

## Il ciclo di vita del software(3)

**Ciclo di vita di un software:** Evolve nel tempo e raggiunge **stati** tramite **transizioni** scatenate da attività che hanno il fine di far avanzare il sw. Si divide in 4 fasi:

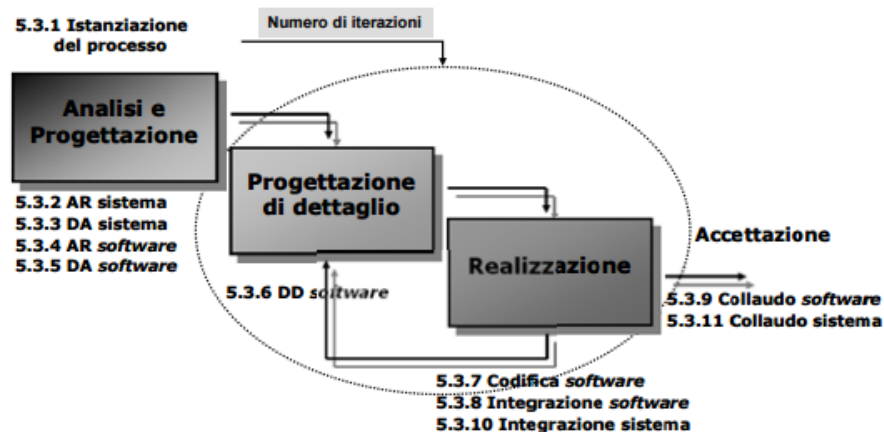
- **Concezione**, quando qualcuno pensa che ci sia (o abbia) bisogno di qualcosa;
- **Sviluppo**;
- **Utilizzo**;
- **Ritiro**.

**Fase:** DefPeriodo di tempo contiguo che cattura transizioni o segmenti di ciclo di vita

La scelta del ciclo di vita pone vincoli sulla pianificazione del progetto, è indipendente da metodi e strumenti di sviluppo.

Tipi di ciclo di vita:

- **Cascata:** successione di fasi rigidamente sequenziali con dipendenze causali tra di loro, non ammette ritorno a fasi precedenti. (Pre e Post solide e definite all'origine, modello che non spreca, al tempo t so dove sono). Modello **Document driven** (documenti che spiegano la fiducia nelle scelte fatte), non sono previsti prototipi.  
Difetti: eccessivamente rigido, non ammette cambiamenti ai requisiti in corso d'opera, il committente vede solo l'opera finita.
- **Incrementale:** modello iterativo, potenzialmente distruttivo (avanzo nel tempo ma non nei risultati). In questo modello procedo per fasi che possono anche essere completamente separate le une dalle altre, ma a differenza del waterfall posso integrare anche le piccole parti. Produco valore ad ogni incremento, ed ogni incremento riduce il rischio di fallimento; le funzionalità essenziali inoltre sono sviluppate nei primi incrementi. Qui si prevedono rilasci multipli e successivi (base stabile ed aggiungo successivamente). Non si ripetono mai l'analisi e la progettazione architeturale.

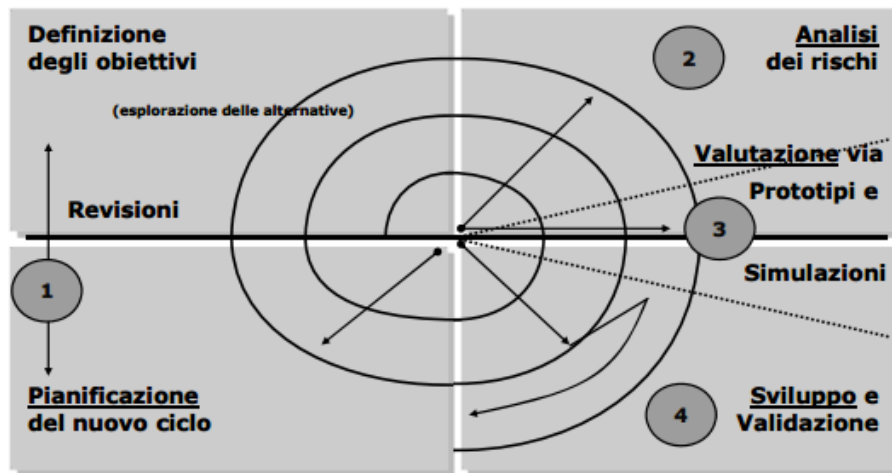


Non ritorno mai sul progetto generale.

- **Iterativo:** comporta il rischio di non convergenza all'obiettivo, tuttavia consentono maggior capacità di adattamento. Ogni iterazione comporta un ritorno all'indietro, nella direzione opposta rispetto al tempo.

- **Evolutivo:** insegue il futuro non ritirando il passato, ho tante attività concorrenti. Questo modello lo attua chi può sostenere molte versioni in parallelo (e quindi ha buone capacità finanziarie), infatti ogni fase ammette iterazioni multiple e parallele. Si può anche modificare la base ATT.
- **Spirale:** è composto da cicli interni rapidi e ripetuti, mentre quelli esterni aderiscono a qualsiasi altro modello std. Comporta un miglior controllo dei rischi di progetto, infatti pone molta attenzione sugli aspetti gestionali (pianificazione delle fasi, analisi dei rischi), è un modello **risk driven**. Prevede 4 attività principali
  - Definizione degli obiettivi
  - Analisi dei rischi
  - Sviluppo e validazione
  - Pianificazione

Utilizzato solo per progetti innovativi



I quadrati rappresentano dei macrostadi.

- **Componenti:** si basa sul riuso dei componenti già fatti, utilizzo meno risorse ed ho un ecosistema (librerie e strumenti) già pronti.

**Metodi agili**, opposizione al modello sequenziale. 4 principi su cui si basa:

1. Mettere in primo piano persone e iterazioni piuttosto che processi e strumenti. Gli individui sono importanti ed è importante il modo in cui interagiscono tra loro;
2. *"Dei documenti me ne frego, basta che funzioni"*. La documentazione non sempre corrisponde a sw funzionante. E' importante che funzioni. Ma nel lungo periodo farà soffrire chi dovrà prendere in mano il sw;
3. Avere un buon rapporto con il customer, coinvolgerlo, non ingessare il rapporto;
4. Reattività piuttosto che pianificazione. Capacità di adattamento a cambiamenti delle situazioni.

Il modello agile è maturato in qualcosa di plausibile, in alcune tecniche che funzionano.

**Modello agile educato**, tutto ruota intorno al *"document user story"*, è l'espressione precisa di quello che l'utente vuole in quel momento. Associato ad ogni user story produrre una strategia che sia in grado di soddisfare quei bisogni. User story associato a un insieme di cose da fare che dimostrino al customer che c'è una soluzione (**product backlog**). Quello che si ottiene va verso il customer o in feedback. C'è

un supervisore che decide l'urgenza strategica (**sprint**). Tutti fanno un pezzo e poi lo consegnano. Ogni iterazione è rapida e va molto verso le aspettative del cliente.

**Scrum daily**, mucchio giornaliero, un contatto giornaliero dell'avanzamento. Tecnica ragionevole per chi ha poca esperienza.