

Lezione 2

Cicli di Vita e Processi Software

Riferimenti bibliografici

- I. Sommerville – Ingegneria del Software – 8a edizione – Cap.4
- R. Pressman- Principi di Ingegneria del Software- 4 edizione- Cap. 3

Obiettivi della lezione

1. Richiamare i concetti di Ciclo di Vita del Software e di Ciclo di Sviluppo Software
2. Classificare i Modelli di Processo Software fondamentali

Il Ciclo di Vita del Software (CVS)

- Standard IEEE 610.12-1990
- ***Software Life Cycle***: The period of time that starts when a software product is conceived and ends when the product is no longer available for use. The software life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and sometimes, retirement phase. Note: These phases may overlap or be performed iteratively. Contrast with *Software Development Cycle*.

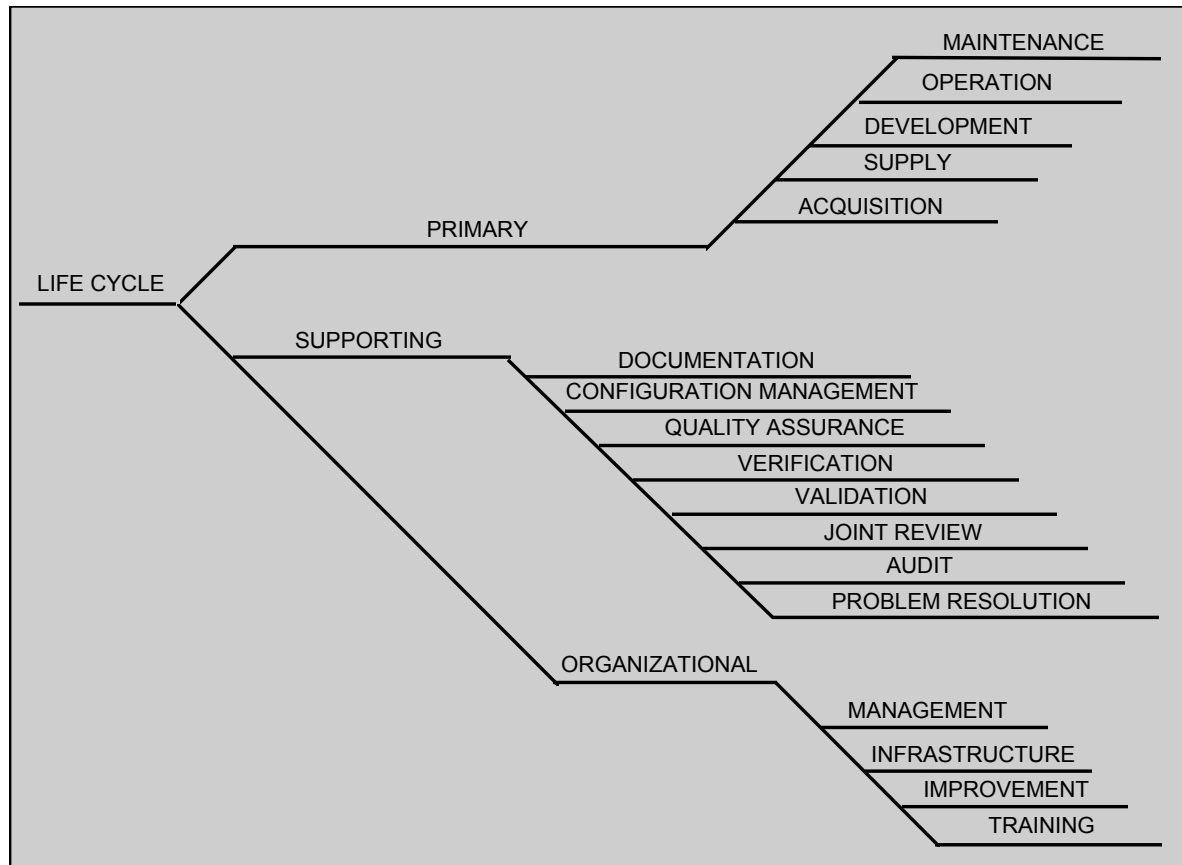
Ciclo di Sviluppo Software

- Standard IEEE 610.12-1990
- ***Software Development Cycle*** : (1) The period of time that begins with the decision to develop a software product and ends when the product is delivered. This cycle typically includes a requirements phase, design phase, implementation phase, test phase, and sometimes, installation and checkout phase. Contrast with *Software Life Cycle*.
- **Notes:** The phases listed above may overlap or be performed iteratively, depending upon the software development approach used. (2) This term is sometimes used to mean a longer period of time, either the period that ends when the software is no longer being enhanced by the developer, or the entire software life cycle

Una architettura per la definizione di Cicli di Vita del Software

- Standard IEEE 12207.0-1996
- ***Software Life Cycle***: a framework containing the processes, activities, and tasks in the development, operation and maintenance of a software product, spanning the life of the system from the definition of its requirements to the termination of its use

Ciclo di Vita ed i suoi Processi (St.ISO 12207) – vedi appendice



Processi Software: definizione

- Processo :
 - un insieme di attività concentrate nel tempo finalizzate alla realizzazione di un particolare output.
- Processo Software:
 - Un insieme strutturato di attività necessarie per lo sviluppo di un sistema software.
 - Specifica;
 - Progettazione (o Design);
 - Validazione;
 - Evoluzione

Modelli di Processo Software

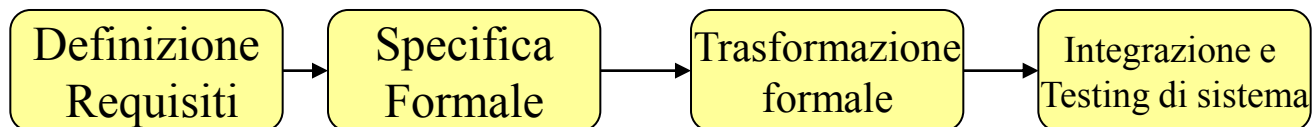
- Un processo software può essere organizzato usando i *Modelli di Processo Software*:
 - I **Modelli di Processo Software** (talora detti ancora CVS) definiscono la struttura di massima di un processo software, indicando le fasi in cui si articola e i criteri di successione.
 - Un Modello di processo software è una rappresentazione astratta di un processo.
 - Fornisce una descrizione del processo da una particolare prospettiva.

Quali Modelli di Processo Software?

- Una possibile classificazione:
 1. Processi basati su documentazione
 - Basati su linguaggi semi-formali per la specifica dei documenti (es. UML), e richiedono trasformazioni e controlli manuali dei prodotti intermedi
 2. Processi basati su Metodi Formali
 - Basati su linguaggi formali di specifica, con trasformazioni e controlli automatici dei prodotti intermedi
 3. Approcci Agili
 - Uso ridotto di documentazione

Processi basati su Metodi Formali (cenni)

- Uso di formalismo matematico per esprimere i requisiti
 - Es. Specifiche algebriche (es. per tipi di dato astratto)
 - Es. approccio basato su modelli di stato
- Uso di tecniche di model checking per provare la correttezza
- Trasformano automaticamente le specifiche in codice
 - La correttezza è preservata
 - La verifica è ottenuta implicitamente



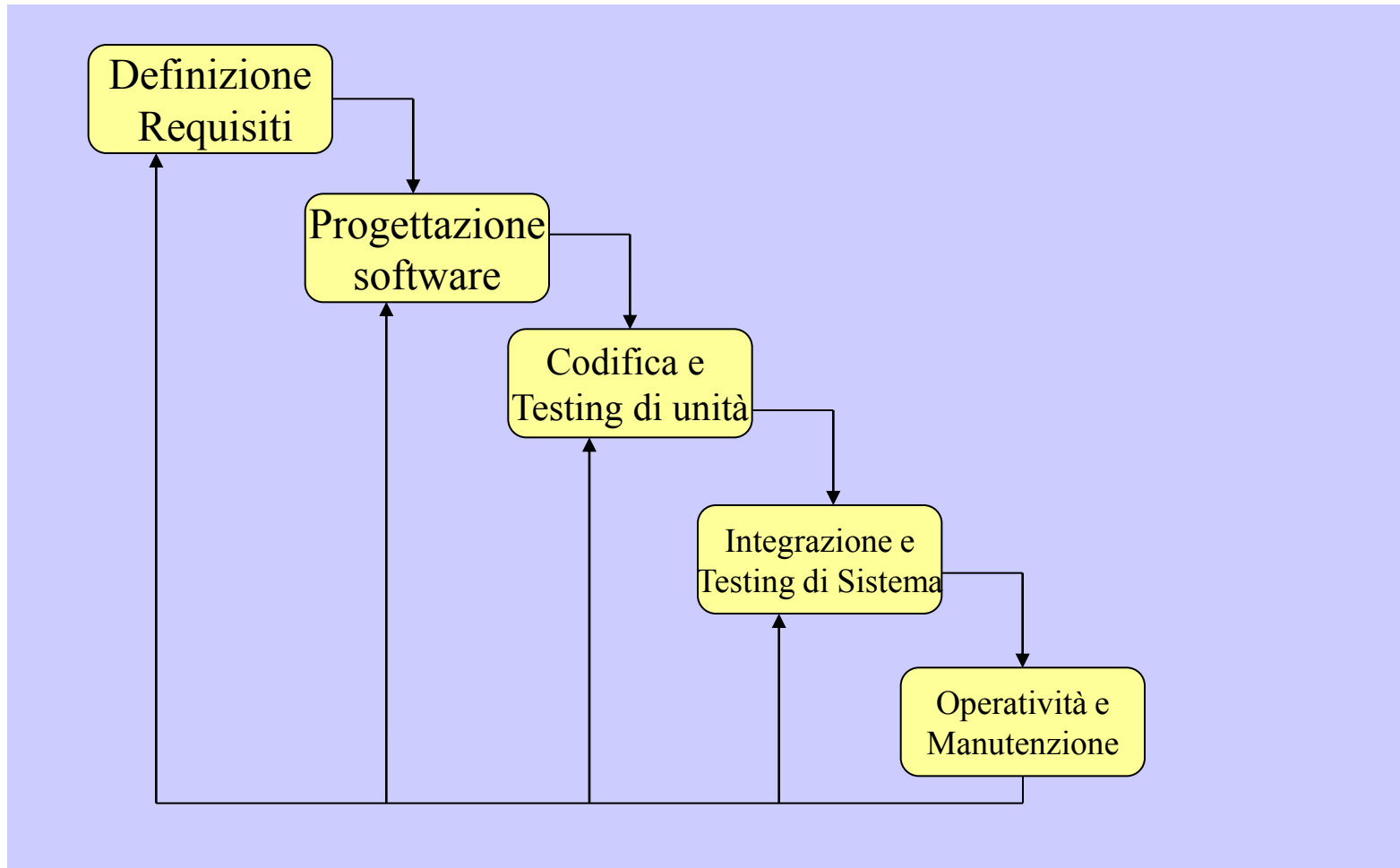
Processi basati su Metodi Formali

- Problemi
 - necessità di specifici skill (in linguaggi matematici)
 - Impossibilità a specificare formalmente alcune parti
 - Difficoltà del cliente nella convalida dei requisiti
- Applicabilità
 - Non adatti per sistemi di grandi dimensioni
 - Usati solo per parti critiche

Classificazione dei Modelli di Processo

- Basati su documentazione:
 - Waterfall
 - Evolutivo
 - Basato sul Riutilizzo (di Componenti, Linee di Prodotto...)
 - Iterativo / Incrementale
 - Rational Unified Process (UP e RUP) ...
- Agili:
 - XP (eXtreme Programming)
 - TDD (Test Driven Development)

Il Waterfall model



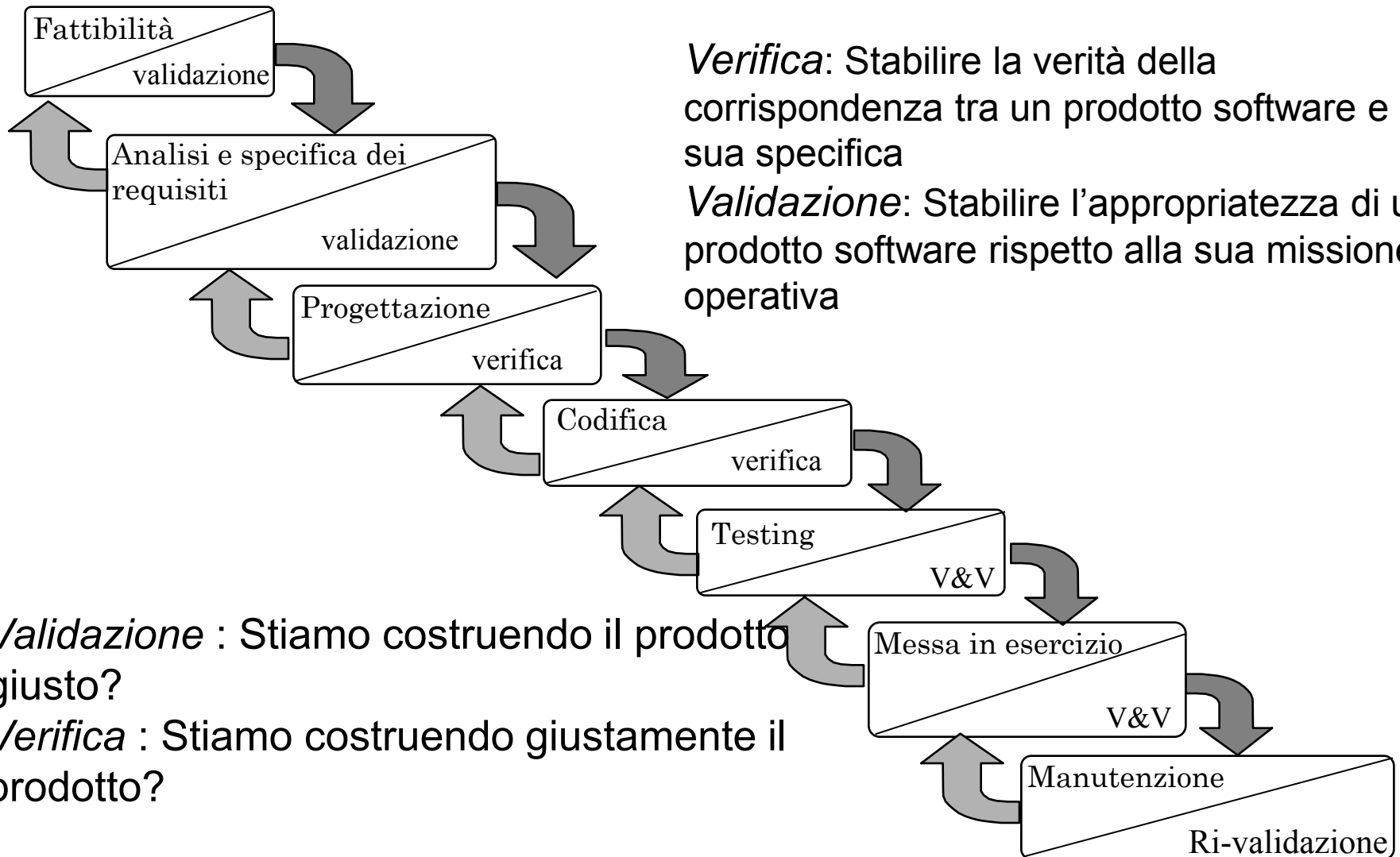
Il Waterfall model

- Fasi sequenziali ben definite
- Sviluppo Guidato dalla documentazione

ma...

- Troppo rigido/ burocratizzato
- Rilascia software solo a completamento di tutte le fasi
- Richiede conoscenza immediata e stabilità dei requisiti

Modello a cascata: una variante



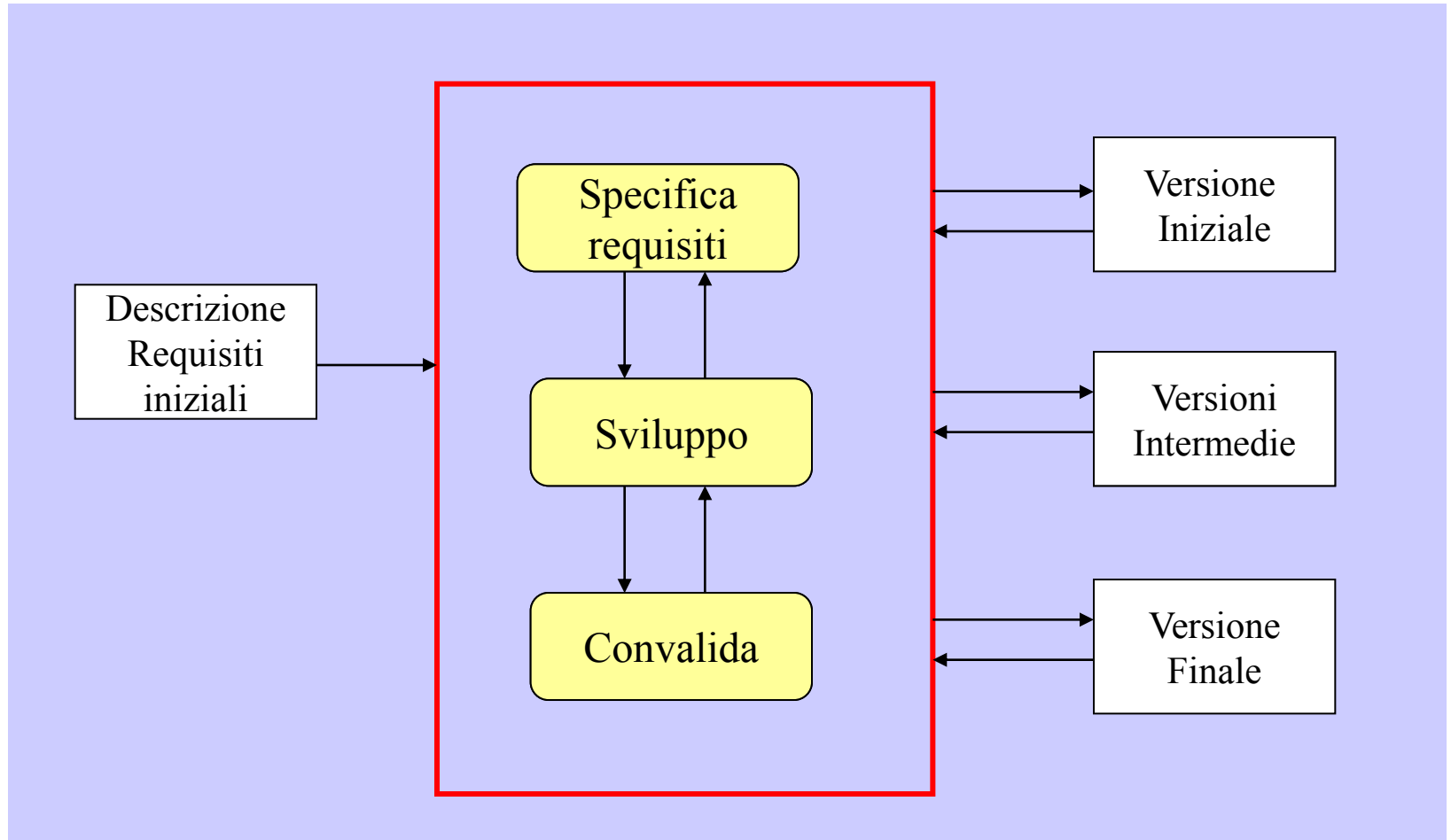
Validazione : Stiamo costruendo il prodotto giusto?

Verifica : Stiamo costruendo giustamente il prodotto?

Sviluppo Evolutivo

- Basato sull'idea di produrre una versione iniziale del software, esporla agli utenti e perfezionarla attraverso varie versioni. Due modelli fondamentali:
- **(A) Sviluppo Esplorativo**
 - L'obiettivo è di lavorare col cliente per esaminare i requisiti iniziali e farli evolvere fino al sistema finale. Dovrebbe partire da pochi requisiti ben compresi e aggiungere nuove caratteristiche proposte dal cliente.
- **(B) Prototipale (Usa e Getta)**
 - L'obiettivo è di comprendere i requisiti del sistema. Si parte da requisiti poco chiari e si realizzano prototipi per esplorare i requisiti e chiarirli.

(A) Sviluppo evolutivo - Esplorativo



(A) Sviluppo evolutivo- Esplorativo

- Vantaggi:
 - Rapido feedback e possibilità di far cambiare i requisiti
- Problemi
 - Mancanza di visibilità del processo (è anti-economico documentare ogni versione del sistema);
 - I sistemi diventano spesso mal strutturati (per i continui cambiamenti);
 - Richiedono particolari skills (es. Uso di linguaggi di prototipazione rapida)
- Applicabilità
 - A sistemi interattivi di piccole e medie dimensioni (<500.000 linee di codice (LOC));
 - Per sviluppare alcune parti di sistemi di grandi dimensioni (per es. l'interfaccia utente);
 - Per sistemi destinati a vita breve.
- Per grandi sistemi è consigliabile un approccio misto.

(B) Sviluppo Evolutivo- Prototipale

- Realizzazione di una prima implementazione (prototipo), più o meno incompleta da considerare come una 'prova', con lo scopo di:
 - accertare la Fattibilità del prodotto
 - validare i Requisiti
- Dopo la fase di utilizzo del prototipo si passa alla produzione della versione definitiva del Sistema Sw mediante un modello che, in generale, è di tipo waterfall.

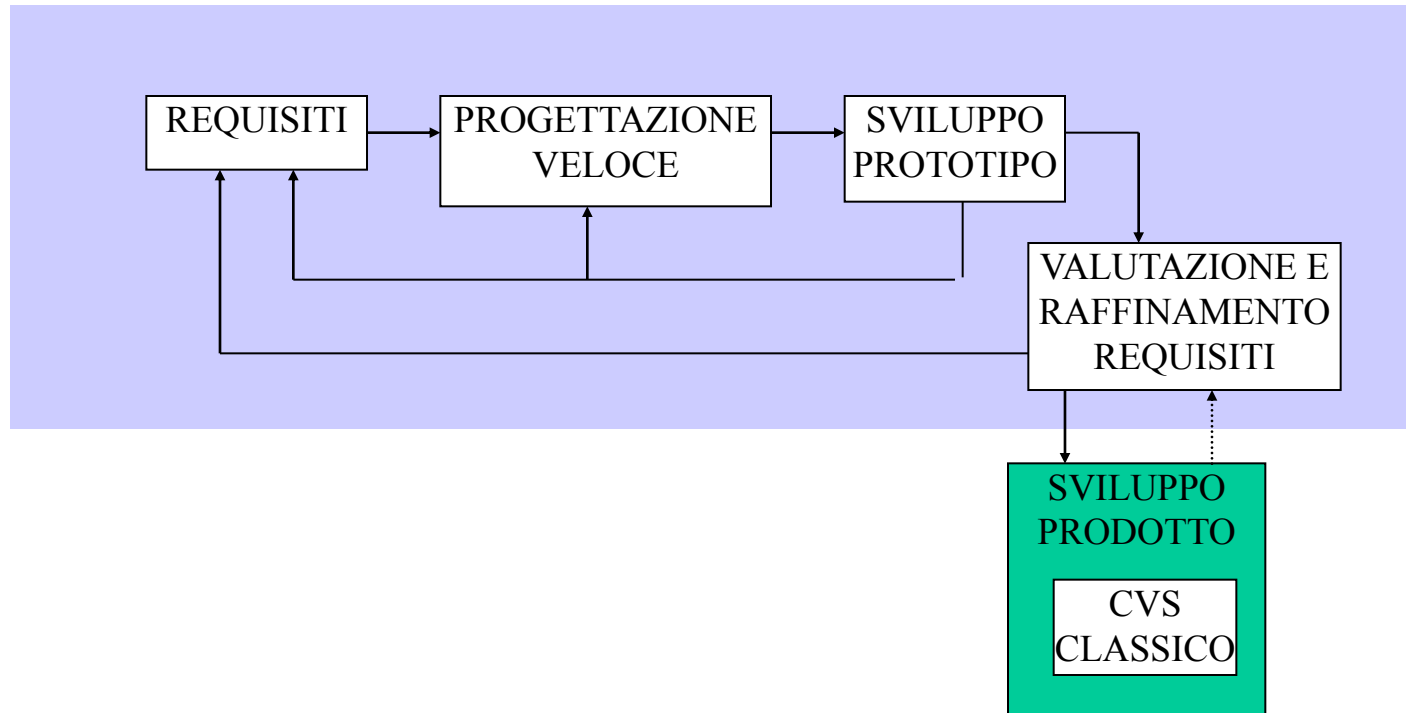
Prototipizzazione del Software

- A volte per problemi contrattuali, non può essere usato un approccio incrementale, tuttavia i requisiti non sono ben chiari o altri aspetti progettuali non sono ben definiti.
- Si può allora sviluppare un prototipo (usa e getta) a scopi esplorativi.
- Un prototipo può essere usato:
 - Nel processo di ingegnerizzazione dei requisiti per raccogliere e validare requisiti;
 - Nei processi di progettazione per esplorare nuove soluzioni e sviluppare una UI;
 - Nel processo di testing per eseguire test back-to-back (cioè confrontare il risultato mostrato dal prototipo con quello del sistema complessivo).

Modelli con prototipo

- 2 tipi di prototipazione:
 - **mock-ups**: produzione completa dell'interfaccia utente. Consente di definire con completezza e senza ambiguità i requisiti (si può, già in questa fase, definire il manuale di utente)
 - **breadboards**: implementazione di sottoinsiemi di funzionalità critiche del SS, non nel senso della fattibilità ma in quello dei vincoli pesanti che sono posti nel funzionamento del SS (carichi elevati, tempo di risposta, ...), senza le interfacce utente. Produce feedbacks su come implementare la funzionalità (in pratica si cerca di conoscere prima di garantire).

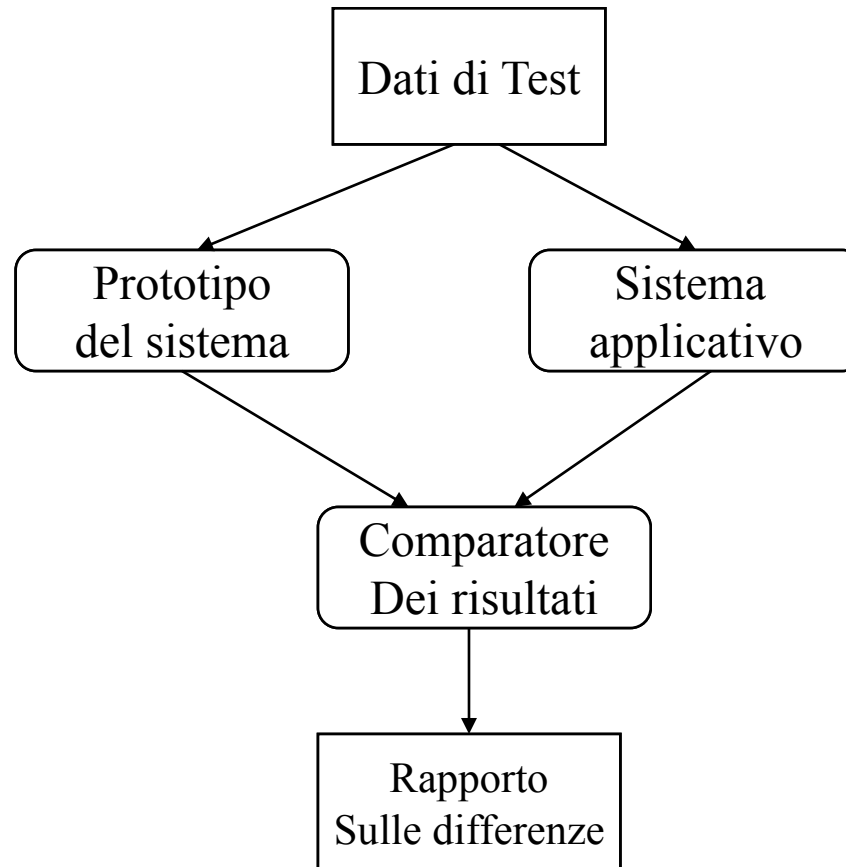
Modello con Prototipo (Usa e Getta)



Benefici della prototipazione

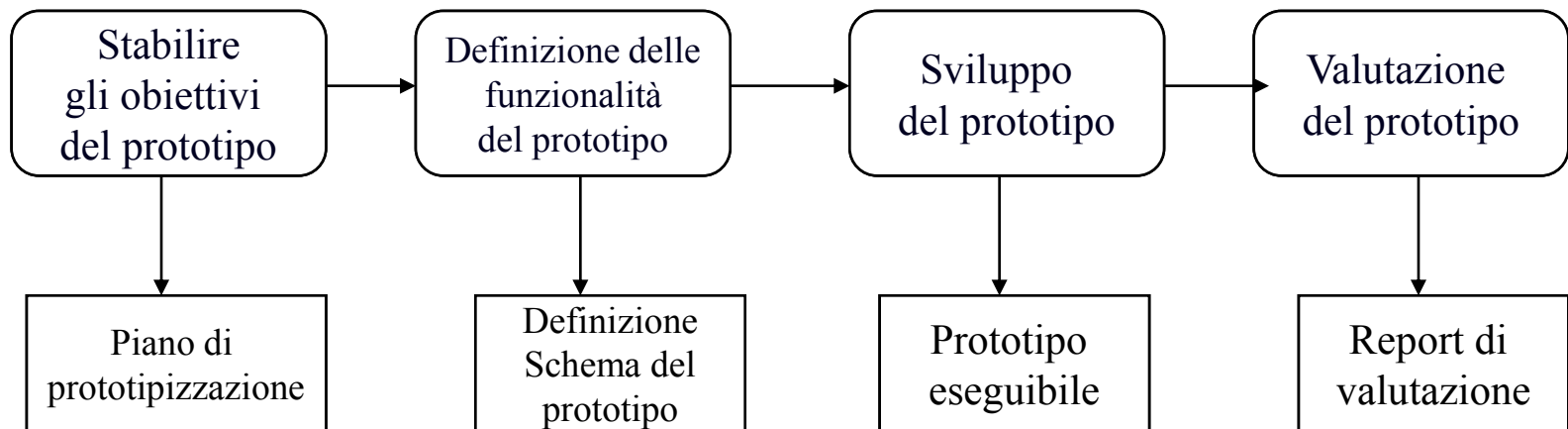
- Migliora l'usabilità del sistema.
- Garantisce una maggiore corrispondenza con i reali bisogni utente.
- Permette di esplorare soluzioni di progetto di migliore qualità.
- Migliora la capacità di manutenzione.
- Riduce lo sforzo di sviluppo.

Uso del prototipo nel Back to back testing



Il prototipo
può fare,
talvolta, le veci
dell'oracolo

Il processo di prototipazione



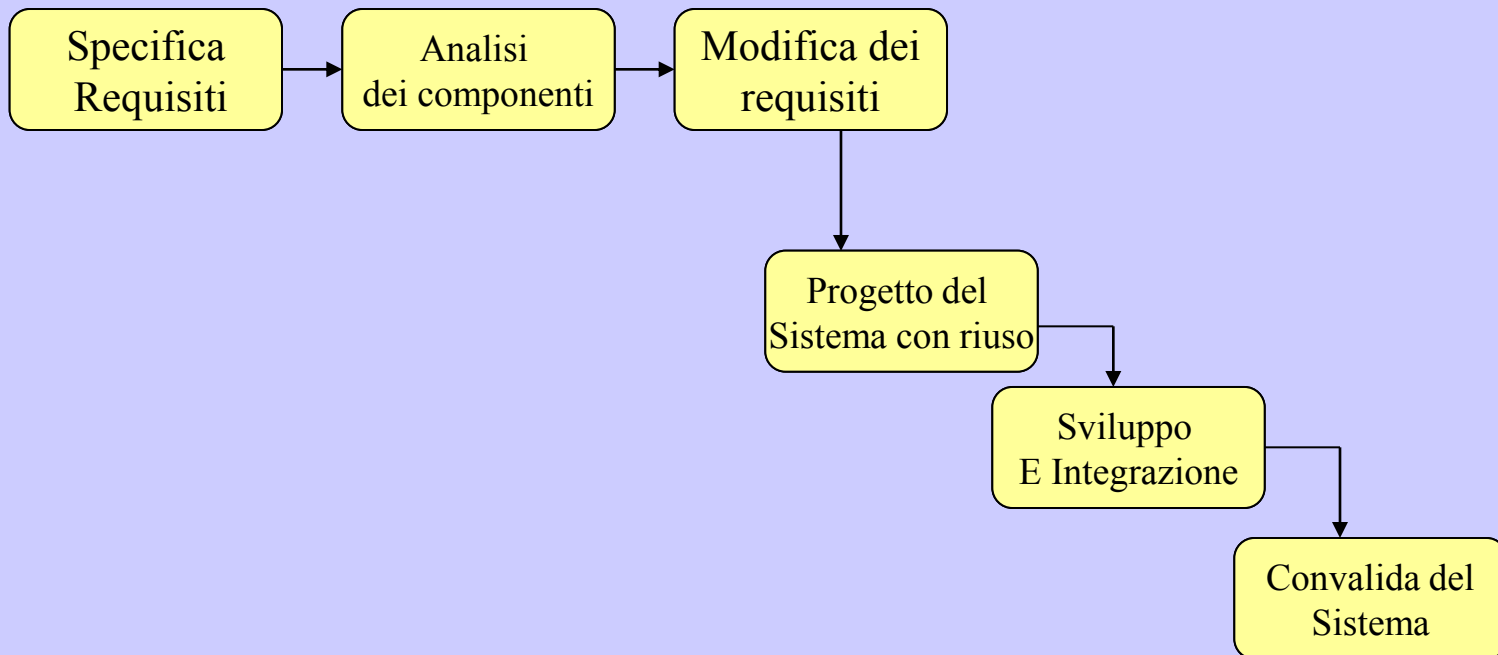
Prototipi usa e getta

- I prototipi dovrebbero essere eliminati dopo l'uso, perchè non sono una buona base di partenza per la produzione del sistema :
 - Potrebbe essere impossibile adattare il prototipo per soddisfare i requisiti non-funzionali;
 - I prototipi non sono ben documentati;
 - La struttura del prototipo si degrada a causa dei rapidi cambiamenti;
 - Il prototipo normalmente non soddisfa gli standard di qualità dell'organizzazione.

IS basata sul Riuso

- Sviluppo basato sul riuso sistematico, integrando Componenti esistenti o interi sistemi COTS (Commercial-off-the-shelf)
- Fasi del Processo (CBSE)
 - Specifica dei requisiti;
 - Analisi dei componenti;
 - Modifica dei Requisiti (specificando i componenti disponibili);
 - Progettazione con riuso;
 - Sviluppo e Integrazione.
- Necessita di appositi standard per la specifica di componenti.

Sviluppo basato sul Riuso



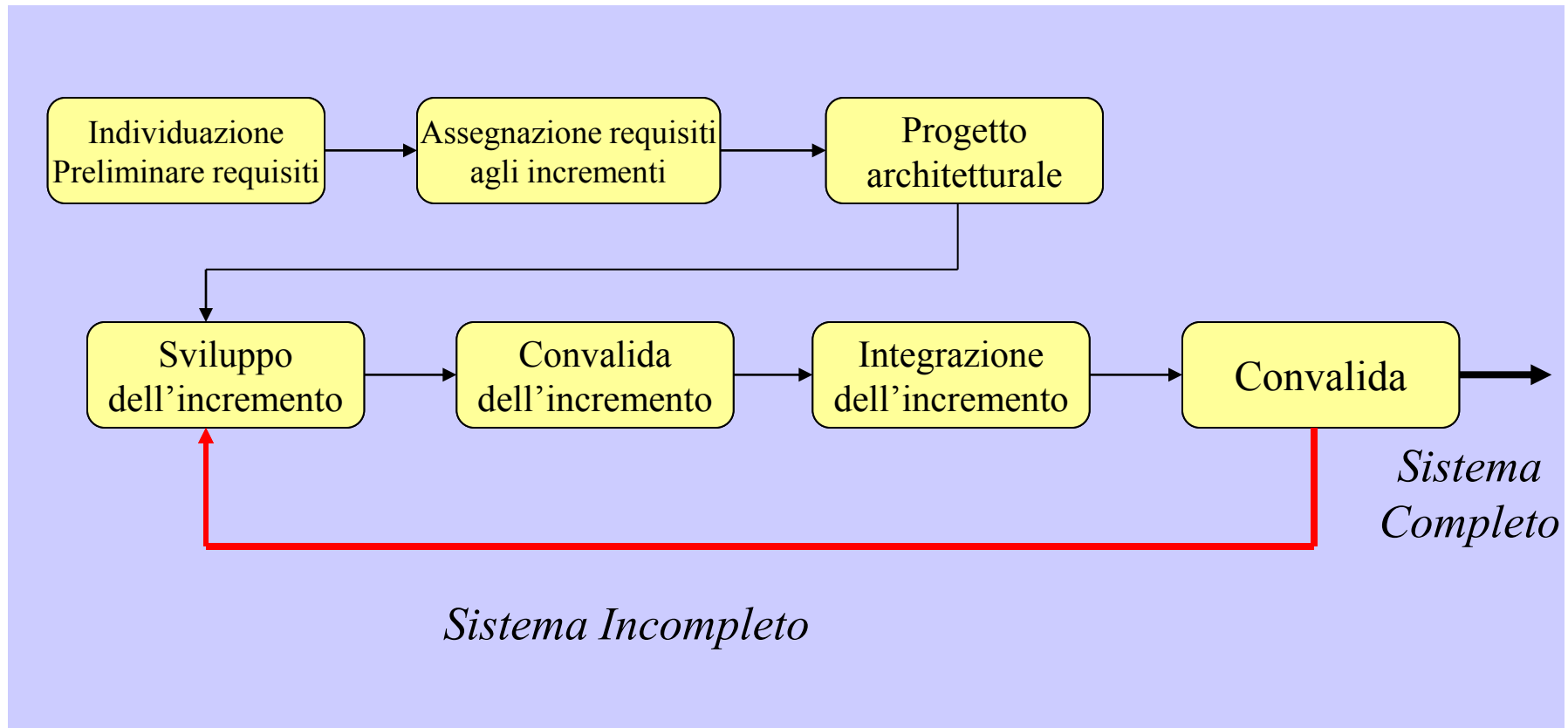
Processi di sviluppo Iterativi

- In un ambiente globale di rapidi cambiamenti, è essenziale per la competitività delle aziende che il software di cui necessitano sia sviluppato e consegnato in tempi rapidi!
- La mutevolezza ed instabilità dei requisiti impongono di adottare processi di sviluppo ciclici, in cui il software viene consegnato in una serie di incrementi, prodotti in vari cicli di processo (iterazioni) in cui si ritorna sulle fasi già condotte
- I cicli di processo possono essere applicati ad ogni generico modello di processo.

(A) Sviluppo e Consegna Incrementale

- Piuttosto che consegnare il sistema tutto in una volta, lo sviluppo e la consegna sono eseguiti per incrementi, dove ogni incremento rilascia parte delle funzionalità richieste.
 - Ai requisiti Utente vengono associati **livelli di priorità** e quelli a priorità maggiore vengono rilasciati con i primi incrementi.
 - Una volta partito lo sviluppo di un incremento, i relativi requisiti devono essere congelati, mentre i requisiti coinvolti in incrementi successivi possono continuare ad evolvere.
 - I servizi comuni possono essere implementati all'inizio del processo, o quando una funzione è richiesta da un dato incremento.

(A) Sviluppo e Consegna Incrementale



Vantaggi dello sviluppo incrementale

- I clienti non devono aspettare il sistema completo per la consegna, ma possono disporre al più presto dei requisiti più critici, attraverso i primi incrementi.
- I primi incrementi possono essere usati come *prototipo* per aiutare a definire i requisiti degli incrementi successivi.
- Si riduce il rischio di un fallimento totale del progetto.
- I servizi a più alta priorità saranno anche testati più intensamente degli altri.

Problemi dello sviluppo Incrementale

- Problemi
 - Gli incrementi devono essere relativamente piccoli (≤ 20.000 LOC) ma può essere difficile predisporre i requisiti in incrementi della dimensione giusta.
 - Le funzionalità comuni (richieste da più requisiti) potrebbero non essere identificate abbastanza presto, giacchè bisogna prima attendere che gli incrementi siano completati per avere ben chiari tutti i requisiti.

Un esempio: Extreme programming

- Extreme Programming
 - Un esempio di approccio allo sviluppo basato su sviluppo e consegna di piccolissimi incrementi di funzionalità.
 - Si basa su un continuo miglioramento del codice, sul coinvolgimento dell'utente nel processo di sviluppo, e sulla programmazione a coppie.
 - ... verrà approfondito in futuro

Confronto tra sviluppo evolutivo e incrementale

- Nello sviluppo evolutivo il software viene realizzato in diverse evoluzioni
 - La versione prototipale è una possibile prima versione
 - Ma le singole evoluzioni non sono rilasciate al cliente
- Nello sviluppo incrementale il software viene realizzato per incrementi successivi
 - Al termine di ogni incremento può essere rilasciata una release al cliente
 - Riduzione dei rischi
 - Possibile peggioramento della qualità della progettazione
 - Necessità di gestire bene la transizione tra le versioni
- Entrambi gli approcci sono adatti a software i cui requisiti non sono completamente noti e compresi all'atto dell'inizio della progettazione e dello sviluppo

- **Il RUP (Rational Unified Process)**
 - Un esempio di modello di processo “ibrido”

Unified Process (UP)

Booch, Rumbaugh, Jacobson (autori sia di UML che UP)

UML (Unified Modeling Language)

- Linguaggio per la modellazione e costruzione di sistemi software.
- È Standard OMG (Object Management Group) dal 1997

UP (Unified Process)

- È il processo definito dai tre autori per lo sviluppo con utilizzo di UML
- Non è uno standard
- La versione commerciale è nota come RUP (Rational Unified Process) creata da IBM

Rational Unified Process (RUP)

- E' un framework (schema generale) di processo da adattare alle diverse tipologie di progetto
 - Approccio iterativo e incrementale
 - In ogni iterazione si svolgono tutti i passi del processo per una parte del progetto
 - Il rilascio avviene in maniera progressiva
 - Guidato dai casi d'uso
 - Incentrato sull'architettura
- **Individua 3 prospettive sul processo:**
 - Una prospettiva **dinamica** che mostra le *fasi* del modello al fluire del tempo;
 - Una prospettiva **statica** che mostra le *attività* (workflow) coinvolte;
 - Una prospettiva **pratica** che suggerisce le buone *regole* da seguire.

Cosa c'è in RUP

- Un'organizzazione del piano di progetto per fasi sequenziali
- Indicazioni sulle attività da svolgere e sulla loro sequenza
- Un insieme di ruoli predefiniti
- Un insieme di documenti da produrre, con template ed esempi già realizzati
- Un insieme di *regole del gioco* da seguire nel lavoro quotidiano
 - Alcune (poche) regole sono obbligatorie
 - La maggioranza sono da adattare allo specifico progetto

Fasi e Workflow

- In RUP ci sono 6 workflow, ognuno composto di 4 fasi, ognuna delle quali viene eseguita in più iterazioni, più 3 workflow di supporto

Workflow

1. Modellazione processi di business
2. Requisiti
3. Analisi e Progettazione
4. Implementazione
5. Test
6. Rilascio

Fasi

- Inception
- Elaboration
- Construction
- Transition

Le fasi del RUP

- Avvio (Inception)
 - Stabilire gli obiettivi di business (e relativi limiti, fattibilità e giustificazioni economiche) per il sistema.
- Elaborazione (Elaboration)
 - Ottenere una comprensione del dominio del problema (e specificare i requisiti), stabilire una struttura architeturale ed il piano di progetto.
- Costruzione (Construction)
 - Progettare, programmare e testare il sistema incrementalmente.
- Transizione (Transition)
 - Trasferire il sistema nel suo ambiente operativo.
 - Caratteristica tipica di RUP

I Workflow

- **Il RUP prevede vari workflow (6 attività principali e 3 di supporto):**

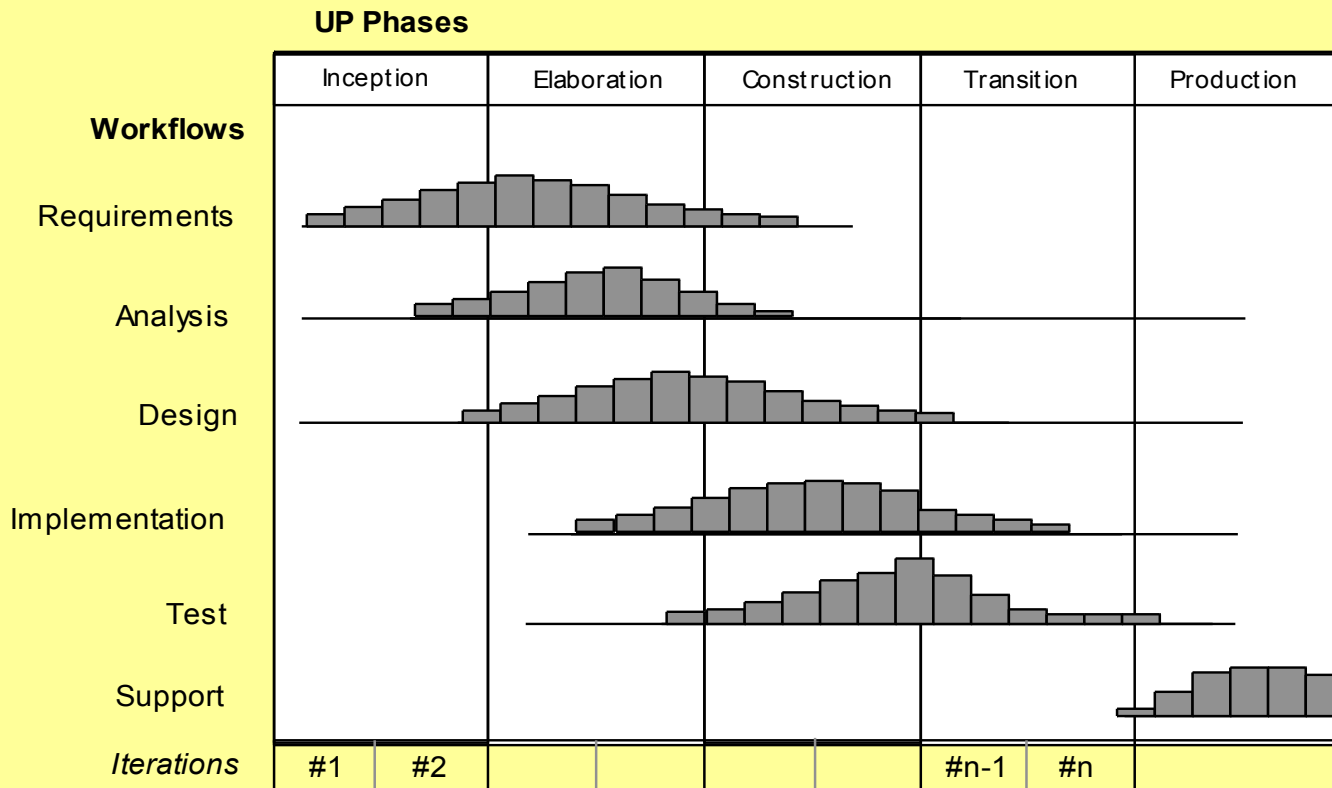
- Modellazione Processi di business
- Requisiti
- Analisi e Progettazione
- Implementazione
- Test
- Rilascio

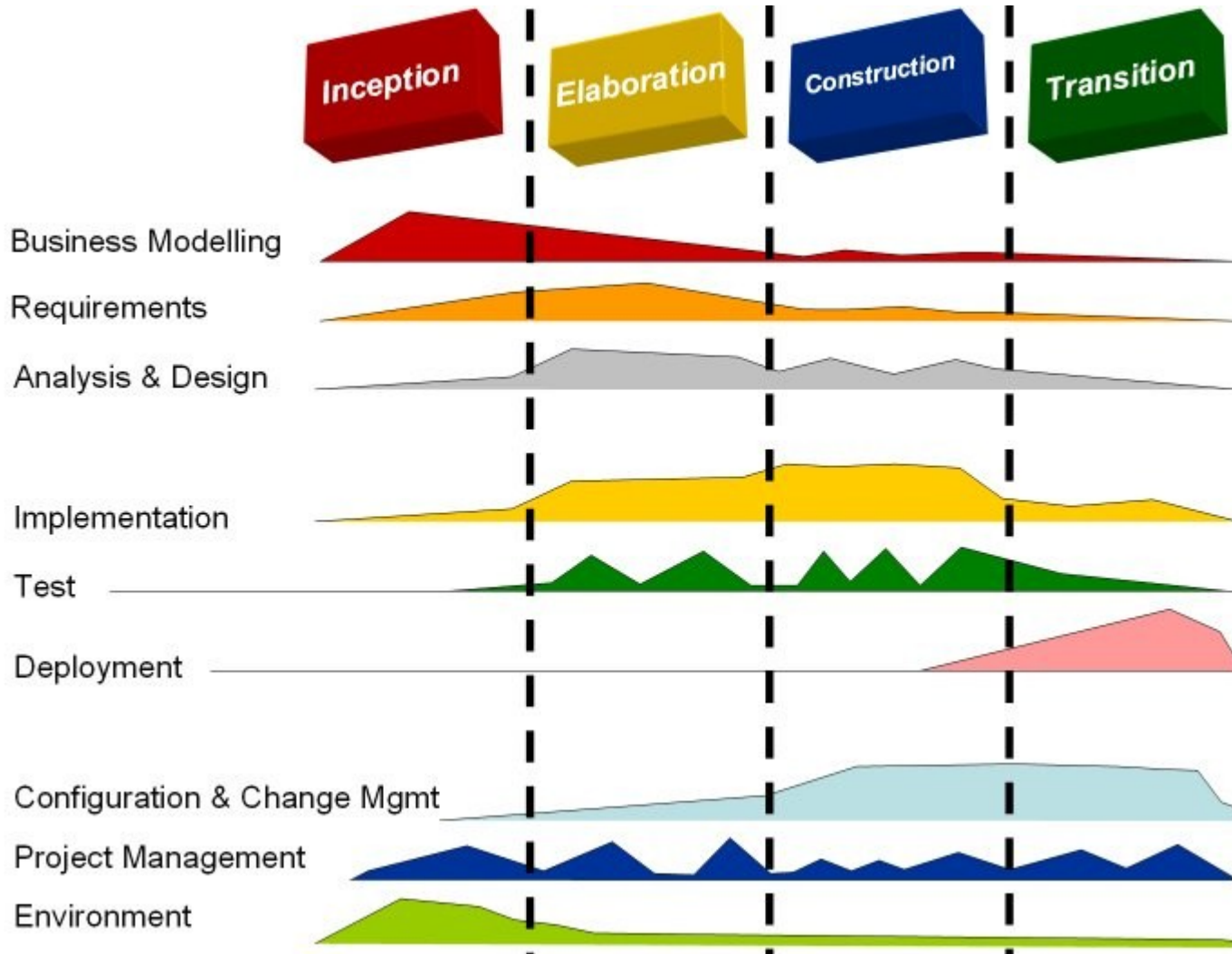
- Gestione della configurazione e delle modifiche
- Gestione del Progetto
- Ambiente

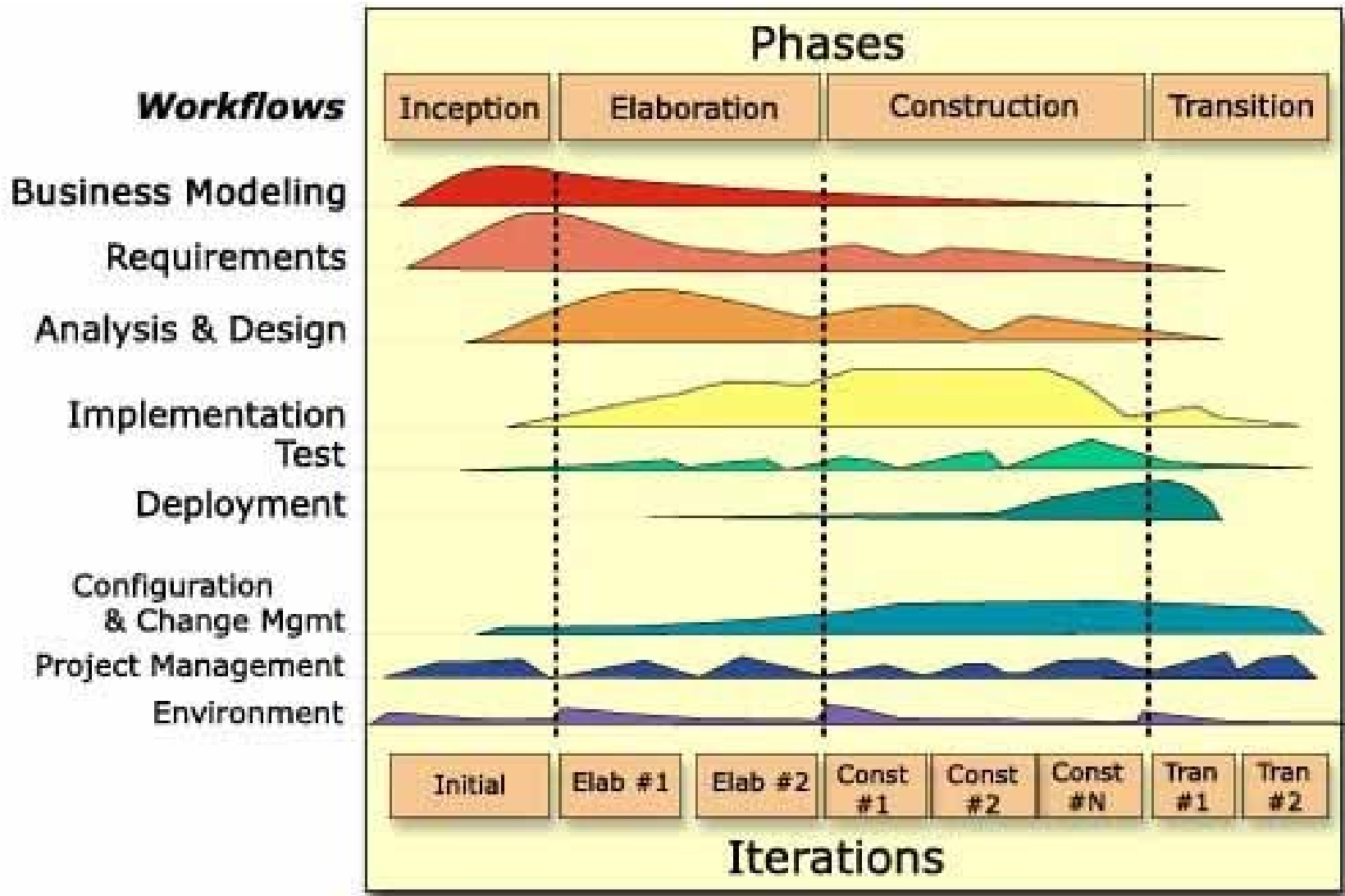
Workflow e Fasi

- **Il RUP non vincola l'esecuzione delle attività a specifiche fasi**
- **Tutti i workflow del RUP possono essere eseguiti in qualunque iterazione del processo.**
- **Ovviamente nelle prime iterazioni gli sforzi si concentreranno sui workflow di modellazione e dei requisiti, nelle successive sulla implementazione e sul test.**

Distribuzione dei workflow per fasi

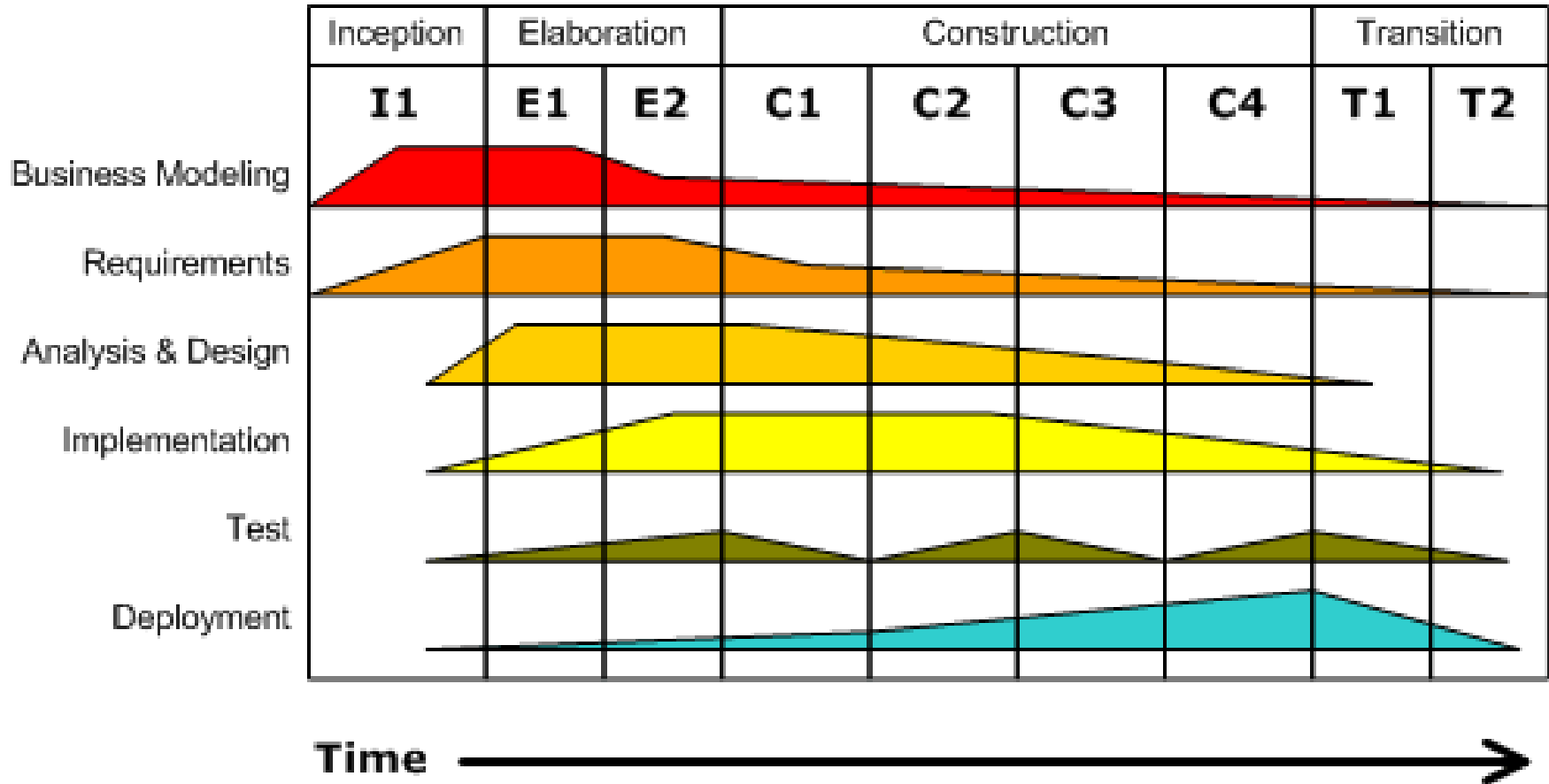


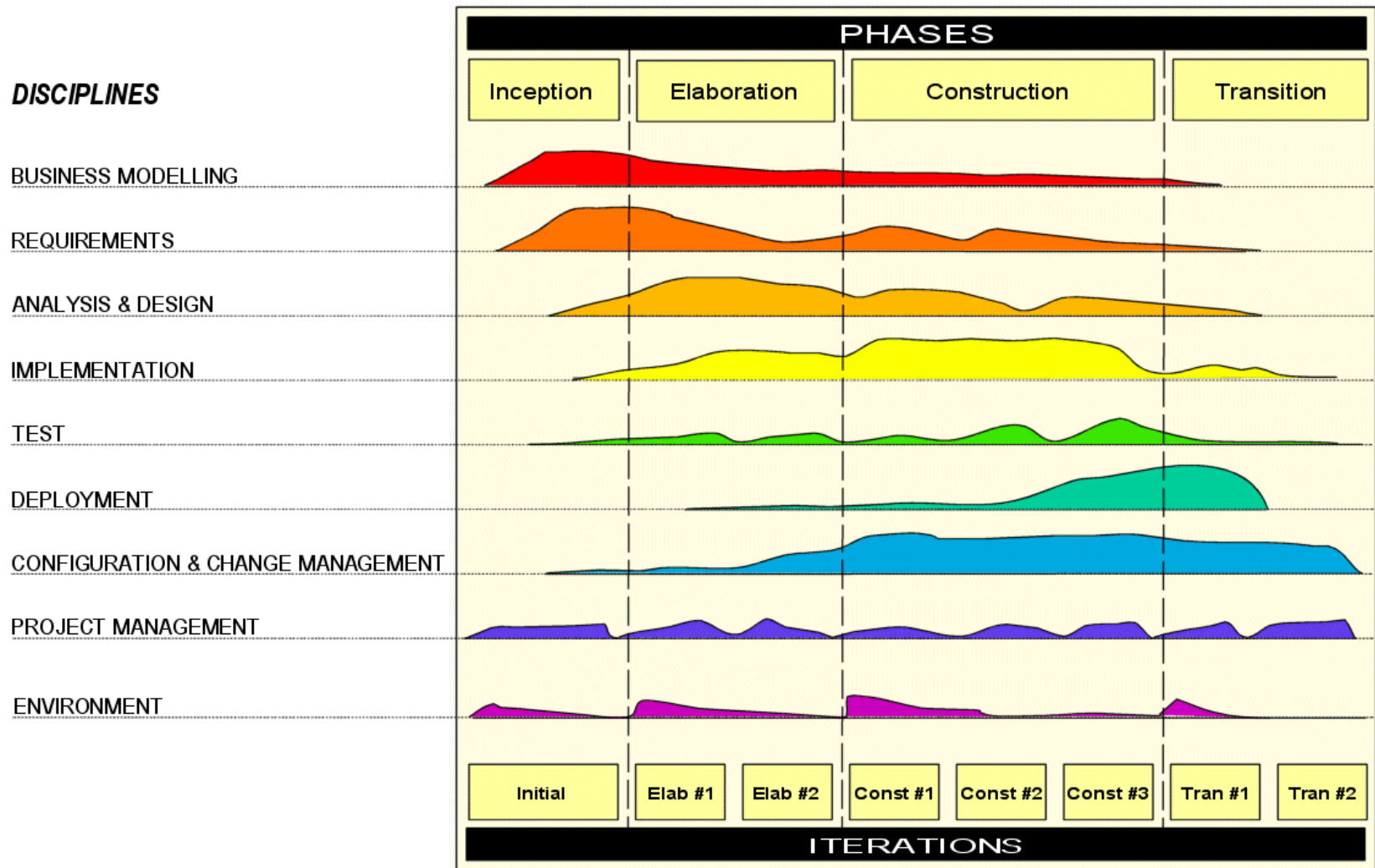




Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.





Le sei pratiche fondamentali del RUP

- **Sviluppare il software iterativamente:**
 - Pianificare gli incrementi in base alle priorità del cliente
- **Gestire i requisiti:**
 - Documentare esplicitamente i requisiti ed i cambiamenti effettuati
- **Usare architetture basate su componenti:**
 - Strutturare l'architettura con un approccio a componenti
- **Creare modelli visuali del software:**
 - Usare modelli grafici UML per rappresentare viste statiche e dinamiche del sistema
- **Verificare la qualità del software:**
 - Verificare l'aderenza a standard di qualità aziendali
- **Controllare le modifiche al software:**
 - Gestire i cambiamenti e le configurazioni del software

Configurazione del RUP

- **Il RUP è un processo generico di sviluppo software:**
 - deve essere istanziato per ciascuna organizzazione e per ciascun progetto specifico, aggiungendo: standard, modelli di documento standard, strumenti, ...
- **Punti di forza:**
 - Separazione di fasi e workflow
 - Le fasi sono dinamiche e vanno pianificate, i workflow sono statici e sono attività tecniche condotte nelle varie fasi
 - Comprende un vasto insieme di linee guida e template per operare con approccio OO e basato su componenti
 - Definisce in modo accurato: Ruoli, Attività, Input e Output delle varie attività

Costi del RUP

- Impatto organizzativo
 - Può portare ad una riorganizzazione completa del lavoro
- Impatto culturale
 - Può comportare una ridefinizione del modo di lavorare
- Costi tecnologici
 - L'utilizzo del processo é agevolato dall'uso di strumenti (tool) specifici (come quelli della suite Rational)
- Costi di avvio
 - Adattamento del processo
 - Divulgazione
 - Inquadramento dei processi esistenti
- **Spesso si rinuncia ad adottare il RUP proprio perchè esso comporta un drastico cambiamento nel modo di lavorare delle persone, che potrebbero reagire (sul breve termine) diminuendo la loro produttività**

Limiti di RUP

- E' un processo limitato ad un singolo progetto: non tratta aspetti che escono dall'ottica di un singolo progetto
- Non é mirato ad alcuna tipologia specifica di applicazioni
 - Deve essere personalizzato, adattandolo alle esigenze e alle caratteristiche dell'organizzazione

Il futuro di RUP: SEMAT



- Da alcuni degli stessi autori di RUP (Ivar Jacobson) viene proposto in questi anni SEMAT
 - Software Engineering Methods And Theory
- Si propone di armonizzare aspetti teorici con gli aspetti pratici necessari alla implementazione pratica dei processi
 - E' una iniziativa OMG
 - Organizza conferenze scientifiche per discuterne gli aspetti e per pubblicare lavori
 - Tutte le notizie sono su semat.org (ci si può liberamente iscrivere)
 - E' di prossima uscita un libro che riassume tutti i progressi fatti nella metodologia

Ulteriori Riferimenti bibliografici

- Booch, Rumbaugh, Jacobson, *Unified Software Development Process*, Addison Wesley 1999
- Kroll, Kruchten, Booch, *The Rational Unified Process Made Easy: a Practitioner's Guide to RUP*, Addison Wesley, 2003
- J.Arlow, I. Neustadt, *UML2 e Unified Process- Analisi e progettazione Object-Oriented*, 2 ed., McGraw Hill 2006
- Ivar Jacobson; Pan-Wei Ng; Paul McMahon; Ian Spence; Svante Lidman, *The Essence of Software Engineering: Applying the SEMAT Kernel*, Addison-Wesley Professional, 2013

Scelta di un Modello di Processo: alcuni aspetti da valutare

- tolleranza del modello ai rischi che si potranno incontrare
- facilità di comunicazione tra sviluppatori e utilizzatori
- stabilità dei requisiti noti; probabilità di esistenza di requisiti (ancora) non noti
- importanza di rilasci 'early' di (parziali) funzionalità
- Intrinseca complessità del problema e delle probabili soluzioni
- (anticipata) conoscenza di frequenza e dimensione di richieste di cambiamento
- maturità dell'applicazione (o del dominio applicativo)
- disponibilità e priorità di fondi
- flessibilità dello scheduling e budget (scadenze per il ricevimento e spesa dei fondi; possibilità di modificare gli incrementi per ottimizzare costi e minimizzare rischi)
- criticità del rispetto di scheduling e budget
- adeguatezza del processo 'istituzionale' e tool di sviluppo dello sviluppatore
- corrispondenza tra organizzazione del management e il modello da adottare

Appendice di approfondimento

- **Standard ISO 12207**

Standard internazionale ISO/IEC 12207

- Lo Standard "ISO/IEC 12207:1995 Standard for Information Technology: Software Life Cycle Processes" vuole stabilire un framework comune che può essere usato da chi produce software.
- L'ISO 12207 stabilisce una struttura comune per il ciclo di vita del software in termini di processi che possono essere impiegati per:
 - (1) acquisire, fornire, sviluppare, eseguire e mantenere software;
 - (2) gestire, controllare, migliorare e monitorare i processi stessi.

- Lo standard ISO12207 ha una struttura che si basa sui seguenti principi :
 - Principi ingegneristici : Ingegneria del Software e dei Sistemi
 - Criteri di collaborazione Cliente Fornitore
 - Tailoring
 - Gestione documentazione (compreso il contratto)
 - Indipendenza da un particolare modello di un ciclo di vita

- Lo standard suddivide i processi in tre categorie:
 - Processi Primari
 - provvedono a gestire le funzioni principali durante il ciclo di vita.
 - Processi Organizzativi
 - sono utilizzati per stabilire, controllare e migliorare la struttura ed i processi stessi
 - Processi di supporto
 - forniscono un supporto ad un altro processo nell'eseguire una specifica funzione
 - Processo di Tailoring (eventuale)
 - serve ad adattare lo standard a particolari progetti cancellando attività inutili, improduttive e non significative o inserendo nuove attività e compiti se necessario.
- Ogni processo è a sua volta suddiviso in attività e ciascuna attività in task.

Processi primari

- Esistono 5 Processi Primari, destinati a acquirenti, fornitori, sviluppatori, operatori e manutentori.
- I processi primari sono:
 - Acquisizione,
 - Fornitura,
 - Sviluppo,
 - Esercizio
 - Manutenzione.

Processo di acquisizione

- Ha attività e compiti relativi a colui che acquista un sistema, un prodotto software o un servizio.
- Le attività di questo processo sono :
 - Inizializzazione (Definizione delle necessità per l'acquisizione)
 - Preparazione proposta richiesta di acquisizione
 - Preparazione e update del contratto
 - Monitoraggio del fornitore
 - Accettazione e compimento

Processo di fornitura

- Interessa il fornitore. In base a tale processo il fornitore dovrà essere in grado di determinare le risorse e le procedure necessarie per la gestione del progetto, eseguire il progetto, distribuire ciò che l'acquirente ha richiesto e se necessario e previsto nel contratto, fornire formazione al personale.
- Le Attività di tale processo sono 7:
 - Inizializzazione
 - Preparazione risposta a proposta di acquisizione
 - Contratto
 - Planning
 - Esecuzione e controllo
 - Revisione e Valutazione
 - Rilascio e Completamento
- Le prime due attività sono svolte prima dell'accordo mentre le ultime 5 dopo l'accordo.

Processo di sviluppo

- E' relativo a tutte le attività e compiti dello sviluppatore del software.
 - Il termine sviluppo è riferito sia allo sviluppo di nuovo software sia a modifiche di software già esistente. Consiste delle attività inerenti:
 - all'implementazione del processo,
 - alle specifiche dei requisiti del sistema,
 - alle specifiche dei requisiti del software,
 - alla progettazione di alto livello del sistema ,
 - alla progettazione di alto livello del software,
 - alla progettazione dettagliata del sistema o del software,
 - alla codifica,
 - al testing di modulo, al testing di sistema ed al testing di integrazione,
 - all'installazione
 - all'accettazione.

Processo di esercizio

- Ha attività e compiti relativi all'erogazione del servizio informatico richiesto, al collaudo, ed alla gestione operativa del sistema.
- Il collaudo consiste nell'esecuzione di test operazionali per verificare l'uso del prodotto software operante nell'ambiente per il quale è stato richiesto.
- Le attività componenti il processo sono:
 - Implementazione del processo
 - Testing operazionali
 - System operation
 - Supporto Utente

Processo di manutenzione

- E' volto a garantire la manutenzione di un prodotto dopo la sua consegna all'acquirente.
- Il processo di manutenzione dovrà essere attivato all'atto di una richiesta di modifica del codice e della documentazione associata, in modo da preservare l'integrità del sistema software o servizio richiesto.
- La manutenzione deve essere pianificata in modo che la conduzione delle attività, di gestione delle modifiche e di analisi del tipo di manutenzione (sia adattativa, che correttiva o di emergenza oppure programmata) sia il più possibile semplice e a costi minimi.

Processo di manutenzione

- Le attività del processo sono:
 - implementazione del processo
 - analisi del problema e delle modifiche
 - implementazione delle modifiche
 - revisione/accettazione della manutenzione
 - migrazione
 - ritiro del software

Processi di supporto e di controllo

- I Processi di Supporto e di Controllo supportano gli altri processi del ciclo di vita del software e contribuiscono al successo e alla qualità di un progetto software.
- I Processi di Supporto e di Controllo sono:
 - Gestione della documentazione,
 - Gestione della configurazione,
 - Assicurazione di Qualità,
 - Verifica,
 - Revisione congiunta (joint review),
 - Revisione (Audit),
 - Validazione,
 - Risoluzione del Problema.

Processo di gestione della documentazione

- Si occupa della registrazione di tutto ciò che è stato prodotto nel processo di produzione: dalle informazioni ai requisiti del sistema dagli input agli output del prodotto.
- I documenti relativi a questo processo presentano tutte le attività relative alla pianificazione, alla progettazione, allo sviluppo, alla manutenzione e alla gestione totale del processo produttivo.
- Le attività del processo sono:
 - implementazione del processo
 - Progetto e sviluppo della documentazione
 - Produzione della documentazione
 - Manutenzione della documentazione

Processo di gestione della configurazione

- Prevede attività e compiti relativi all'applicazione di procedure tecniche ed amministrative per assicurare la completezza, la consistenza e la correttezza degli elementi software.
- In questo processo bisognerà identificare, registrare, analizzare e verificare i possibili cambiamenti dell'elemento software specificando approvazioni e rifiuti (Controllo Configurazioni).
- Si dovranno inoltre preparare i documenti riguardanti lo stato e la storia del software controllato e valutarne la completezza funzionale e fisica.
- Gestione della configurazione fornisce un meccanismo per identificare, controllare e tracciare le versioni di ogni elemento del software.
 - In molti casi le versioni del software precedentemente prodotto e ancora in uso devono subire manutenzione e devono essere anche controllate in maniera accurata.

Processo di gestione della configurazione

- Le attività sono :
 - implementazione del processo
 - identificazione della configurazione
 - controllo della configurazione
 - status e accounting della configurazione
 - valutazione della configurazione
 - gestione delle versioni e dei rilasci

Processo di assicurazione della qualità

- Ha lo scopo di prevenire il presentarsi dei problemi, di individuarli quando si presentano, di identificarne le cause e di trovarvi rimedio.
- L'assicurazione della qualità comprende tutte le attività della funzione manageriale, la quale determina le politiche per la qualità, gli obiettivi e le responsabilità, e le implementa attraverso:
 - la pianificazione della qualità
 - il controllo della qualità
 - il miglioramento della qualità nell'ambito del sistema della qualità.

Processo di assicurazione della qualità

- L'assicurazione di qualità riguarda sia il prodotto sia il processo.
 - Il prodotto deve essere conforme ai requisiti ed aderire ai vari piani di progetto e sviluppo al fine di poter possedere gli specificati attributi di qualità.
 - Il processo e le proprie pratiche interne di ingegnerizzazione, oltre ad essere conforme a quanto specificato ed aderire alla pianificazione, deve rispettare gli standard utilizzati per implementare il processo stesso.
- Le attività del processo sono :
 - implementazione del processo
 - assicurazione qualità del prodotto
 - assicurazione qualità del processo
 - assicurazione del sistema qualità

Processo di verifica

- E' costituito da attività e compiti atti a determinare se i requisiti del prodotto, sistema o servizio software sono completi e corretti e se essi sono soddisfatti.
- Tale processo deve essere integrato nei processi di sviluppo, fornitura e manutenzione.
 - Nel caso in cui la verifica venga effettuata da terzi viene definita Processo di Verifica Indipendente.
- Le attività del processo sono:
 - implementazione del processo
 - Verifica
 - Contract verification
 - Process verification
 - Requirements verification
 - Design verification
 - Code verification
 - Integration verification
 - Documentation verification

Processo di validazione

- Per determinare se il sistema o il prodotto finale offerto soddisfa l'utilizzo per cui è stato creato.
- Se esso viene svolto da terzi viene definito Processo di validazione Indipendente.
- Le attività del processo sono:
 - implementazione del processo
 - validazione

Processo di revisione congiunta (joint review)

- Serve a valutare se lo stato e i prodotti di un'attività o di un progetto siano appropriati a quel processo di produzione.
- La revisione congiunta avviene tra gli stessi componenti di un'organizzazione se si deve revisionare un elemento software (peer Review), oppure tra il fornitore e l'acquirente se si deve revisionare l'intero prodotto.
- La revisione congiunta può essere svolta sia a livello tecnico che a livello organizzativo.

Processo di revisione congiunta (joint review)

- In questo processo si procede alla pianificazione delle revisioni congiunte che avverranno sulla base di un calendario di incontri fissato all'inizio del progetto.
- Le attività del processo sono:
 - implementazione del processo
 - revisione della gestione del progetto
 - revisione tecnica del progetto

Processo di Audit

- Relativo alla determinazione della verifica della conformità ai requisiti, ai piani ed ai contratti.
- Il processo di Revisione dovrà essere condotto da personale che non ha direttamente partecipato allo sviluppo dei prodotti, dei servizi o dei sistemi che si revisionano.
- Le attività della revisione dovranno essere condotte rispettando i piani e il contratto.
- Le attività del processo sono:
 - implementazione del processo
 - Audit (Revisione)

Processo di risoluzione di problemi

- Scopo di questo processo è fornire tempestivamente le risorse per assicurare la risoluzione dei problemi appena essi sono rilevati.
- Quando si verifica o si rileva un problema durante le fasi di sviluppo, di manutenzione, o di altri processi è necessario documentarlo dettagliatamente, accertarne l'effettiva esistenza per poi risolverlo (anche in diversi modi).
- Prima però di risolverlo sarà opportuno valutarne la priorità e l'importanza rispetto agli altri utilizzando un parametro di riferimento.
- Le attività del processo sono:
 - implementazione del processo
 - risoluzione del problema

Processi organizzativi

- I Processi Organizzativi sono impiegati in un organizzazione per stabilire e implementare una struttura a supporto dei processi del ciclo di vita e del personale e per il continuo miglioramento dei processi e della struttura stessa.
- I Processi Organizzativi sono così suddivisi:
 - Gestione (Management),
 - Gestione dell'infrastruttura,
 - Miglioramento
 - Addestramento.

Processo di gestione o Project Management

- Il processo contiene tutte le generiche attività e tasks, che possono essere impiegate all'interno del processo di produzione per gestire i rispettivi "processi".
- Il management è responsabile per la gestione del prodotto, la gestione del progetto e della gestione dei tasks applicabili nei processi primari o di supporto.
- Le attività del processo sono:
 - Iniziazione e definizione dello scopo
 - Pianificazione
 - Esecuzione e controllo
 - Revisione e valutazione
 - Chiusura

Processo di gestione dell'infrastruttura

- Il processo di gestione delle infrastrutture è un processo che utilizza metodi, tecniche e strumenti il più possibile sinergici e coerenti fra loro per supportare i diversi processi.
- Tale processo deve assicurare lo svolgimento di tutte le attività del processo di produzione.
- Le attività del processo di configurazione delle infrastrutture sono:
 - implementazione del processo
 - costituzione dell'infrastruttura
 - manutenzione dell'infrastruttura

Processo di miglioramento

- I principali scopi del processo sono il monitoraggio, il controllo, la realizzazione del miglioramento del processo.
- Le attività del processo sono :
 - Costituzione del processo
 - Controllo e valutazione del processo
 - Miglioramento del processo

Processo di addestramento (Training)

- Tale processo è inerente alla formazione e addestramento del personale.
- I processi primari dipendono dal know how e dalle capacità del personale sia dell'azienda fornitrice sia di quella acquirente.
 - Il personale deve essere addestrato e formato costantemente.
- Le attività del processo sono :
 - implementazione del processo
 - sviluppo del materiale di formazione
 - pianificazione della realizzazione della formazione.

Processo di Tailoring

- Tale processo, nell'istanziamento di uno specifico processo, consiste o nella cancellazione di processi, attività o tasks presenti nello standard e nel caso specifico ritenuti inutili oppure nell'aggiunta di processi, attività o tasks non presenti nello standard e necessari per il caso specifico.
- Le attività sono:
 - identificazione dell'ambiente di sviluppo
 - individuare gli ingressi
 - individuazione dei processi, delle attività e dei task necessari
 - documentazione della decisione di eliminare o aggiungere processi, attività o tasks.