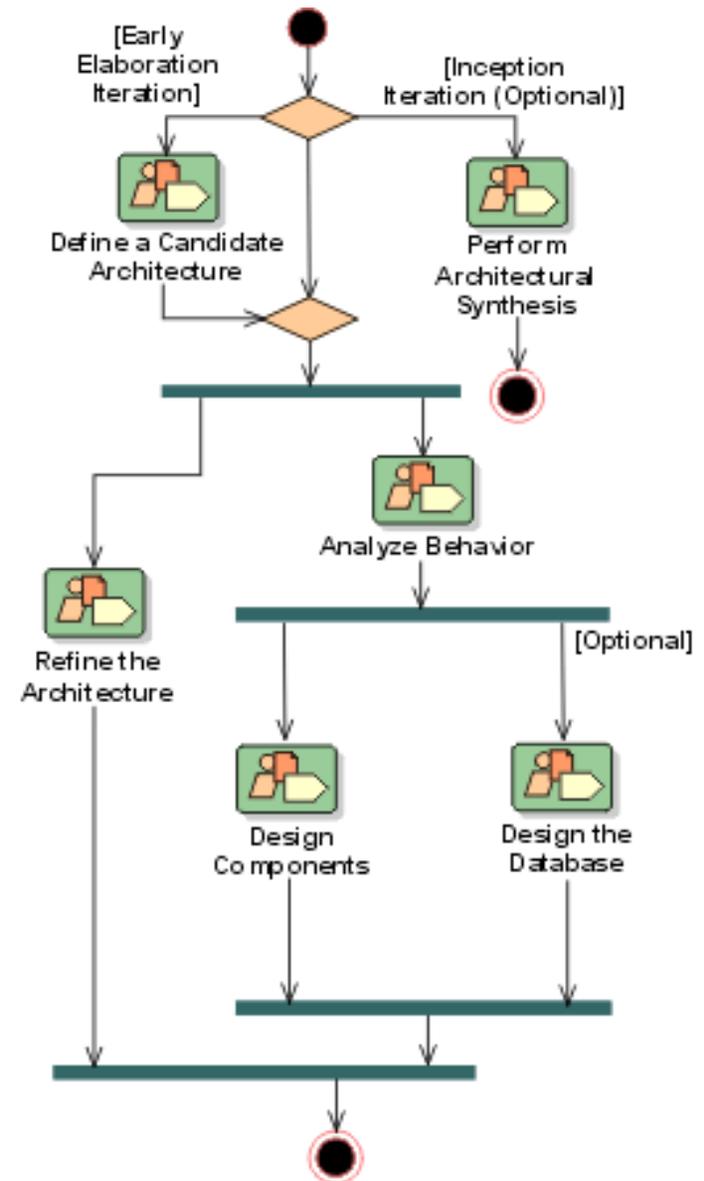
## ■ Elaborati Principali

- Documento di Vision
- Modello dei casi d'uso
- Specifiche dei casi d'uso
- Specifiche supplementari
- Attributi dei requisiti e matrici di tracciabilità
- Glossario



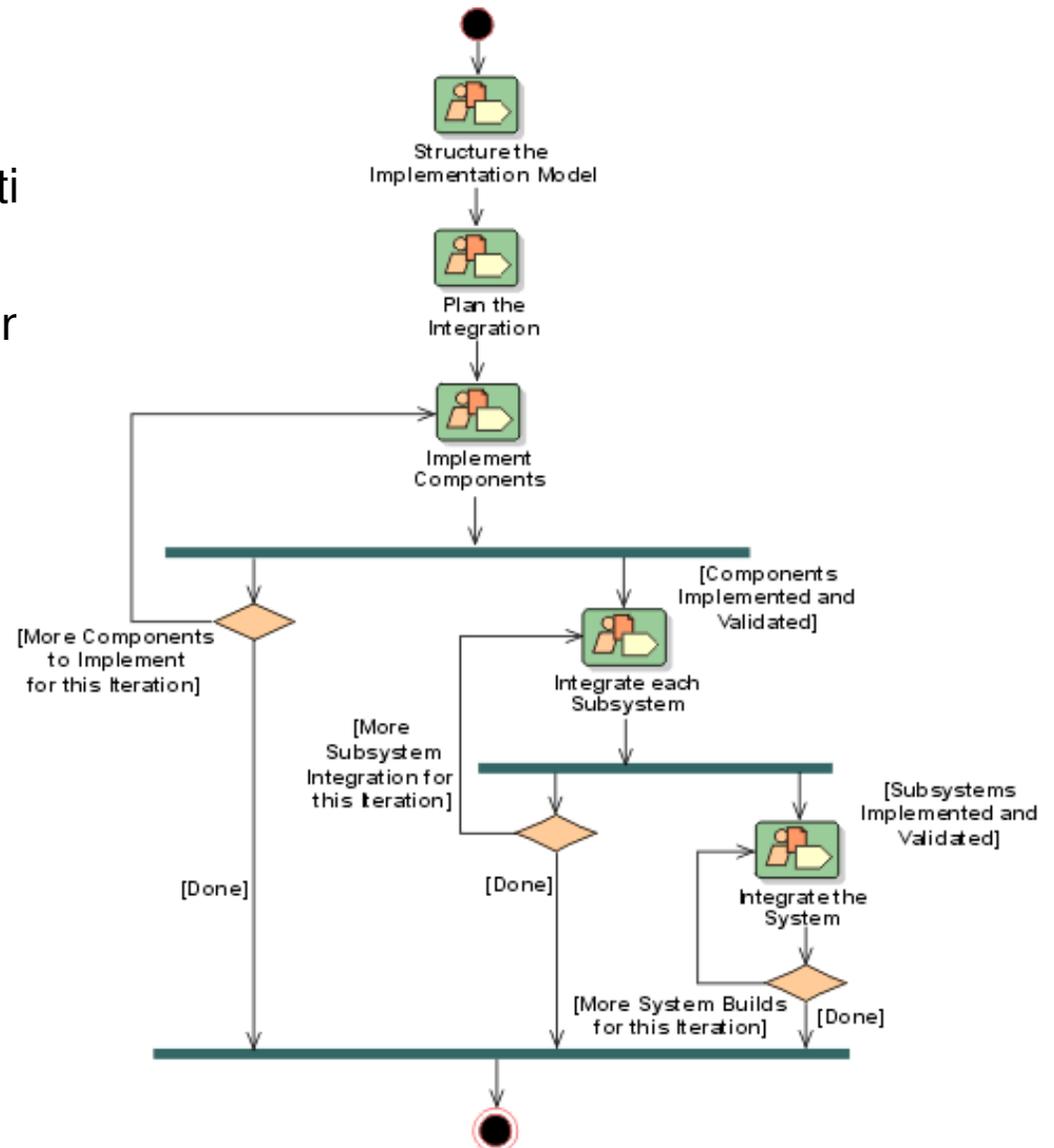
# Analisi e Progettazione

- Obiettivi
  - Ideare un modello dell'implementazione che realizzerà il sistema soddisfacendo requisiti e bisogno degli stakeholder
- Ruoli Principali
  - Architetto
  - Progettista Software
  - Progettista Database
- Elaborati Principali
  - Modello di analisi
  - Modello di design
  - Documento di Architettura software
  - Progetto del DB



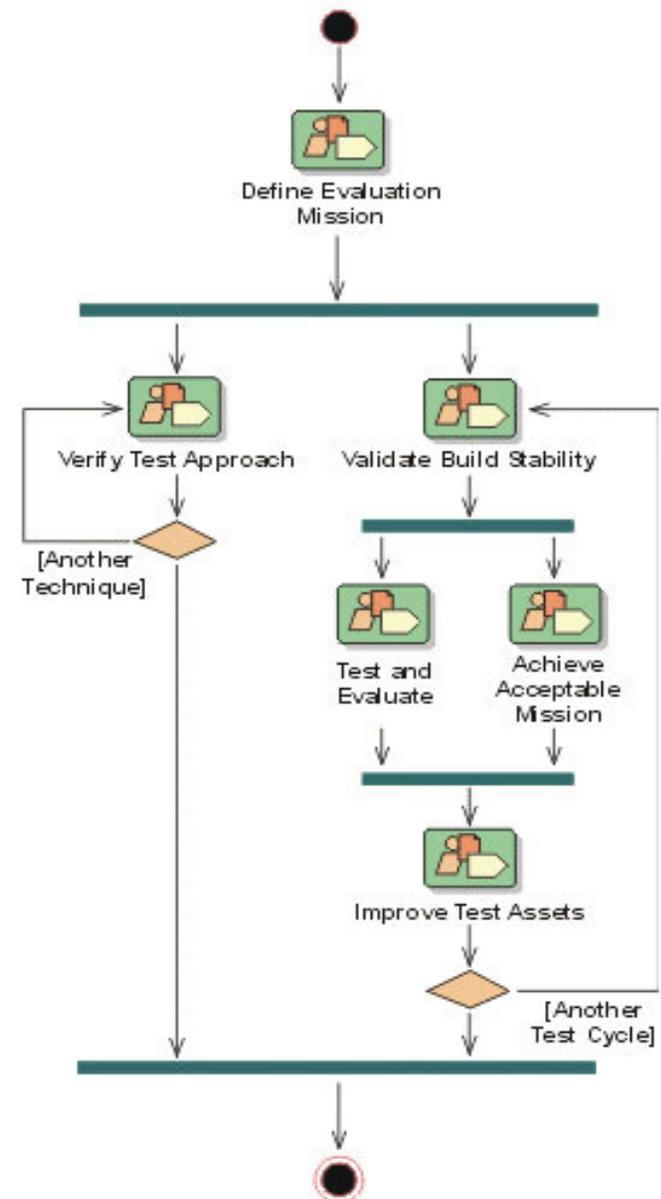
# Implementazione

- Obiettivi
  - Implementare classi e oggetti
  - Test di unità
  - Integrare quanto sviluppato in un sistema eseguibile
- Ruoli Principali
  - Sviluppatore
  - Sistemista d'integrazione
- Elaborati Principali
  - Componenti software (Sorgenti, eseguibili, DDL)
  - Piano d'integrazione



# Test

- Obiettivi
  - Verificare la corretta integrazione delle varie componenti
  - Verificare implementazione corretta dei requisiti
  - Identificare e risolvere eventuali difetti
- Ruoli Principali
  - Progettista Test e Tecnico dei test
  - Tecnico test prestazionali
  - Tecnico test sistema
- Elaborati Principali
  - Piano di esecuzione dei test
  - Modello dei test
  - Modello di carico dei test prestazionali
  - Documentazione difetti
  - Ambienti di esecuzione dei test



# Rilascio

## ■ Obiettivi

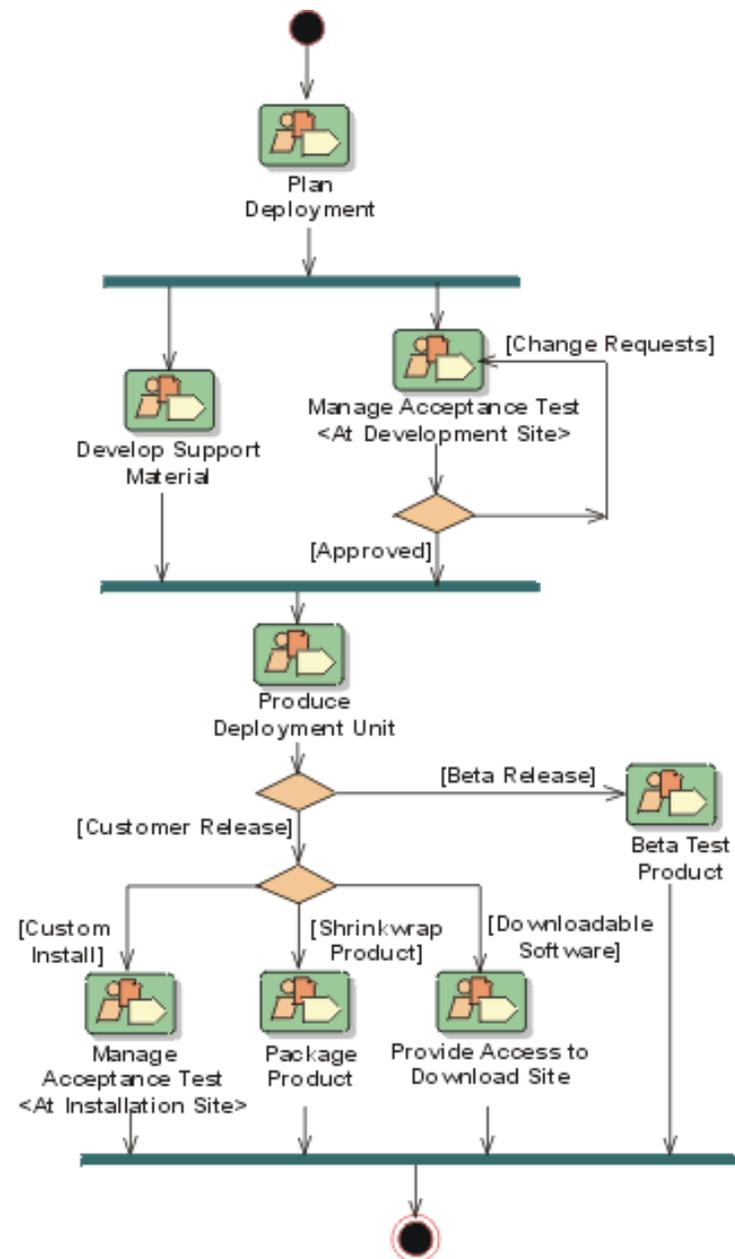
- Pianificare ed eseguire le attività di supporto al rilascio del sistema in esercizio e agli utenti finali
  - Produzione release installabile
  - Distribuzione e installazione
  - Formazione utenti
  - Migrazione dati e software esistenti

## ■ Ruoli Principali

- Responsabile Rilascio
- Redattore documentazione tecnica

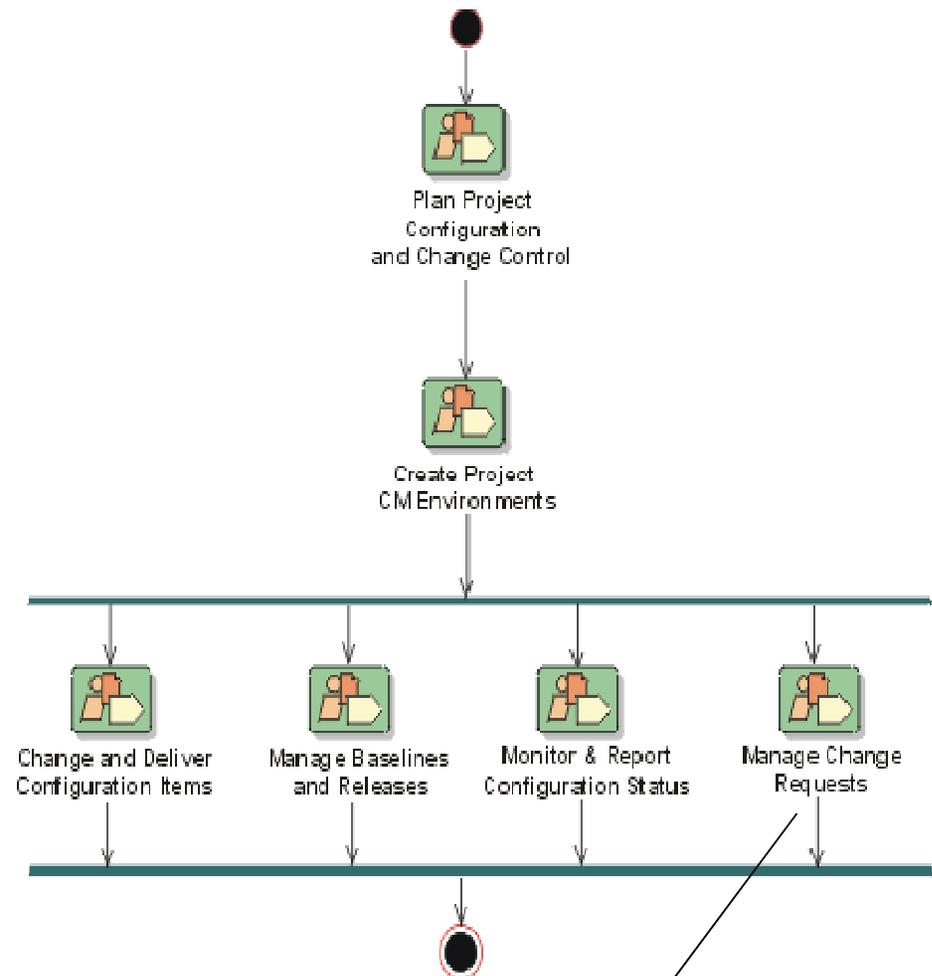
## ■ Elaborati Principali

- Piano di rilascio
- Manuali o loro assimilabili
- Materiale di formazione



# Controllo Configurazione e Cambiamenti

- Obiettivi
  - Tracciare e mantenere l'integrità del sistema man mano che evolve ed in presenza di modifiche concorrenti ed in parallelo
- Ruoli Principali
  - Responsabile gestione configurazione
- Elaborati Principali
  - Piano di gestione della configurazione (Piano di CM per change Management)
  - Richieste di modifica



Il Change Control Board (CCB) è un gruppo di stakeholder quali il PM, il resp config, l'architetto, e i rappresentanti dei clienti, che valutano le richieste di cambiamento e le approvano o rigettano ufficialmente, dopo averne valutato l'impatto sull'intero progetto

# Gestione del Progetto

## ■ Obiettivi

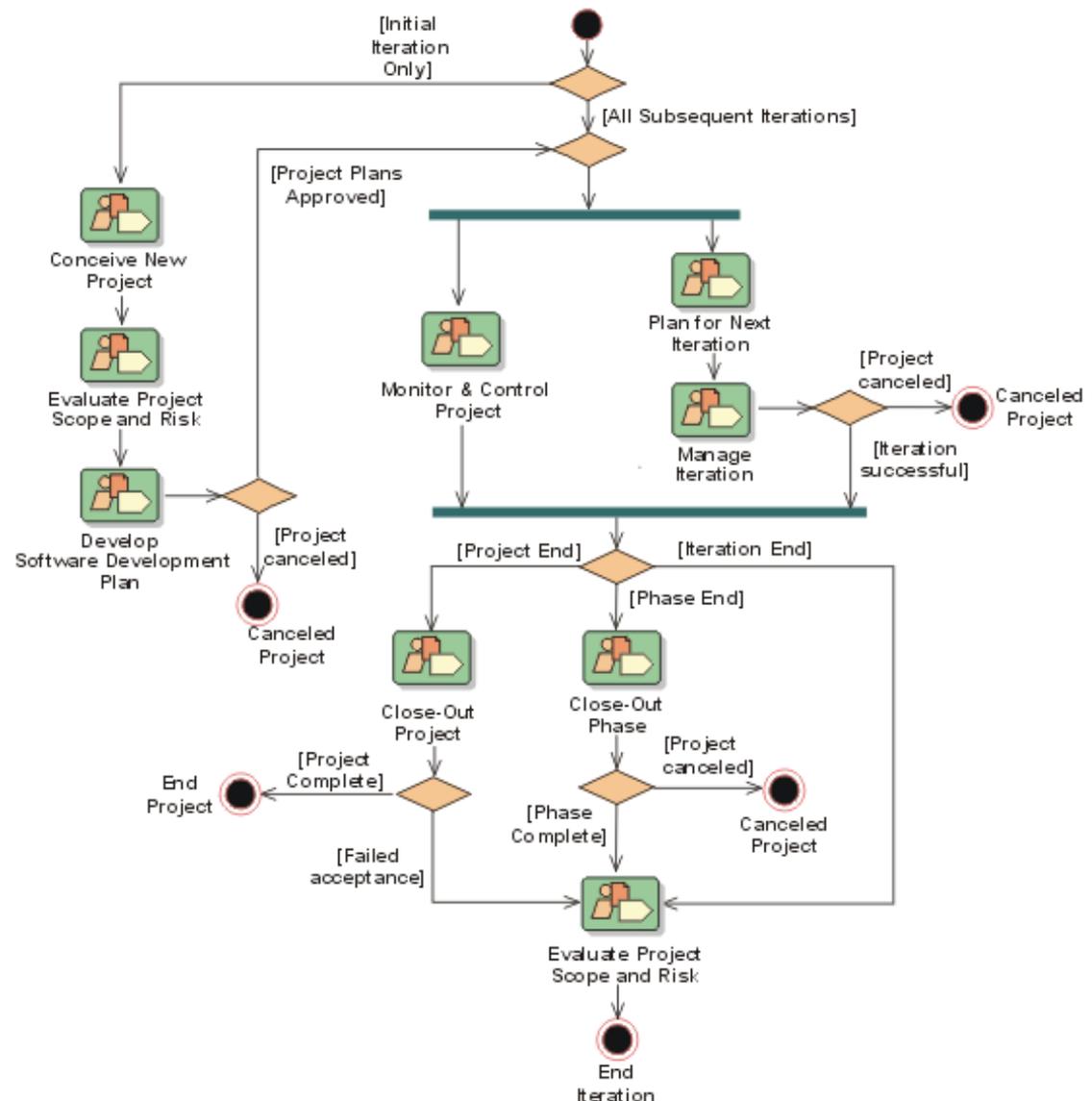
- Pianificazione del progetto e delle sue iterazioni
- Controllo dei tempi, costi, risorse umane assegnate, e dei rischi

## ■ Ruoli Principali

- Capo Progetto o PM per Project Manager

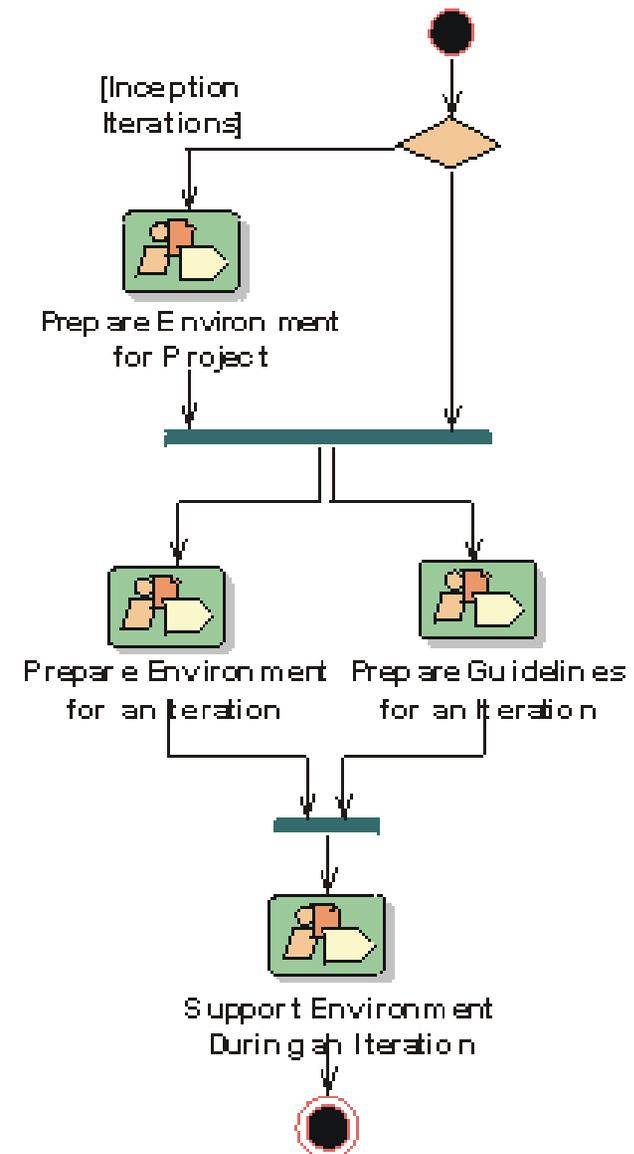
## ■ Elaborati Principali

- Piano di progetto
  - Piano delle fasi
  - Lista rischi
  - Piano misurazioni
  - Piano economico
- Piano delle iterazioni
  - Uno per ogni iterazione

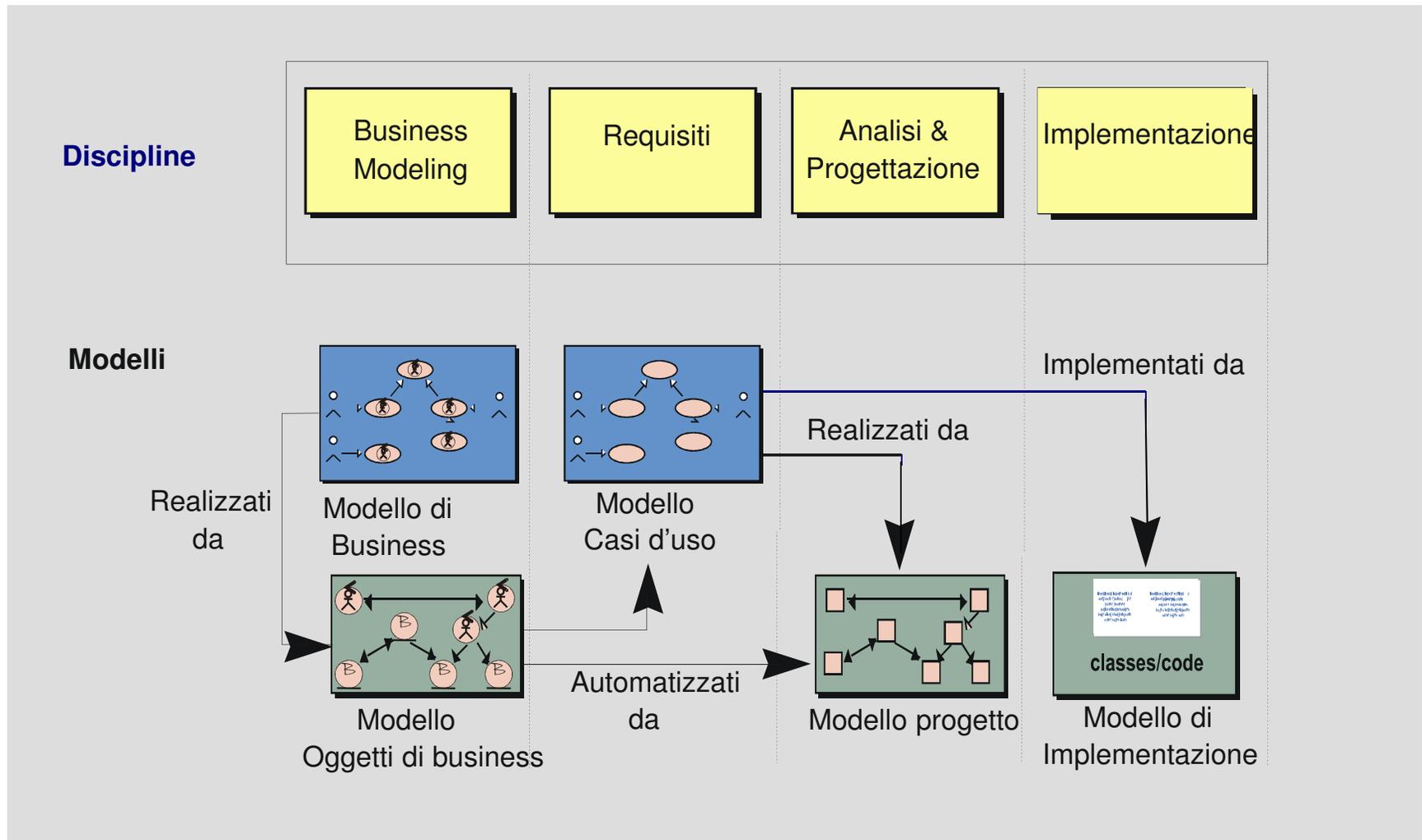


# Gestione degli Ambienti e Strumenti di infrastruttura

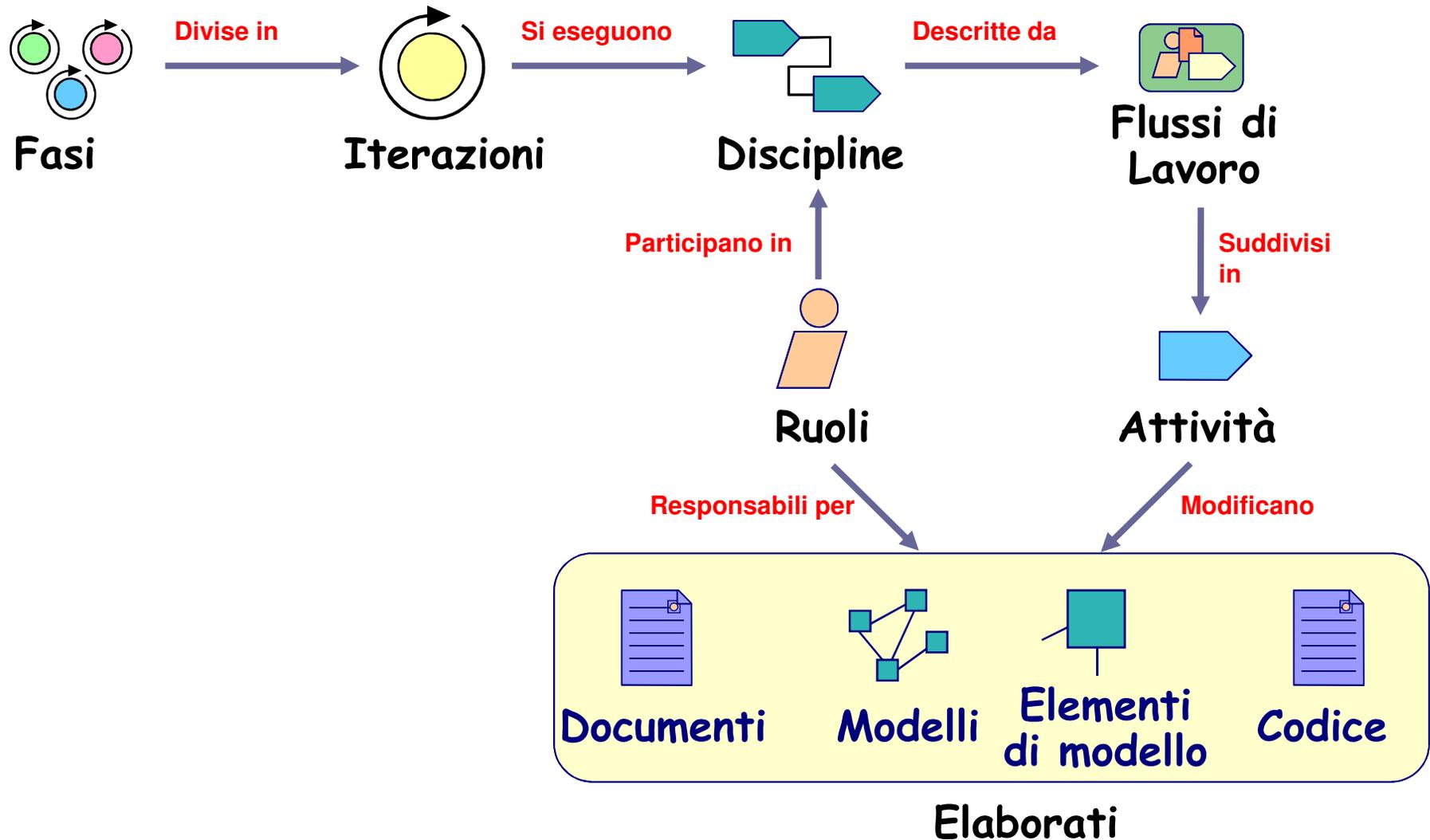
- Obiettivi
  - Fornire strumenti, piattaforme di sviluppo e target, tailoring dei processi, linee guida e standard al team di progetto
  - Configurazione degli strumenti e dei processi per il progetto
  - Servizi tecnici di accesso e sicurezza
- Ruoli Principali
  - Resp Infrastruttura, Resp. Qualità, Resp. Processi interni
  - Sistemisti e amministratori di sistema
- Elaborati Principali
  - Ambienti di supporto dell'intero ciclo di vita del software
    - IDE, Ambienti di Test, Strumenti di CM, Piattaforme target, etc
  - Hardware e Infrastrutture di rete



# I modelli delle discipline principali si supportano l'un l'altro



# Sommario dei concetti principali del RUP



# Come si usa il RUP

- RUP come detto, non è un processo ma un framework di processo dal quale istanziare il proprio processo di sviluppo software
  - Dovrebbe essere personalizzato con una attività di tailoring da ogni organizzazione sulla base delle proprie esigenze e delle caratteristiche di processo che vogliono mettere in piedi
- IBM Rational Method Composer è lo strumento che permette questa attività di tailoring
  - Permette di selezionare le componenti del processo di cui si ha bisogno, fare eventuali modifiche e quindi pubblicare l'intero processo su una intranet
  - **Una volta pubblicata la nostra istanza di RUP, ogni persona del nostro team, sa dove trovare le istruzioni, linee guida, e descrizioni delle attività che deve svolgere a seconda del suo ruolo**
  - **Il RUP inoltre è un repository di strumenti, tecniche, template, esempi di documenti e di modelli utili al progetto.**

# Evoluzione del RUP

Il RUP è evoluto in 3 direzioni:

1. Influenzando la standardizzazione di OMG del metaprocesso SPEM
  - Dal quale sono nati processi come OpenUP, Eclipse Open Framework (EOF) etc.
    - <http://www.omg.org/spec/SPEM/>
2. In un prodotto per la definizione di processi di sviluppo
  - IBM Rational Method Composer
    - Strumento venduto da IBM che permette di fare il tailoring e pubblicare il proprio processo adattato alle proprie esigenze ed obiettivi
    - <http://www-01.ibm.com/software/rational/rup/>
3. Facendo da padre putativo a vari altri metodi di sviluppo
  - Evoluzioni ed adattamenti di processi, pratiche e tecniche emergenti, come per esempio quelle tipiche dei movimenti di “Sviluppo Agile”, al RUP
  - Per esempio: Unified Process (UP), Agile Modeling (AM), Disciplined Agile Delivery (DAD) e Enterprise Unified Process (EUP)
    - [http://en.wikipedia.org/wiki/Unified\\_Process](http://en.wikipedia.org/wiki/Unified_Process)
    - <http://www.agilemodeling.com/>
    - <http://www.disciplinedagiledelivery.com/>
    - <http://www.enterpriseunifiedprocess.com/>

# SPEM – Software & System Process Engineering

- E' un metamodello per descrivere processi e framework di processi definito da OMG
  - Scaricabile gratuitamente da: <http://www.omg.org/spec/SPEM/>
- Utilizzato per definire IBM Method Composer
- Sulla specifica formale, oltre ai riferimenti al RUP, si trova:
  - Esempio di come il “Microsoft Solution Framework (MSF) Agile” può essere rappresentato usando SPEM 2.0
  - Descrizione di come il progetto EPF-Eclipse Open Framework ([www.eclipse.org/epf](http://www.eclipse.org/epf)), che è un progetto open source, fornisce un framework di processi basati su SPEM 2.0
    - OpenUP/Basic è un metodo di sviluppo software minimale, completo ed estensibile, sviluppato utilizzando EPF

# EPF – Eclipse Open Framework

- I due goal di EPF: (<http://projects.eclipse.org/projects/technology.epf>)
  1. Fornire un framework estensibile e strumenti esemplari per la definizione, configurazione e pubblicazione di processi di sviluppo software
  2. Fornire dei processi esemplari ed estensibili che supportano lo sviluppo iterativo, agile e incrementale
- Vi possiamo trovare i seguenti processi (scaricabili dal sito)  
[http://www.eclipse.org/epf/downloads/configurations/pubconfig\\_downloads.php](http://www.eclipse.org/epf/downloads/configurations/pubconfig_downloads.php)
  - **OpenUP**

E' una versione snella del processo UP, che applica un approccio iterativo ed incrementale ad un ciclo di vita strutturato. OpenUP adotta un approccio pragmatico dell'Agile che si focalizza sulla natura collaborativa dello sviluppo software. Questo processo è indipendente da strumenti a suo supporto ed è un processo a basso cerimoniale che può essere esteso per indirizzare una grande varietà di tipi di progetti.
  - **Scrum**

Framework di gestione dei processi Agile empirico usato per sviluppare in maniera iterativa e rilasciare al cliente incrementale di valore sempre maggiore. Scrum si basa su team che si organizzano autonomamente e su un cliente, detto Product Owner, che fornisce al team la lista delle feature desiderate usando un meccanismo di priorità basato sul valore di business.
  - **XP – eXtreme Programming**

Processo di sviluppo che può essere usato da team di sviluppo di piccole o medie dimensioni per lo sviluppo di software di alta qualità con tempistiche e budget predicibili e minimo overhead

# IBM Rational Method Composer

- Processi basati sulle best practice adottate da migliaia di progetti nel mondo
- Evitare di reinventare da zero un processo, riusando ciò che è stato implementato con successo e generalizzato dagli specialisti Rational
- Può comunque mettere a disposizione dei processi pronti-all'uso senza personalizzazioni, per iniziare velocemente e poi modificare il processo sulla base delle proprie esperienze e lessons-learned.

# Open Unified Process (OpenUP)

## ■ OpenUP

- Una istanza agile di UP che contiene l'insieme minimo di pratiche per adottare un processo di sviluppo agile, snello e leggero, applicando un approccio iterativo ed incrementale ad un ciclo di vita strutturato.
- OpenUP adotta un approccio pragmatico dell'Agile che si focalizza sulla natura collaborativa dello sviluppo software.
- Questo processo è indipendente da strumenti a suo supporto ed è un processo a basso cerimoniale che può essere esteso per indirizzare una grande varietà di tipi di progetti
- Essendo un processo minimale, non copre molti argomenti che un team di progetto di medie-grandi dimensioni potrebbe avere l'esigenza di coprire

## ■ Potete scaricare OpenUP dal seguente sito:

[http://www.eclipse.org/epf/downloads/configurations/pubconfig\\_downloads.php](http://www.eclipse.org/epf/downloads/configurations/pubconfig_downloads.php)

# Sviluppo Agile e RUP

- Il RUP ha divulgato e sdoganato molti dei concetti alla base del movimento Agile
  - E' tuttora difficile convincere alcune organizzazioni ad evolvere da un processo a cascata ed adottare un processo iterativo
  - Il concetto del testing continuativo, sviluppare una versione eseguibile per ogni iterazione, e collaborazione con gli stakeholder sono solo alcune dei concetti che RUP ha contribuito a far divenire ben accetti al giorno d'oggi.
- RUP has adottato molte delle tecniche "Agile"
  - Essendo un framework che contiene delle best practice che debbono essere selezionate e pubblicate, questo insieme di best practice evolve continuamente
  - Le tecniche tipiche e di maggior successo dello sviluppo Agile sono ora presenti nel RUP, tipo Test-Driven Development (TDD), integrazione continua, etc.
- RUP è Agile o non Agile, a seconda di come vogliamo farlo essere
  - Come detto, RUP è un framework che deve essere adattato ed istanziato
  - L'adattamento, o personalizzazione, o meglio "tailoring" del processo che facciamo noi, lo possiamo indirizzare all'Agile o in direzione diversa a nostro piacimento

"Un processo RUP ben definito è Agile. Un processo RUP mal definito è semplicemente sbagliato!"

by Scott Ambler, Senior Consulting Partner of Scott Ambler and Associated

# Esercitarsi sui concetti visti

## ■ Scaricarsi:

- IBM Rational Method Composer
  - da: <http://www-01.ibm.com/software/rational/rup>
- OpenUP
- Scrum
- XP – eXtreme Programming
  - da: [http://www.eclipse.org/epf/downloads/configurations/pubconfig\\_downloads.php](http://www.eclipse.org/epf/downloads/configurations/pubconfig_downloads.php)

## ■ Task:

1. Scoprire come sono descritte le fasi, le discipline e i ruoli dei partecipanti al team di sviluppo nei vari processi
2. Scoprire quali template mettono a disposizione
3. Scoprire come personalizzarli alle proprie esigenze, definendo un nuovo template di Vision ed una nuova attività di validazione dei requisiti rispetto alla normativa italiana sull'accessibilità dei siti web (<http://www.governo.it/Presidenza/web/accessibilita.html>)