

Unità di apprendimento 5

Gestione di progetti
informatici



The background of the slide is a blue gradient with a pattern of binary code (0s and 1s) floating around. On the left side, there is a partial view of a laptop screen and keyboard.

Unità di apprendimento 5

Lezione 7

Le fasi nei modelli di sviluppo
dei progetti informatici

Generalità

- L'ingegneria del software, nata negli anni '60 per formalizzare e definire le tecniche e le strategie finalizzate all'ottimizzazione e al miglioramento della produzione dei programmi, ha indicato con “ciclo di vita del software” l'insieme di tutte le attività connesse alla “produzione di un programma” e le ha indicate come illustrato di seguito.

Generalità

- Possiamo sintetizzare il **ciclo del processo di produzione del software** in tre fasi (o macrofasi):
 - progetto
 - sviluppo
 - manutenzione

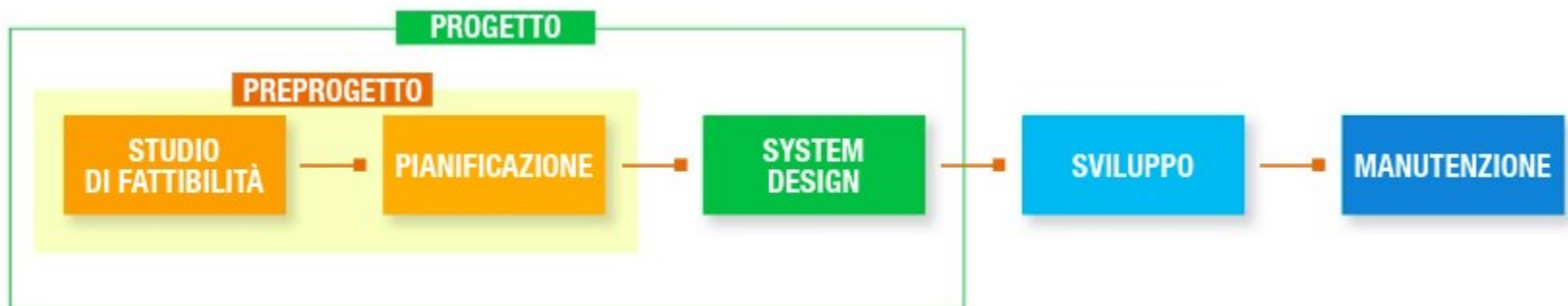


Generalità

- Il progetto comprende le attività che permettono di definire lo schema dove sono elencati tutti gli eventi e le attività coinvolte per il raggiungimento degli obiettivi ed i risultati prefissi, scomponendo e definendo le varie fasi e le relazioni di precedenza tra di esse.
- La scomposizione in fasi è diversa a seconda del modello di sviluppo adottato e ciascuna fase prevede un elenco di possibili attività ed eventi caratteristici.

Generalità

- In ciascun **modello**, però possiamo individuare:
 1. Il **preprogetto**, composto dallo **studio di fattibilità** e dalla **pianificazione**
 2. Il **progetto** vero e proprio (design), dove la fase di preprogetto è comune a tutti i modelli di sviluppo



Generalità

- Il **ciclo di vita del software** (**software lifecycle**) è l'insieme di tutte le attività connesse alla “**produzione di un programma**” e descrive quindi l'insieme di tutte le fasi che un progetto attraversa, dallo studio di fattibilità al rilascio del prodotto finito.

Generalità

- Il passaggio da una fase all'altra nell'ambito dello stesso progetto comporta solitamente una forma di trasferimento tecnico o passaggio di consegne (**deliverable**)
- I **deliverable** ottenuti da una fase vengono generalmente analizzati per verificarne la completezza e l'accuratezza, per essere poi approvati prima che si proceda con la fase successiva del lavoro.

Generalità

- Le principali fasi presenti in ogni ciclo di vita del progetto software sono:
 1. Pianificazione del sistema
 2. Analisi
 3. Progettazione
 4. Codifica (realizzazione)
 5. Test e debug
 6. Installazione, verifica e collaudo
 7. Manutenzione

Generalità

- Le principali fasi presenti in ogni ciclo di vita del progetto software sono:

1. **Pianificazione del sistema**
2. Analisi
3. Progettazione
4. Codifica (realizzazione)
5. Test e debug
6. Installazione, verifica e collaudo
7. Manutenzione

Denominata anche **analisi del sistema**, individua le parti del sistema da realizzare con il software

Generalità

- Le principali fasi presenti in ogni ciclo di vita del progetto software sono:

1. Pianificazione del sistema
2. **Analisi**
3. Progettazione
4. Codifica (realizzazione)
5. Test e debug
6. Installazione, verifica e collaudo
7. Manutenzione

In questa fase vengono stabilite le **caratteristiche** dell'applicazione informatica (si decide cosa il progetto deve fare)

Generalità

- Le principali fasi presenti in ogni ciclo di vita del progetto software sono:
 1. Pianificazione del sistema
 2. Analisi
 3. **Progettazione**
 4. Codifica (realizzazione)
 5. Test e debug
 6. Installazione, verifica e collaudo
 7. Manutenzione

Incorpora tutte le attività connesse alla **realizzazione** del **sistema**, strutture dati e gestione dei problemi

Generalità

- Le principali fasi presenti in ogni ciclo di vita del progetto software sono:
 1. Pianificazione del sistema
 2. Analisi
 3. Progettazione
 4. **Codifica (realizzazione)**
 5. Test e debug
 6. Installazione, verifica e collaudo
 7. Manutenzione

Si scrive e si compila il **codice sorgente**, codificandole istruzioni in linguaggi di programmazione

Generalità

- Le principali fasi presenti in ogni ciclo di vita del progetto software sono:
 1. Pianificazione del sistema
 2. Analisi
 3. Progettazione
 4. Codifica (realizzazione)
 5. **Test e debug**
 6. Installazione, verifica e collaudo
 7. Manutenzione

In questa fase occorre verificare l'**assenza di errori** eseguendo un **test**. In caso di errori, si innesca la fase di **debug**

Generalità

- Le principali fasi presenti in ogni ciclo di vita del progetto software sono:
 1. Pianificazione del sistema
 2. Analisi
 3. Progettazione
 4. Codifica (realizzazione)
 5. Test e debug
 6. **Installazione, verifica e collaudo**
 7. Manutenzione

Consegna del software al cliente, **verifica** delle specifiche del software e **collaudo** del sistema.

Generalità

- Le principali fasi presenti in ogni ciclo di vita del progetto software sono:
 1. Pianificazione del sistema
 2. Analisi
 3. Progettazione
 4. Codifica (realizzazione)
 5. Test e debug
 6. Installazione, verifica e collaudo
 7. **Manutenzione**

Fase di supporto al sistema dopo la consegna al cliente.
Può essere
correttiva,
adattativa,
migliorativa.

Modelli di sviluppo

- I principali modelli di sviluppo, si suddividono in due categorie
 - Modelli di sviluppo tradizionali.
 - Per esplorazione.
 - Build and Fix.
 - Modello a cascata.
 - Modello a prototipazione rapida – RAD.
 - Sviluppo agile o iterativo incrementale.
 - XP (eXtreme Programming).
 - RUP (Rational Unified Process).
 - DSDM (Dynamic Software Development Method).
 - UP – Metodo di processo unificato.



How the customer explained it



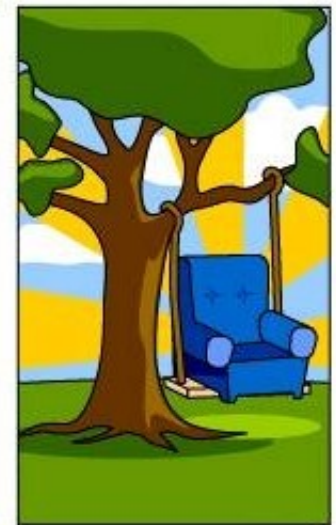
How the Project Leader understood it



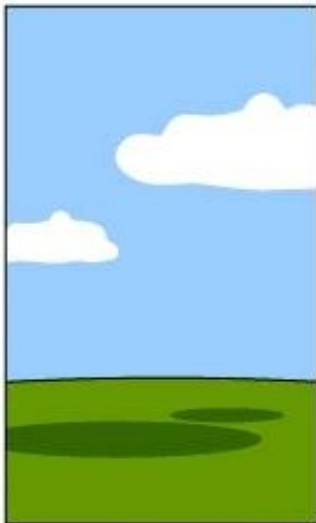
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



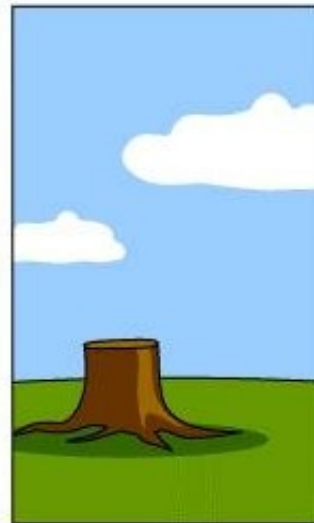
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

Processo per esplorazione e sviluppo Build and Fix

Il primi modelli di sviluppo che ricordiamo sono quelli che **non hanno un modello vero e proprio**.

- ▶ Nel modello di **sviluppo per esplorazione** i progettisti/sviluppatori lavorano con i clienti: a partire dall'individuazione delle specifiche iniziali si arriva, per fasi di trasformazioni successive (evoluzione), fino al sistema software finale, aggiungendo “strada facendo” le successive richieste del cliente che “affiorano” man mano che il prodotto prende forma.
- ▶ In quello che viene definito **sviluppo Build and Fix** si affronta il progetto senza predisporre alcuna documentazione (o quasi), si procede alla realizzazione di un primo prodotto approssimativo e lo si propone al cliente, che interviene con richiesta di modifiche e/o integrazioni fino a che non viene realizzato il prodotto che desidera.

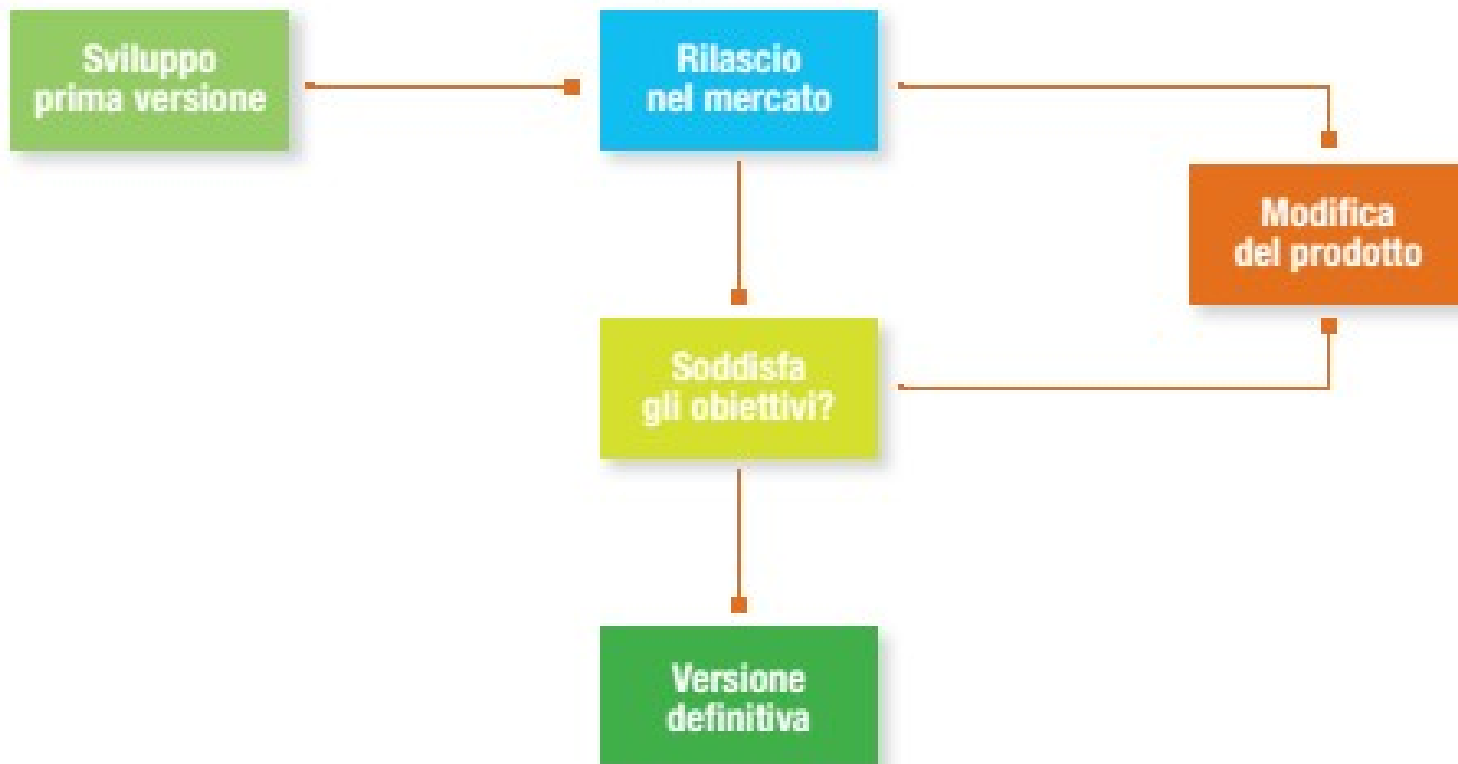
Processo per esplorazione e sviluppo Build and Fix

Il primi modelli di sviluppo che ricordiamo sono quelli che **non hanno un modello vero e proprio**.

- ▶ Nel modello di **sviluppo per esplorazione** i progettisti/sviluppatori lavorano con i clienti: a partire dall'individuazione delle specifiche iniziali si arriva, per fasi di trasformazioni successive (evoluzione), fino al sistema software finale, aggiungendo “strada facendo” le successive richieste del cliente che “affiorano” man mano che il prodotto prende forma.
- ▶ In quello che viene definito **sviluppo Build and Fix** si affronta il progetto senza predisporre alcuna documentazione (o quasi), si procede alla realizzazione di un primo prodotto approssimativo e lo si propone al cliente, che interviene con richiesta di modifiche e/o integrazioni fino a che non viene realizzato il prodotto che desidera.

Entrambi i modelli portano alla realizzazione di **codice pessimo**, senza strutturazione, difficilmente comprensibile e modificabile, probabilmente non corretto, con tempi lunghi di sviluppo a causa della mancanza di una visione d'insieme. Sono metodi che possono essere usati solo per sistemi di piccole dimensioni o di singole parti di sistemi più complessi. !

Processo per esplorazione e sviluppo Build and Fix



Modello a cascata



Modello a cascata



In questo modello, ogni singolo passo viene svolto solo se i passi che lo hanno preceduto sono stati completati: il numero, il contenuto e la denominazione delle fasi può variare da un'organizzazione all'altra e da un progetto all'altro. !

Modello a cascata

WBS	ATTIVITÀ	OUTPUT
1	PIANIFICAZIONE	
1.1	Studio fattibilità	Un documento che presenta diversi scenari e soluzioni e una valutazione dell'impatto del nuovo prodotto sull'organizzazione in termini di costi previsti e benefici.
1.2	Valutazione alternative	Un documento che riporta l'analisi di mercato e le possibili alternative allo sviluppo ex novo del prodotto.
2	ANALISI	
2.1	Analisi del contesto	Un documento che riporta le esigenze dell'utente e le caratteristiche del nuovo sistema descritte dal progettista in modo da essere comprese e condivise dalle parti.
2.2	Analisi dei requisiti	<ul style="list-style-type: none">• Un documento di analisi dei requisiti che descrive le caratteristiche del sistema e le esigenze di ogni futuro utilizzatore.• Un manuale utente: contiene in linea di massima le indicazioni operative per l'uso del nuovo sistema e le modalità di utilizzo da parte degli utenti.• Piano temporale: non è indispensabile che sia fatto e che sia preciso, ma è preferibile allegare un Gantt di massima dove si indicano i tempi di rilascio e l'ipotesi di funzionamento a regime.
2.3	Firma contratto	Approvazione del documento di analisi e stipula del contratto.

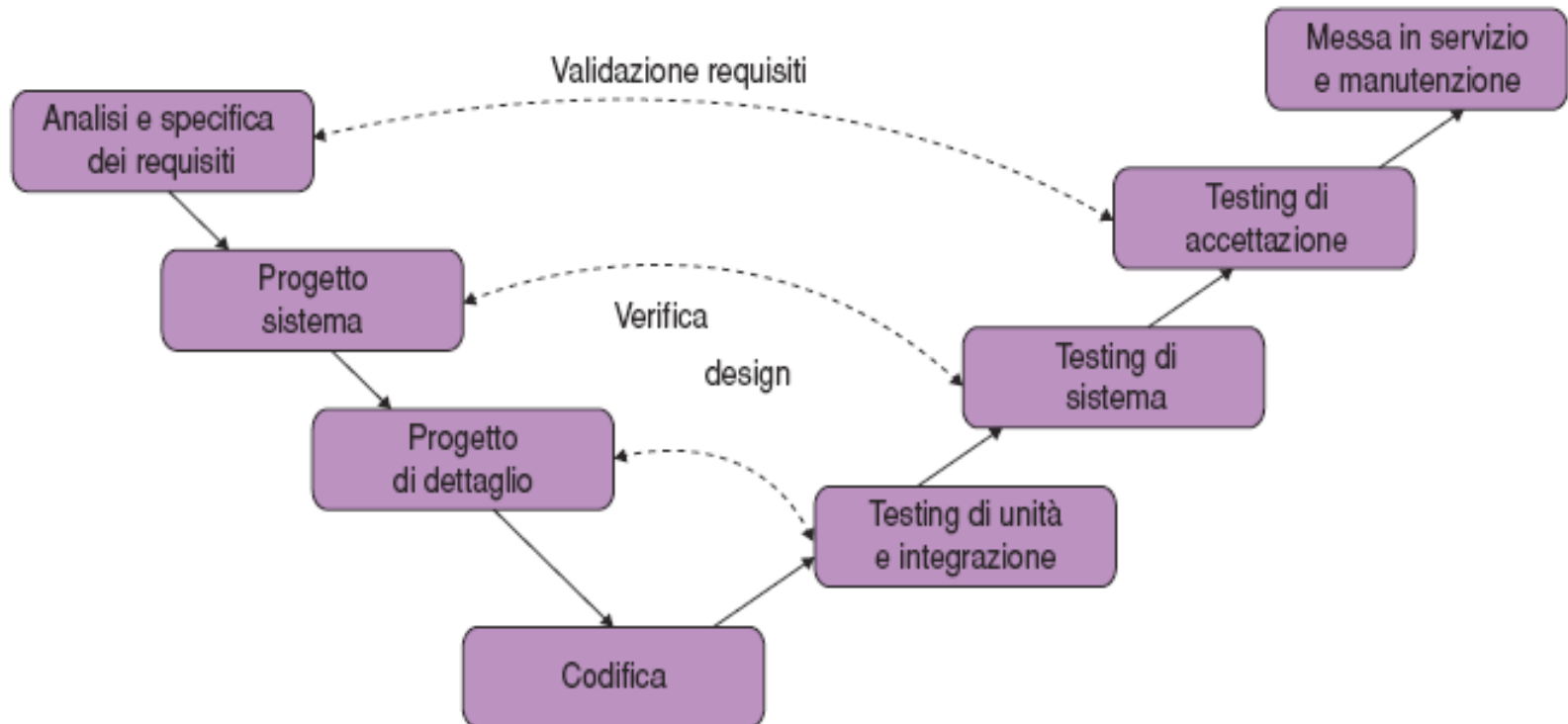
Modello a cascata

3	PROGETTAZIONE	
3.1	Progetto architettura	Progetto generale e definizione delle interfacce.
3.1	Progetto di dettaglio	Progetto dei singoli moduli.
3.3	Stesura del progetto	Integrazione dei moduli e stesura del progetto definitivo.
4	CODIFICA/REALIZZAZIONE	
4.1	Sviluppo singoli moduli	Codice dei singoli moduli in linguaggio di programmazione.
4.2	Integrazione dei moduli	Configurazione di integrazione dei singoli moduli.
4.3	Delivery	Il prodotto finito.
5	TEST/COLLAUDO	
5.1	Test singoli	Elenco dei test eseguiti per ciascun modulo.
5.2	Test di integrazione	Elenco dei test eseguiti per l'integrazione di ciascun modulo.
5.3	Verbalizzazione	Verbale di collaudo.
6	INSTALLAZIONE	
6.1	Installazione software	Il prodotto installato e configurato.
6.2	Integrazione di sistema	Piano di formazione operativa degli utenti.
6.3	Verbale di consegna	Verbale di consegna.
7	MANUTENZIONE	
7.1	Manutenzione adattativa	Il prodotto aggiornato.
7.2	Manutenzione evolutiva	Il prodotto aggiornato.

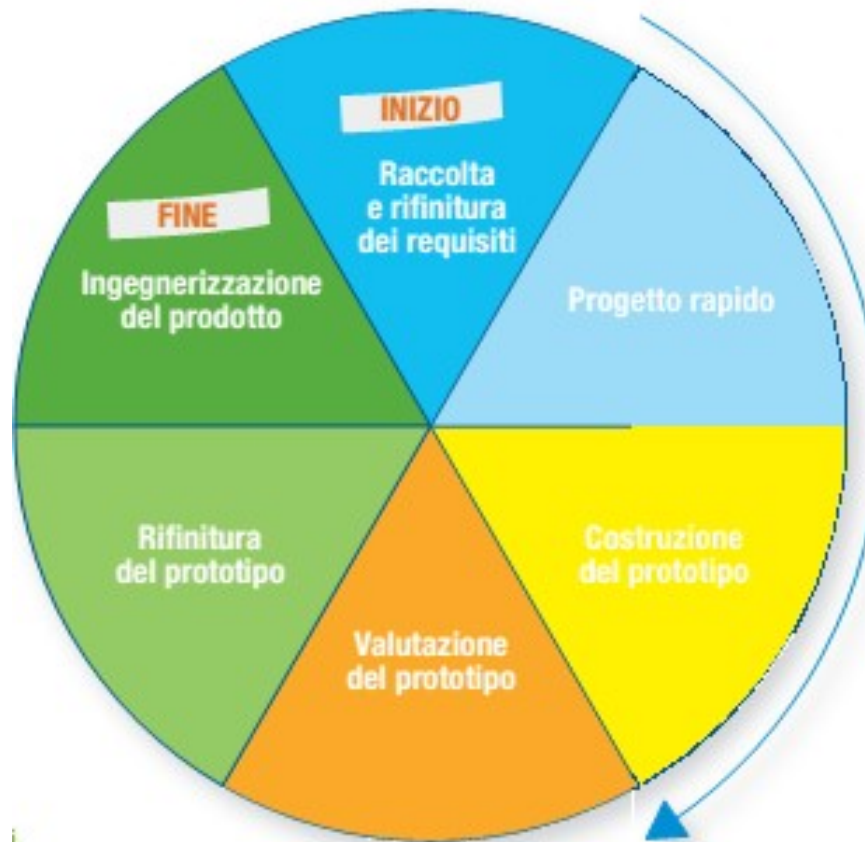
Modello a cascata

- I principali difetti del modello a cascata sono:
 - Assenza di flessibilità;
 - Difficoltà nella risoluzione dei malfunzionamenti;
 - Tempi elevati per l'implementazione di nuove funzionalità;
 - Tempi non rispettati;
 - Notevole aumento dei costi.

Modello a V



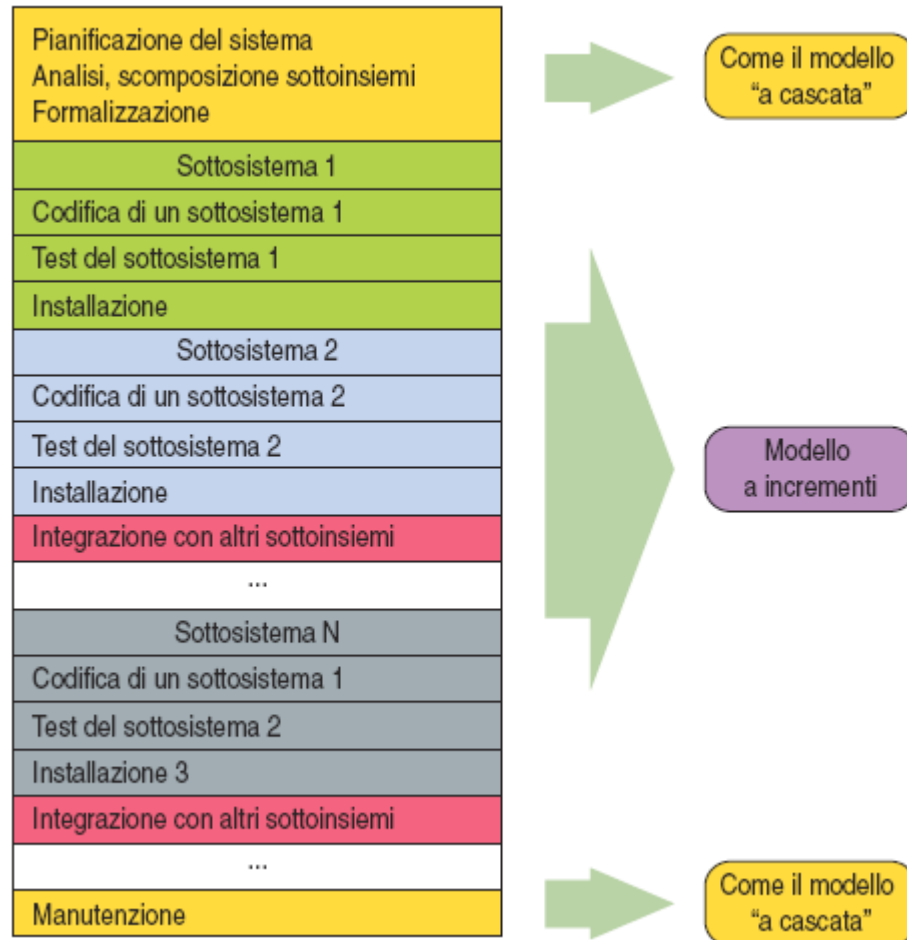
Modello a prototipazione rapida



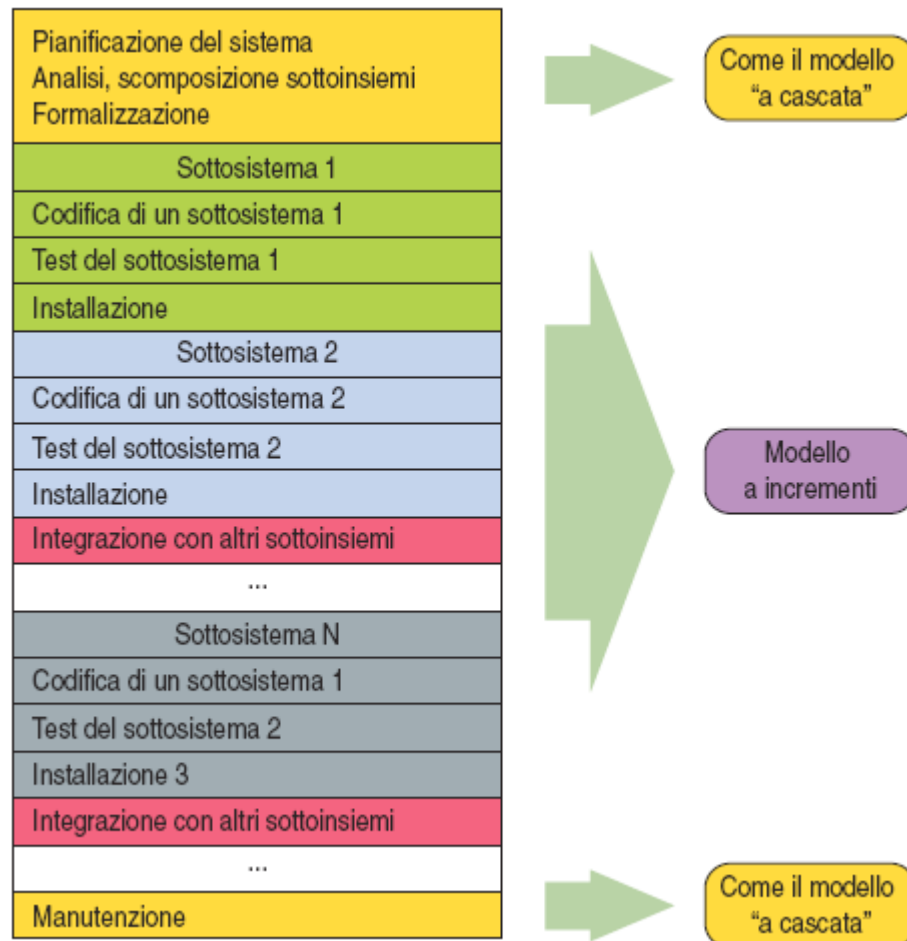
Modello a prototipazione rapida

- I principali difetti del modello a prototipazione rapida sono:
 - La valutazione del prototipo spesso si concentra su aspetti marginali (come il layout) e non sugli aspetti fondamentali che il prototipo voleva evidenziare;
 - Il prototipo viene spesso usato come base di sviluppo e non come “chiarificatore dei requisiti”.

Modello incrementale

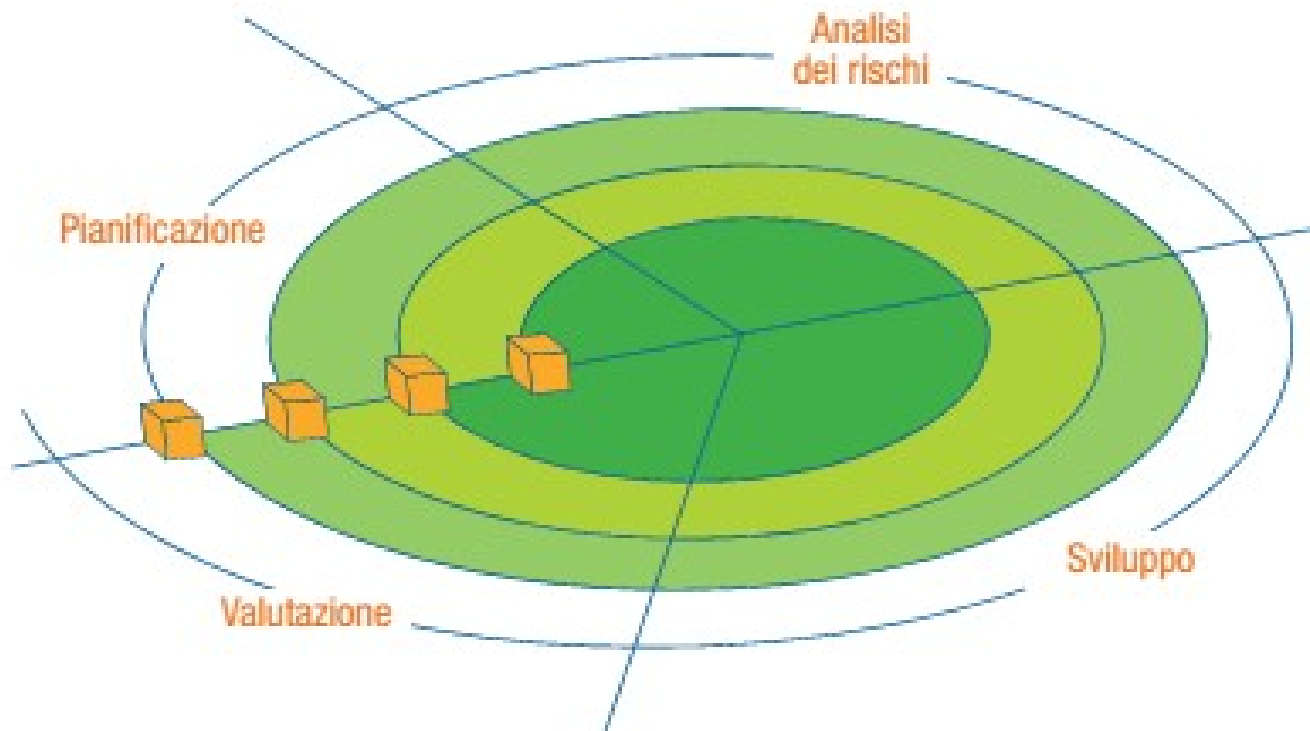


Modello incrementale



I principali problemi sono localizzati nel **riutilizzo** dei sottosistemi e **nell'integrazione** di tali sottosistemi per evitare un aumento dei costi di sviluppo.

Modello a spirale



Metodologie agili

- Le metodologie Agili si focalizzano sul raggiungimento di un risultato alla volta, piccolo e ben definito.
- L'idea di base è che queste metodologie non sono predettive, bensì **adattive**, ossia si adattano all'evoluzione dei requisiti del cliente.

Metodologie agili

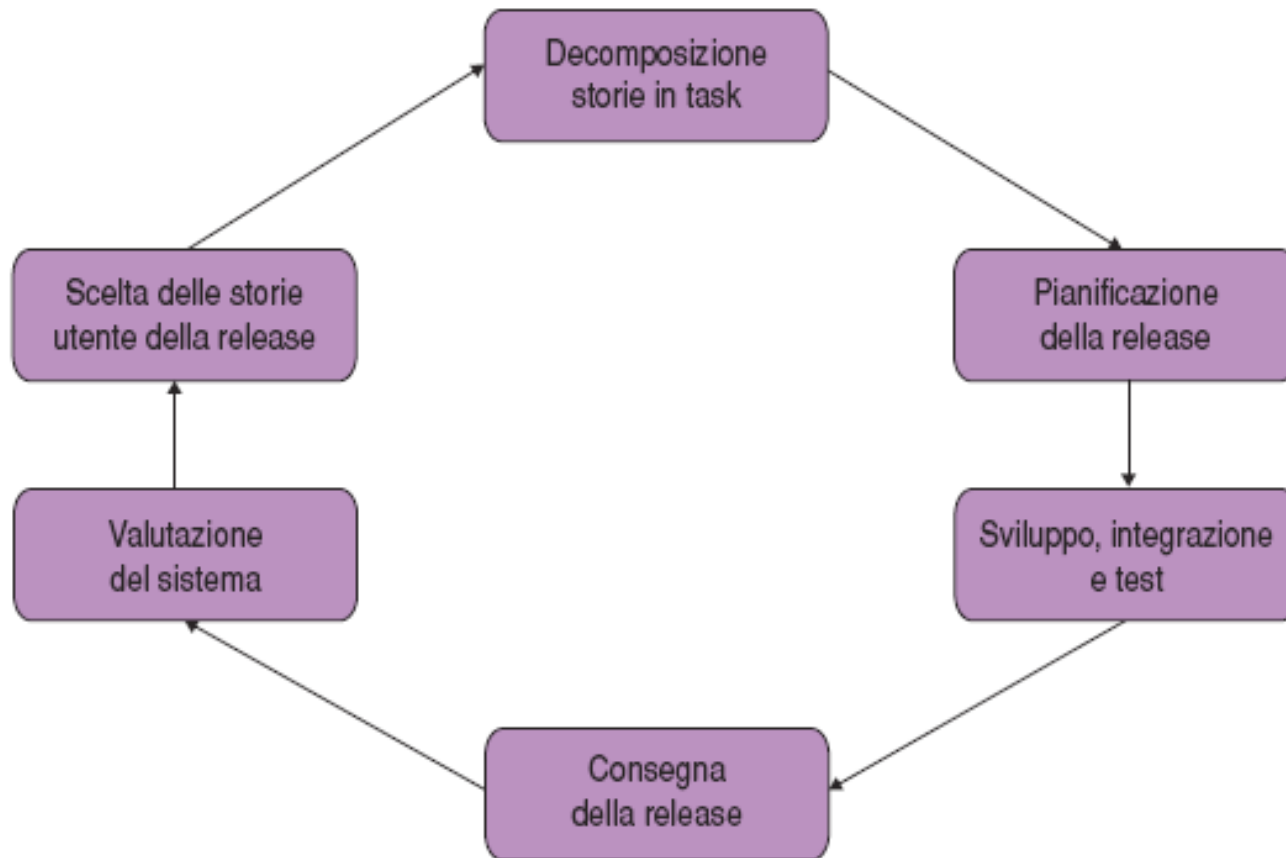
- Le tre principali caratteristiche delle metodologie Agili, sono:
 - Iterattività ed incrementalità
 - Rilasci frequenti
 - Testing
- Sono indicate in progetti dove i requisiti utente cambiano continuamente.

Metodologie agili

Tra i metodi agili ricordiamo:

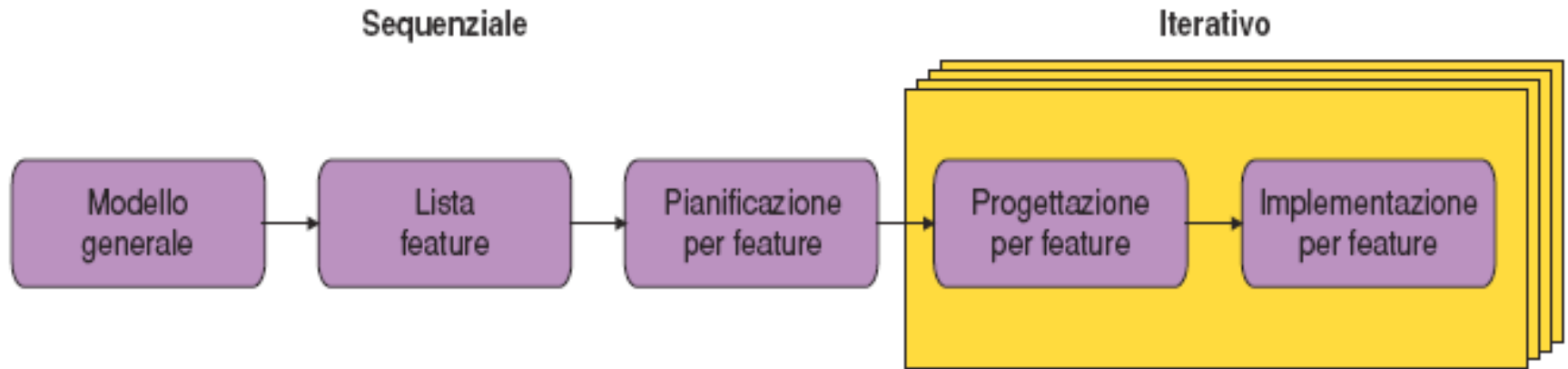
- Extreme Programming XP (Beck & Cunningham)
- Feature Driven Development FDD (De Luca & Coad)
- SCRUM (Sutherland)
- Crystal (Cockburn)
- DSDM (Dynamic System Development Method)
- Lean Software Development (Poppendieck)

Extreme Programming



Feature Driven Development

FDD



Conclusioni

MODELLI	PUNTI DI FORZA	PUNTI DEBOLI
Build & Fix	Adatto a progetti piccoli che non necessitano manutenzione.	Non adatto a progetti complessi; non ripetibile; legato alla specifica competenza.
A cascata (Waterfall)	Comprensibile, controllabile, adeguato ai contratti, milestone chiare.	Difficile gestione dei requisiti e rapporto con cliente, ritarda e rimanda l'individuazione dei problemi.
RAD/Incrementale	Produce in anticipo prime versioni del prodotto.	Necessita di capacità nella pianificazione e nella gestione delle iterazioni.
Spirale	Consente di utilizzare i feedback, migliore capacità di gestire la variabilità dei requisiti.	Necessita di grande disciplina e motivazione; non facilmente comprensibile dal cliente, generalmente non adeguato ai contratti.
Agile	Adatto a progetti particolarmente innovativi e con requisiti non ben definiti.	Necessita di grande disciplina e motivazione; non facilmente comprensibile dal cliente, generalmente non adeguato ai contratti.
RUP	Approccio iterativo; uso di componenti; modellazione grafica.	Generico, poco chiaro.