

2. Modelli del ciclo di vita del software

Paola Barra
a.a. 2022/2023

Il processo software

Con processo software si indica il percorso da svolgere per sviluppare un prodotto o sistema software:

- inizia con l'esplorazione dell'idea e finisce con la dismissione del software
- durante la sua vita il prodotto attraversa varie fasi
- il processo software include anche gli strumenti e le tecniche per svilupparlo e i professionisti del software coinvolti

Modellare significa suddividere in attività che si occupino di :

- cosa
- quali prodotti
- quando

I processi software

Un processo software è un insieme di attività che porta alla creazione di un prodotto software.

Tutti i processi software hanno in comune quattro importanti attività:

1. **Specifica del software:** vengono definite le funzionalità del software fissati i vincoli operativi
2. **Sviluppo del software:** deve essere prodotto il software che realizza i requisiti definiti nella specifica
3. **Convalida del software:** il software deve essere convalidato per garantire ciò che il cliente richiede
4. **Evoluzione del software:** il software deve evolversi per soddisfare le necessità e i cambiamenti dei requisiti dell'utente

Oltre i processi

- **prodotti e consegne** sono il risultato di un'attività di processo
- bisogna definire i **ruoli** che riflettono le responsabilità delle persone all'interno del progetto
- **precondizioni e postcondizioni** da rispettare per portare a termine il progetto con successo (ad esempio approvazione dei requisiti)

Modello di ciclo di vita:

Organizzazione delle attività:

- ordinamento delle attività
- criteri per terminare e passare alla successiva

Esempio: modello di sviluppo generico per la preparazione di un dolce

- fare la spesa, mentre il forno si scalda: impastare, mettere nella teglia e infornare

Sono i passi da seguire per la preparazione del dolce, non è la ricetta.

I modelli di ciclo di vita

Ogni modello che presenteremo è importante perchè affronta un nuovo e differente aspetto:

- gli aspetti affrontati rappresentano le caratteristiche peculiari che ci interessano e guidano lo sviluppatore
- un ciclo di vita di un progetto reale può mescolare idee e modelli.

Problemi connessi allo sviluppo del software

I requisiti cambiano costantemente

- Il cliente può non essere consapevole di tutti i requisiti in anticipo

Le continue modifiche sono difficili da gestire

- Identificare checkpoint per la pianificazione e la stima dei costi è difficile

C'è più di un sistema software

- Il nuovo sistema spesso deve essere compatibile con sistemi già esistenti (legacy system)

Ciclo di vita del software

Ciclo di vita del software (anche processo software)

- Insieme di attività e loro relazioni reciproche per supportare lo sviluppo di un sistema software

Modello di ciclo di vita (anche modello del processo software)

- Un'astrazione che rappresenta un ciclo di vita del software con lo scopo di capire, monitorare, o controllare lo sviluppo di un sistema software

Ciclo di vita del sw: lo standard IEEE 610.12-1990

Processo di sviluppo del software:

- il processo per cui le necessità dell'utente sono tradotte in un prodotto software

Il processo prevede la traduzione

- delle necessità **dell'utente** nelle
- **richieste del software**, trasformare le richieste del software nel
- **progetto**, implementare il progetto in forma di
- **codice, testing del codice** e talvolta,
- **installare e controllare** il software per un uso operativo
- Nota: queste attività possono sovrapporsi o essere eseguite iterativamente

Ciclo di vita del software

- In pratica ...



Ciclo di vita del software: introduzione

Non esiste un ciclo di vita del software ideale

- Esistono molteplici processi software differenti
 - Le grandi organizzazioni hanno sviluppato propri approcci in base alle proprie necessità
- I processi si sono evoluti anche in base alle specifiche caratteristiche dei sistemi da sviluppare
 - Ad esempio, per i sistemi critici (sistemi per cui un fallimento del sw può portare a gravi perdite economiche, danni fisici o minacce per la vita) un processo di sviluppo strutturato è più adeguato
 - Per i sistemi aziendali in cui i requisiti cambiano velocemente, potrebbe essere più efficace un processo agile e flessibile

Identificazione delle attività dello sviluppo software

Domande nello sviluppo del software

- Qual è il problema?
- Come possiamo dividere il problema?
- Qual è la soluzione?
- Qual è il miglior meccanismo per implementare la soluzione?
- Come è costruita la soluzione?
- Il problema è risolto?
- Il cliente è in grado di usare la soluzione?
- Sono necessarie estensioni?

Attività di sviluppo software - Esempio 1

Analisi requisiti

Qual è il problema?

**Dominio dell'
Applicazione**

Progettazione sistema

Qual è la soluzione?

Progettazione dettagliata

**Quali sono i migliori meccanismi
per implementare la soluzione?**

Implementazione

Come è costruita la soluzione?

**Dominio della
Soluzione**

Testing

Problema risolto?

Consegna

Il cliente in grado di usare la soluzione?

Manutenzione

Sono necessarie estensioni?

Attività di sviluppo software - Esempio 2

Analisi requisiti

Qual è il problema?

**Dominio dell'
Applicazione**

Progettazione sistema

Qual è la soluzione?

Progettazione oggetti

**Quali sono i migliori meccanismi
Per implementare la soluzione?**

Implementazione

**Come è costruita
la soluzione?**

**Dominio della
Soluzione**

Attività del processo software

Le attività del processo software fondamentali, comuni a tutti i processi sono:

- **Specifiche del software:** si definiscono le funzionalità del **sw** e sono fissati i vincoli operativi
- **Progettazione e implementazione:** si sviluppa il **sw** in modo da realizzare i requisiti individuati
- **Convalida:** si verifica che il **sw** garantisca le funzionalità richieste dal cliente
- **Evoluzione:** il **sw** deve poter essere modificato in modo da soddisfare i cambiamenti dei requisiti dell'utente

Ciclo di vita del software: una possibile soluzione

Analisi dei
requisiti

Progettazione

Implementazione

Testing

Manutenzione



Modelli di ciclo di vita del software o di processo di sviluppo del software

Un modello di processo è un'astrazione del processo software

- Rappresenta il processo software da una particolare prospettiva

“Un modello del ciclo di vita del software è una caratterizzazione descrittiva o prescrittiva di come un sistema software è o dovrebbe essere sviluppato”

W. Scacchi - Encyclopedia of Software Engineering Vol II pag 860

I modelli di processo software sono descrizioni precise e formalizzate delle attività, degli oggetti, delle trasformazioni e degli eventi per realizzare e/o ottenere l'evoluzione del software

N.B.: molti autori usano i termini *processo di sviluppo del software* e *ciclo di vita del software* come sinonimi

Modelli di processo: caratteristiche (1)

Una strutturazione dell'organizzazione del lavoro nelle fabbriche del software

- ... fasi della produzione,
- ... tipi di attività,
- ... collegamento ed interfacciamento,
- ... controllo e misura,

ma anche linee guida per: organizzare, pianificare, dimensionare personale, assegnare budget, schedulare e gestire, ...

Modelli di processo: caratteristiche (2)

- definire e prescrivere prodotti e documenti da rilasciare al committente (**deliverables**)
- determinare e classificare metodi e strumenti più adatti a supportare le attività previste
- framework per analizzare, stimare, migliorare ...

Il concetto di ciclo di vita

Concezione -> sviluppo -> utilizzo -> ritiro

(a noi ci interessa solo il segmento concezione -> sviluppo)

La transizione tra stati avviene per azione di processi di ciclo di vita.

L'obiettivo di un progetto è far progredire lo stato di avanzamento di un prodotto software

Per farlo, il progetto mobilita specifiche attività di specifici processi:

- le ordiniamo secondo le dipendenze tra i loro ingressi e uscite
- fissando i criteri di attivazione e completamento
(pre-condizioni e post-condizioni)

Il concetto di ciclo di vita

- Il termine “fase” corrisponde allo stazionamento in uno stato di ciclo di vita o in una transizione tra stati
- Esistono molteplici cicli di vita, che differiscono tra loro per transizioni tra stati e regole di attivazione (ciascuno viene idealizzato da un modello)
- Aderire a un particolare modello comporta vincoli sulla pianificazione e gestione del corrispondente progetto

(questo influenza la selezione del way of working e dei suoi strumenti di supporto)

Esempio

Questo è un modello di ciclo di vita che non contempla il “ritiro” (fine vita)

Lo stadio di “evoluzione” (manutenzione) innesca nuovi cicli di sviluppo



Processi guidati da piani VS processi agili

- I processi guidati da un piano sono processi in cui **tutte** le attività del processo sono pianificate in anticipo ed il progresso è misurato rispetto al piano
- Nei processi agili, la pianificazione è incrementale ed è più facile cambiare il processo per riflettere la modifica dei requisiti del cliente
- In pratica, molti processi concreti includono elementi di ambo gli approcci
 - Non ci sono processi software corretti o sbagliati

I modelli

Build and fix model

Modelli prescrittivi :

- modello a cascata
- rapid prototyping
- modello incrementale

Modelli agili :

- Extreme programming
 - SCRUM
-

Build and fix model

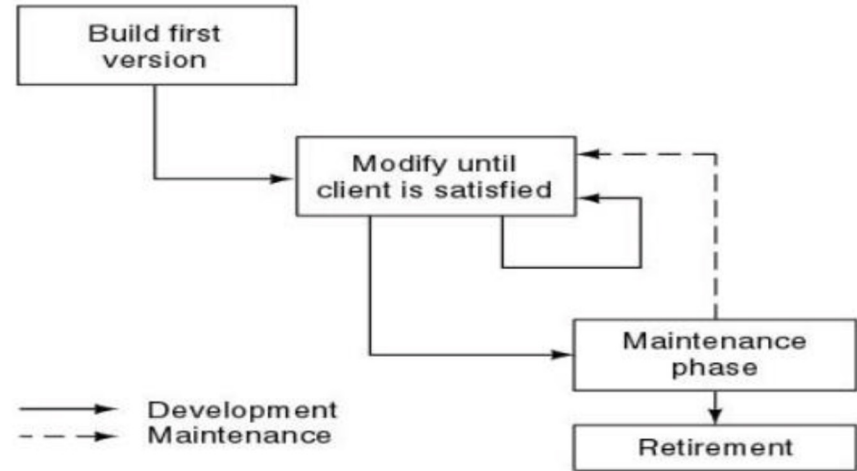
Build and fix model (o anche code and fix)

- è chiamato anche cowboy coding
- Attività senza organizzazione preordinata
- fonte di progetti caotici difficilmente gestibili



Build and fix model

- Le attività non sono identificate
- il prodotto è sviluppato senza specifica e senza un tentativo di progettazione
- lo sviluppatore scrive un programma
- che poi è modificato più volte finchè non soddisfa il committente

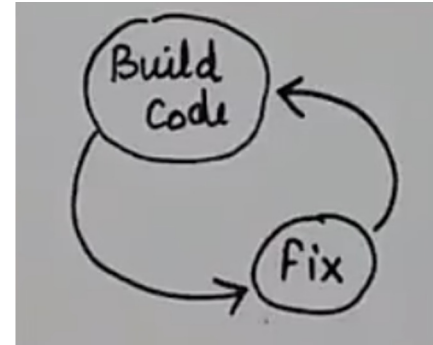


Il più semplice (e peggiore) modello di processo

- Molto veloce, feedback rapido
- Molti strumenti disponibili
- Specializzato per codifica
- Non incoraggia la documentazione
- Ingestibile durante manutenzione

Build and fix model

- forse adeguato a un progetto di 100 linee di codice
- Diventa improponibile per prodotti di ragionevole grandezza
- la manutenzione di un prodotto senza specifica o documentazione che ne spieghi la progettazione è estremamente difficile



Prima di iniziare lo sviluppo di un progetto dobbiamo scegliere un “vero” modello del ciclo di vita

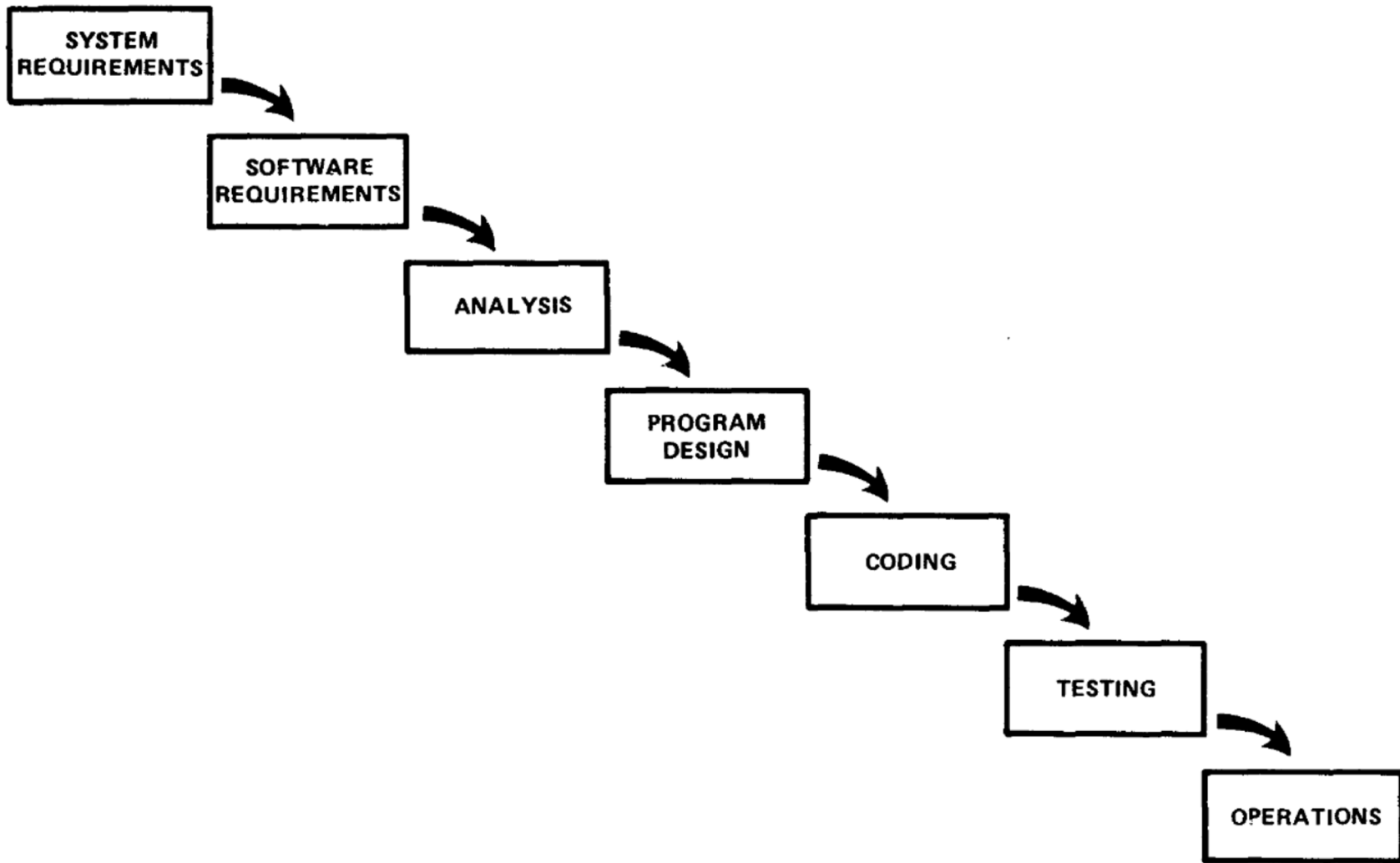
Build and fix model

- Quello stile causò la crisi del software, che portò alla nascita della disciplina software engineering
- Ne nacque una successione di modelli organizzati

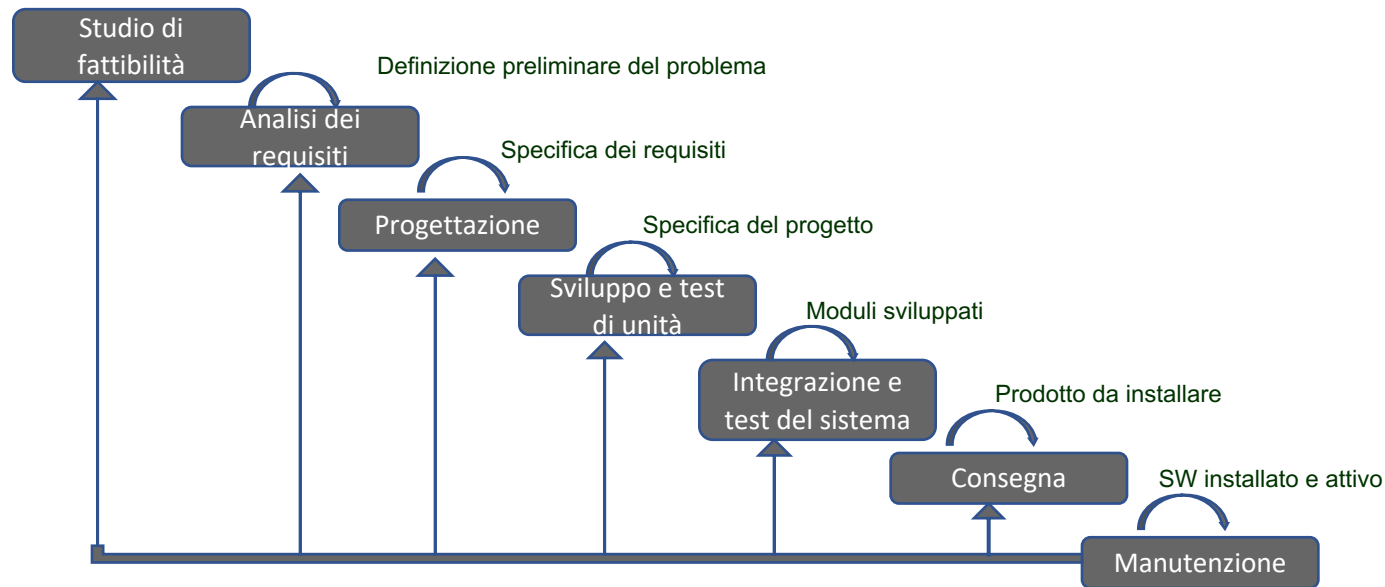
Modello	Tratti caratteristici
Cascata	Rigide fasi sequenziali
Incrementale	realizzazione in più passi
A componenti	Orientato al riuso
Agile	Altamente dinamico, fatto di brevi cicli iterativi e incrementali

Il modello a cascata

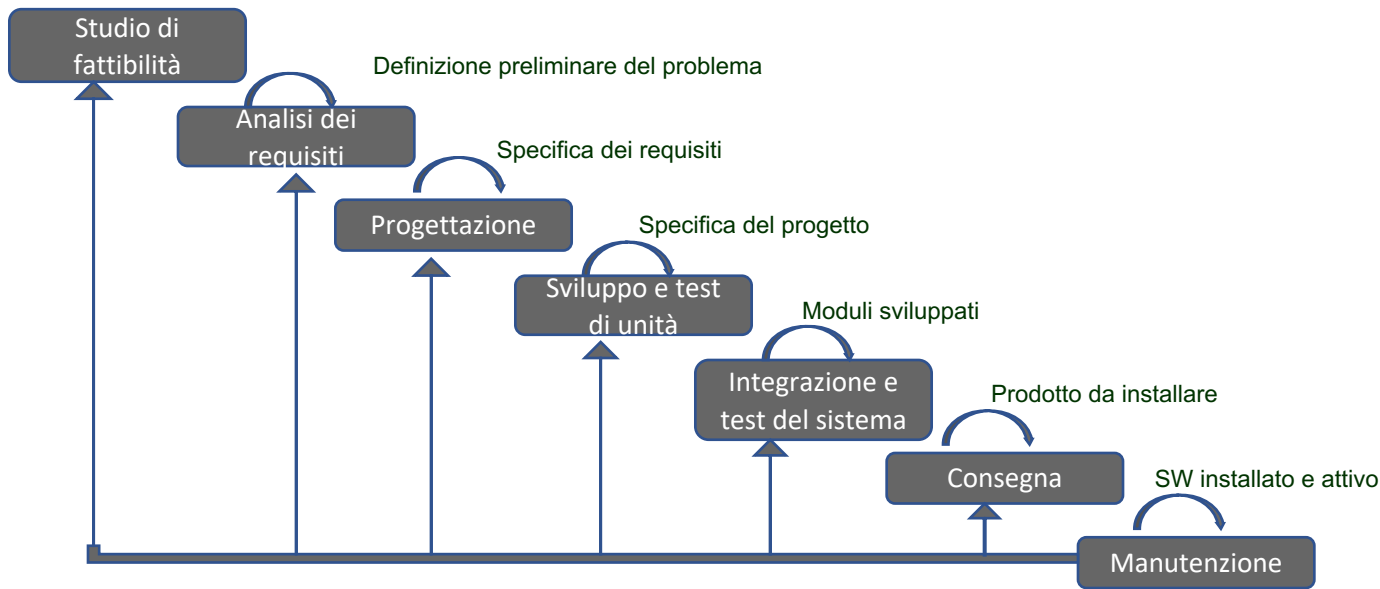
[Royce, 1970]



Le fasi di un modello a cascata

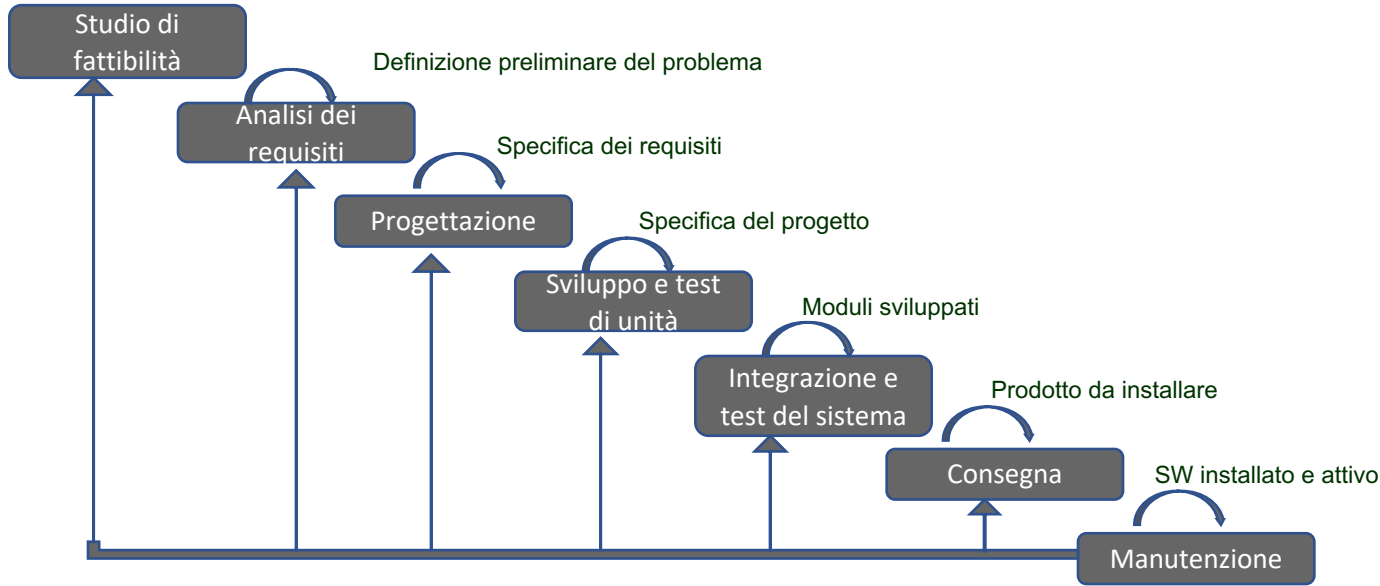


Analisi e definizione dei requisiti



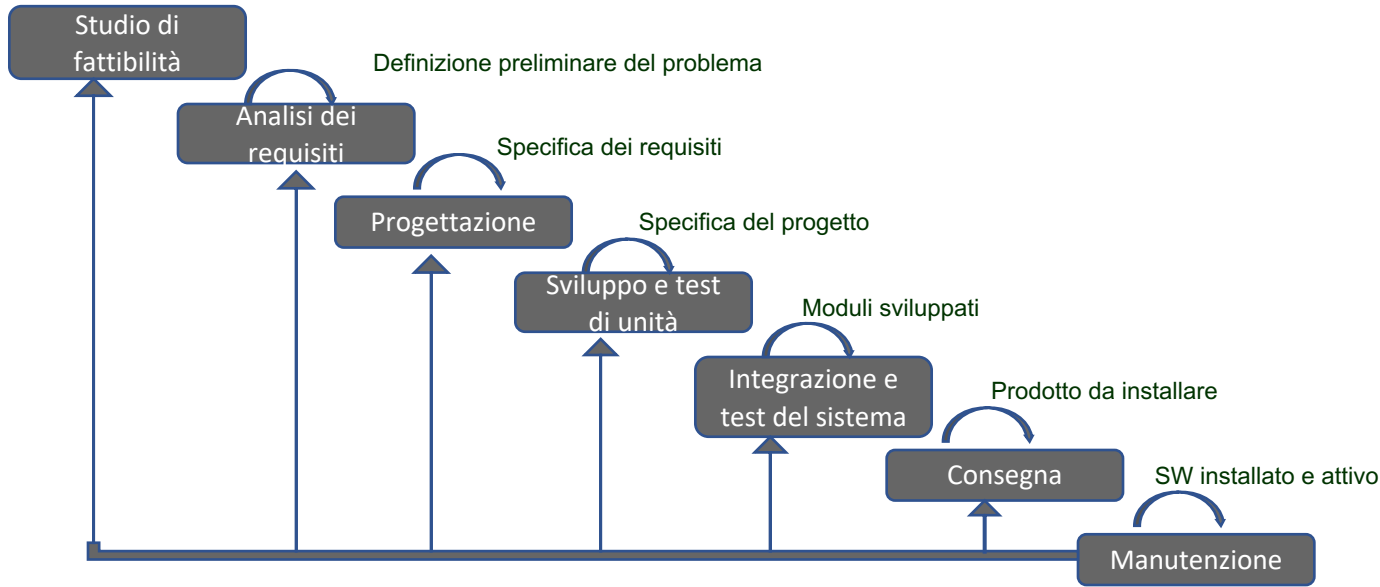
Si determinano i servizi del sistema, i suoi vincoli e obiettivi attraverso incontri con gli utenti del sistema. I requisiti vengono definiti in dettaglio e servono come specifica del sistema.

Progettazione del sistema e del software



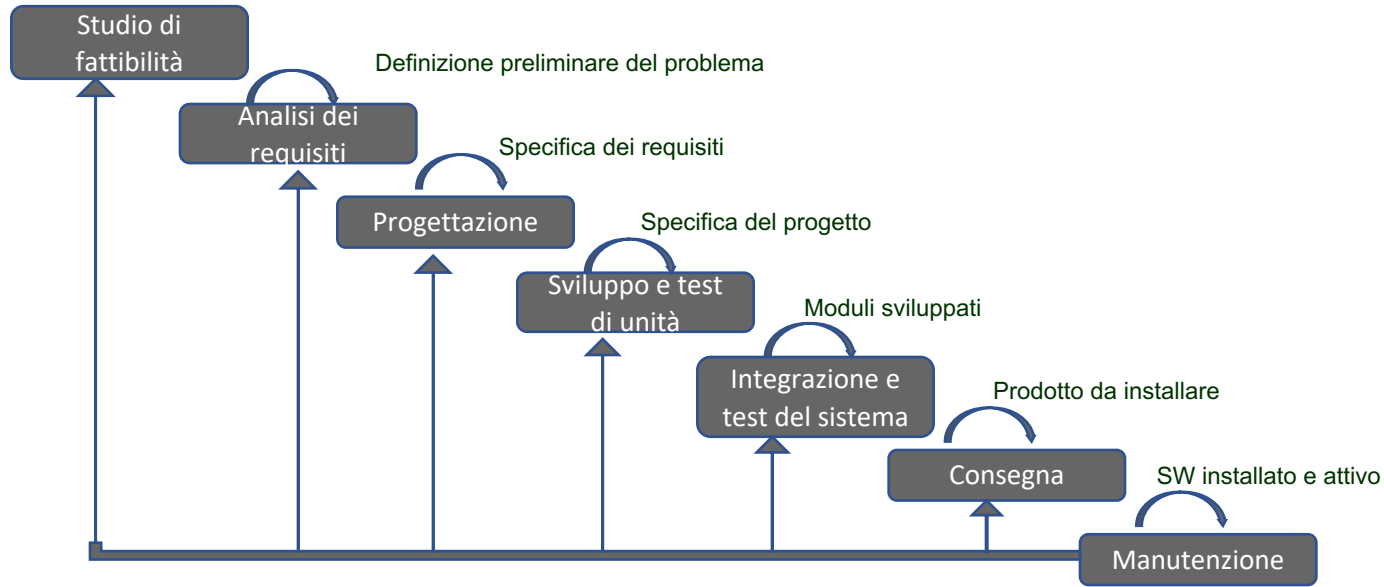
Il processo di progettazione del sistema suddivide i requisiti sul sistema hardware o sul software e stabilisce l'architettura generale del sistema. La progettazione coinvolge l'identificazione e la descrizione delle astrazioni fondamentali del sistema e le reciproche relazioni.

Sviluppo e test di unità



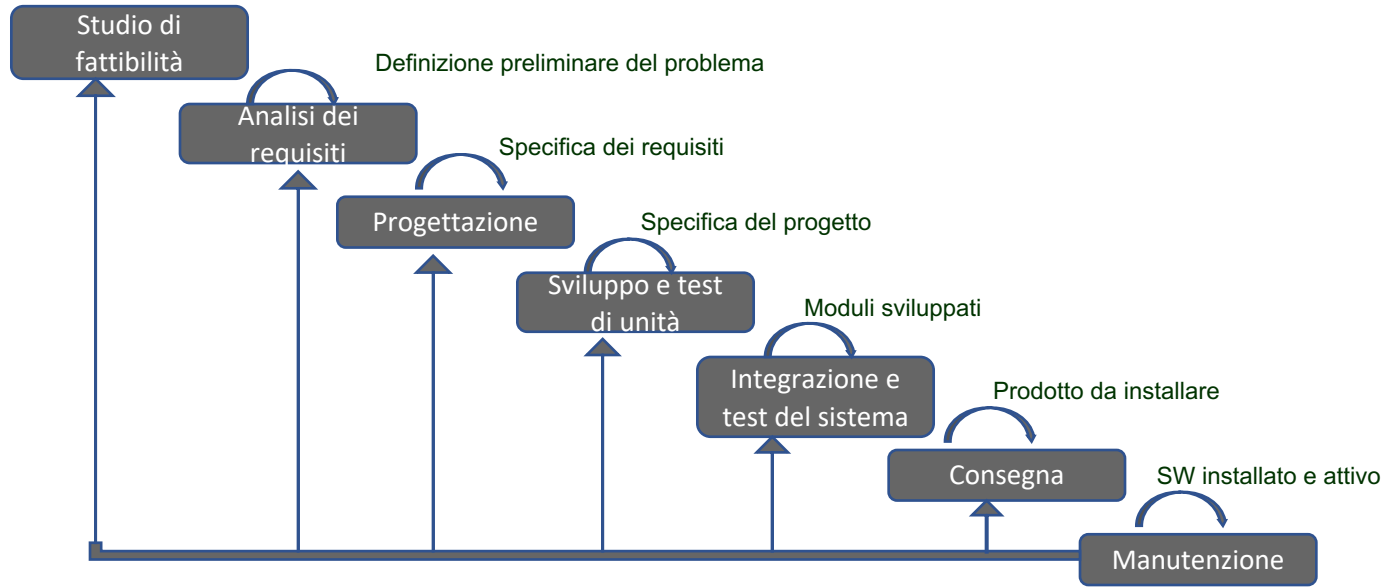
In questa fase il progetto del software è realizzato come un insieme di programmi o di unità del programma. Il test delle unità verifica che ognuna soddisfi le proprie specifiche.

Integrazione e test del sistema



Le singole unità di programma o i programmi sono integrati e testati come un sistema completo per accertare che i requisiti del software siano soddisfatti. Dopo il test finale, il sistema è consegnato al cliente.

Consegna e manutenzione



normalmente (anche se non è obbligatoria), questa è la fase più lunga dell'intero ciclo di vita. Il sistema viene installato e messo in opera, si correggono gli errori non scoperti nei primi stadi del ciclo di vita, si migliora l'implementazione delle unità di sistema quando vengono evidenziati nuovi requisiti.

Modello a cascata

- Popolare negli anni '70
 - reazione al “code and fix” originario
 - ispirazione dall'industria manifatturiera
- Modello sequenziale lineare
 - progressione sequenziale (in cascata) di fasi, senza cicli, al fine di meglio controllare tempi e costi
 - definisce e separa le varie fasi e attività del processo
 - nullo (o minimo) overlap fra le fasi
 - uscite intermedie: semilavorati del processo (documentazione di tipo cartaceo, programmi)
 - formalizzati in struttura e contenuti
 - consente un controllo dell'evoluzione del processo
 - attività trasversali alle diverse fasi

Il modello a cascata

Questo modello permette di distinguere e definire le fasi di un processo software evidenziando l'importanza delle fasi di analisi e progettazione prima di passare alla codifica

Il modello richiede che il passaggio a una nuova fase sia possibile solo dopo il completamento della precedente:

- ogni fase produce un documento che deve essere approvato da un gruppo di valutatori prima di passare alla fase successiva
- si parla anche di documento “document driven”

Critiche al modello a cascata...



Analisi del modello a cascata

Il vero punto debole è che manca l'interazione col cliente che vede solo il prodotto finito, alla fine del processo:

- spesso c'è una sostanziale differenza tra come il cliente immagina un prodotto e come il prodotto viene realizzato
- Alla fine del processo se ci sono discrepanze tutto il processo deve essere ripetuto

Un ulteriore punto critico è l'eccessiva produzione di documenti:

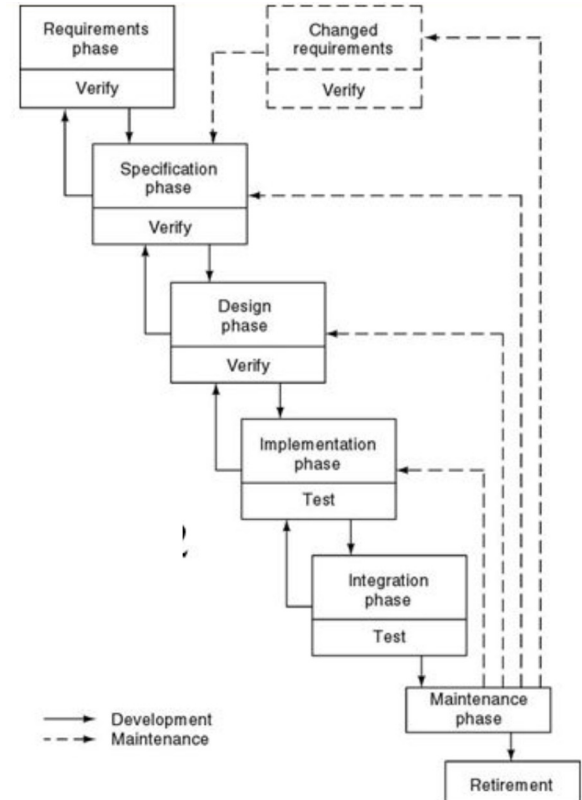
- nessuna fase è completa finché anche il documento per quella fase non è stato terminato e approvato dal gruppo di valutatori (Software Quality Assurance)

Il modello a cascata ha avuto storicamente il valore di aver definito e strutturato le fasi di un processo software.

Royce

Il modello a cascata è a tutti noto come quello visto, rigidamente “in caduta” (tutta la letteratura riporta quel modello)

Royce in realtà già nell’articolo del 1970, evidenzia le limitazioni della cascata e propone altri modelli in alternativa.



Proprietà dei modelli basati su modelli a cascata

- I responsabili amano il modello a cascata
 - Milestone diretti
 - Nessuna necessità di cicli all'indietro (Sistema lineare)
 - Sempre un'attività alla volta
 - Facile controllare il progresso durante lo sviluppo
- Tuttavia, lo sviluppo del software non è lineare
 - Mentre si sta sviluppando una progettazione, sono identificati problemi nei requisiti
 - Mentre si scrive codice, sono trovati problemi di progettazione e di requisiti
 - Mentre è testato un programma, emergono errori di codifica, di progettazione e nei requisiti

Modello a cascata: vantaggi e svantaggi

Vantaggi

- ha definito molti concetti utili
- ha rappresentato un punto di partenza importante per lo studio dei processi SW
- facilmente comprensibile e applicabile
- Ogni fase del processo produce la relativa documentazione

Svantaggi

- interazione con il committente solo all'inizio e alla fine
- requisiti congelati alla fine della fase di analisi
- requisiti utente spesso imprecisi: "l'utente sa quello che vuole solo quando lo vede"
- Errori nei requisiti scoperti solo alla fine del processo
- il nuovo sistema software diventa installabile solo quando è totalmente finito
- né l'utente né il management possono giudicare prima della fine la distanza del sistema dalle proprie aspettative

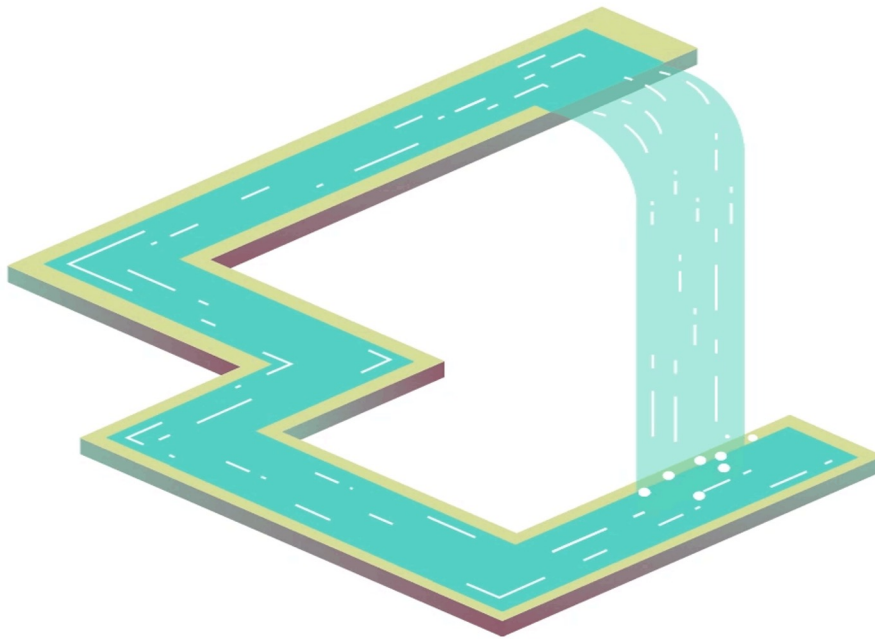
Modello a cascata, in pratica...

Il modello è adeguato solo quando i requisiti sono ben compresi e le modifiche sono piuttosto limitate durante la progettazione

E' maggiormente usato in grossi progetti in cui un sistema è sviluppato in diversi siti

- In tali circostanze, la natura guidata da un piano aiuta a coordinare il lavoro

Iterazione del modello a cascata



Critica del modello a cascata (sequenziale)

Difetto principale: eccessiva rigidità

- stretta sequenzialità: nessun parallelismo nessun ritorno
- non ammette modifiche nei requisiti in corso d'opera
- visione rigida e poco realistica del progetto

Correttivo 1. con prototipazione

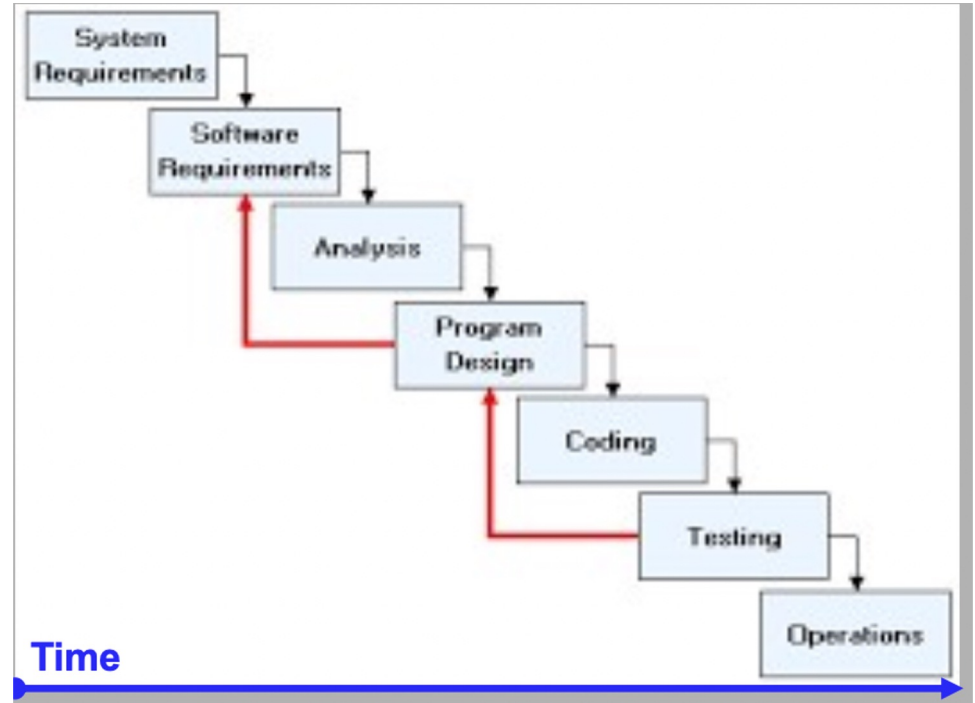
prototipi “usa e getta” :per capire meglio i requisiti o le soluzioni strettamente all'interno delle singole fasi

Correttivo 2. con ritorni

come “allenamenti” prima dell'atto definitivo

Rischi dei modelli iterativi

Ogni iterazione comporta un ritorno all'indietro nella direzione opposta all'avanzamento del tempo.



Rapid prototyping

Rapid prototyping

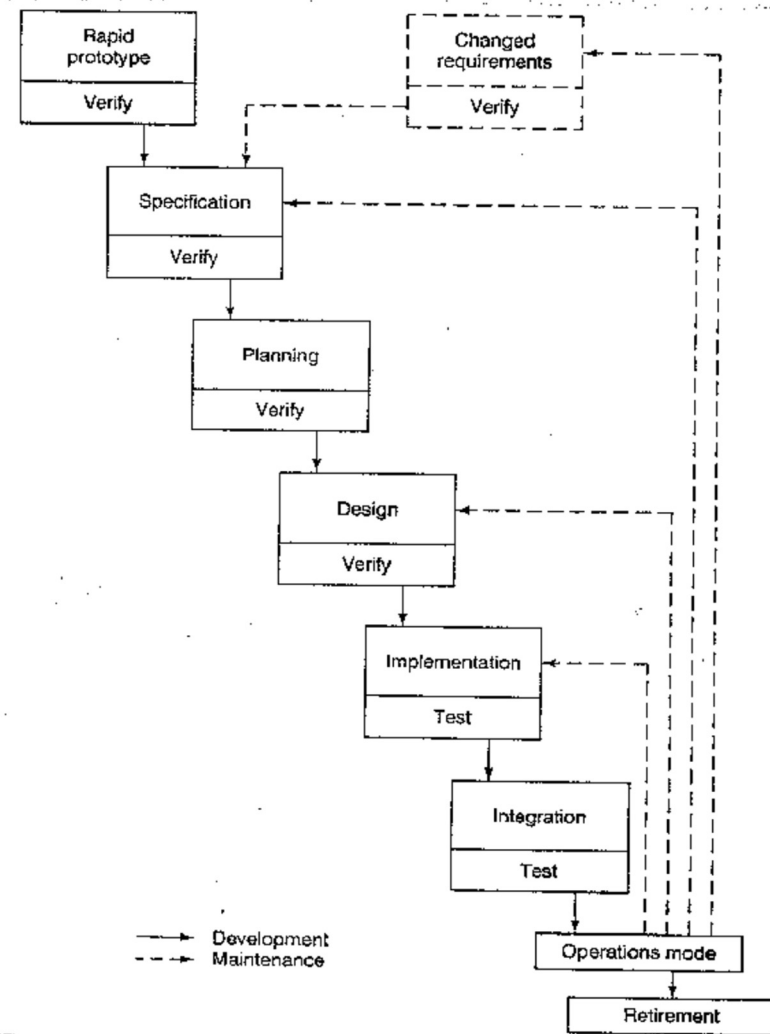
Il primo passo è quello di costruire un prototipo velocemente per permettere al committente di sperimentarlo.

Utile quando i requisiti non sono chiari

Il prototipo aiuta il cliente a meglio descrivere i requisiti:

- si passa a questo punto alla fase di specifica

Anche noto come modello evolutivo



Rapid prototyping

