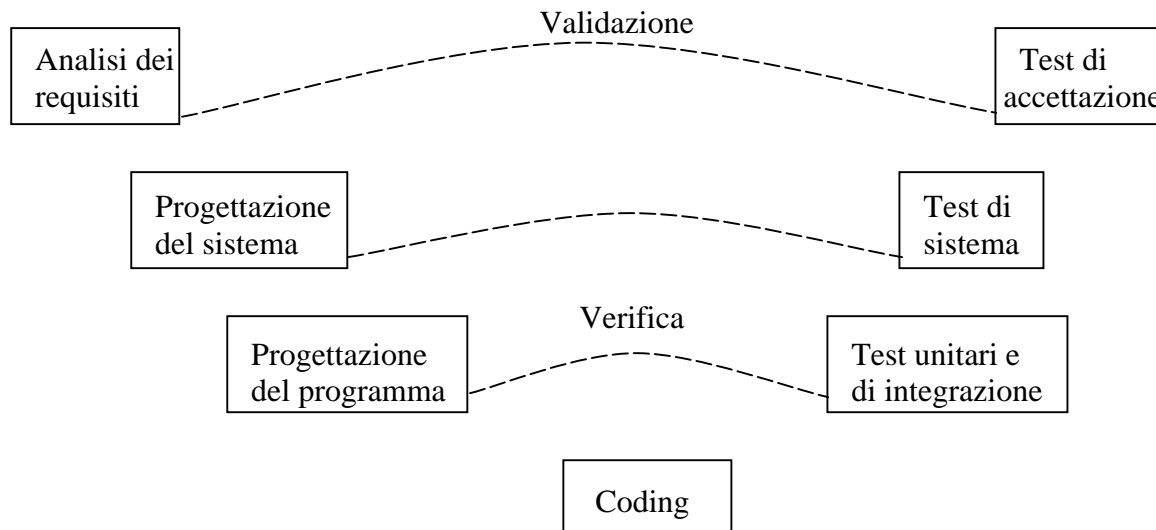


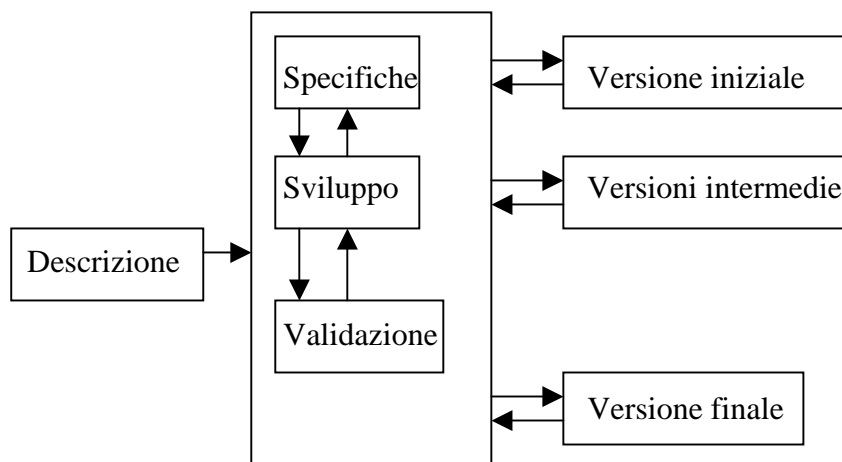
Il modello a V

Una variante del modello Waterfall è il cosiddetto modello a V:



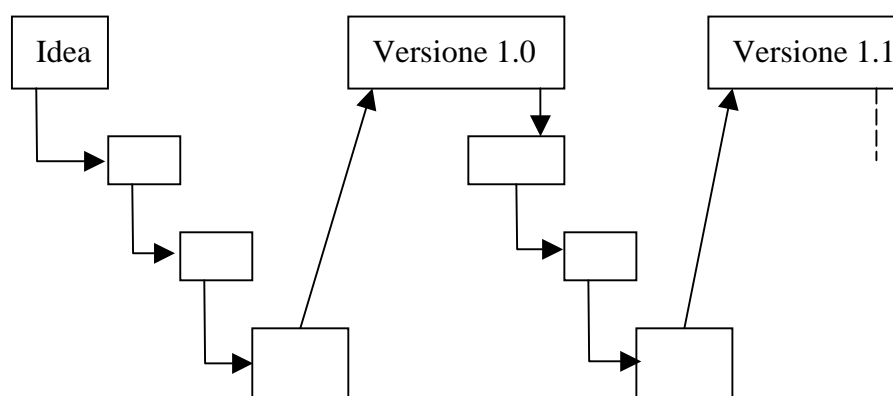
Il modello a V è un modello sequenziale e separa il processo software in due cascate, e prevede, durante lo sviluppo della cascata sinistra, anche uno sviluppo limitatamente parallelo della cascata destra. Ogni fase che compone la cascata di sinistra verrà testata in una fase della cascata destra, in pratica ogni fase sinistra è preparatoria alla sua fase speculare destra. Ciascuna fase non produce molti documenti, inoltre V permette di lavorare limitatamente in parallelo. La differenza con il Waterfall è rappresentata dal fatto che quest'ultimo prevede, alla fine di ogni fase, la produzione di qualcosa, mentre il modello a V prevede di produrre da una parte delle attività di controllo, mentre dall'altra parte si verificano gli approcci scelti.

I modelli evolutivi



Una famiglia importante di modelli di processo è quella dei modelli evolutivi. In tali modelli si dispone di una descrizione, successivamente si formalizzano le specifiche, le quali vengono integrate passo passo con lo sviluppo e la validazione, permettendo di ripercorrere alcune fasi quando si rende necessario ("andata e ritorno"). Attraverso tali modelli è possibile intraprendere un

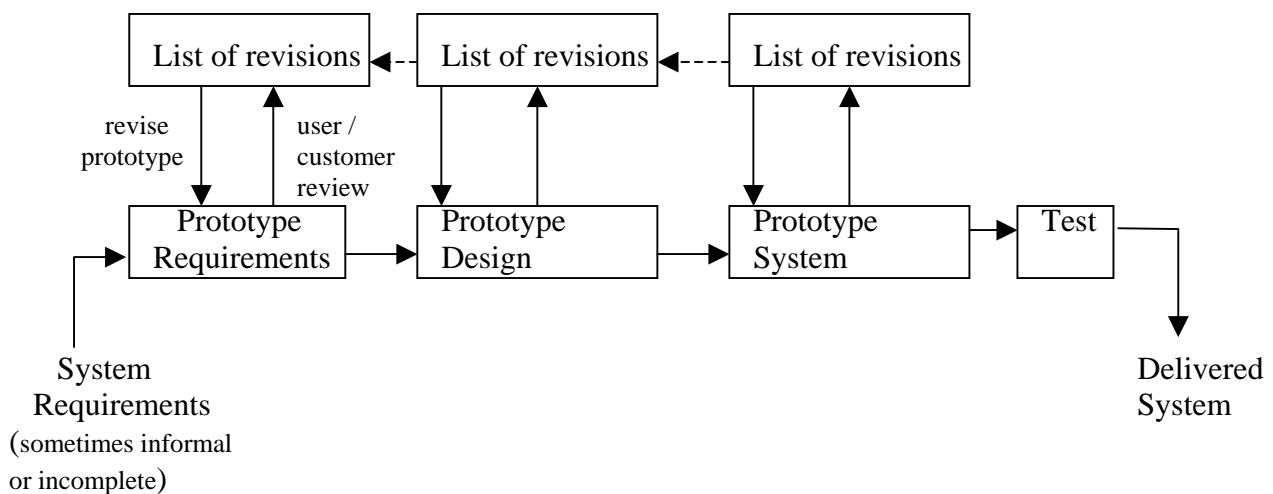
progetto non chiaro, lavorando con ciò che si conosce, e proseguendo sino ad arrivare ad una versione iniziale, sviluppando in seguito delle versioni intermedie, e infine elaborando una versione finale, dunque si procede per passi. I vantaggi dei modelli evolutivi sono parecchi, difatti non bisogna aspettare il completamento del processo per poter vedere qualcosa di pronto, inoltre si utilizza da subito il personale. Uno svantaggio è invece rappresentato dalla poca visibilità che godono i processi che intraprendono questo genere di modelli: come sappiamo la visibilità di un processo sono i suoi progressi, ma un processo che segue un modello evolutivo sviluppa molta documentazione la quale risulterà difficile da gestire ed aggiornare. Per tale motivo implementare accade spesso di non accorgersi dei progressi che si fanno e di dimenticare di implementare alcune funzionalità durante lo sviluppo. In questo caso bisognerà riprendere lo sviluppo dei moduli, modificandoli e introducendo nuove caratteristiche. Infine, se si introducono molte caratteristiche non pianificate, a lungo andare si avrà la tendenza a rovinare la struttura del programma, la quale se all'inizio risultava semplice, procedendo in questi termini, diventerà man mano sempre più prolissa. Per tale motivo si dovrebbe sempre prevedere, ove possibile, eventuali modifiche sulla struttura del programma. Esistono due principali approcci di tali modelli: l'approccio che definisce il modello incrementale, e quello che definisce il modello prototyping.



Nel modello incrementale si procede sviluppando diverse versioni, considerando un determinato gruppo di specifiche che risultano chiare, ed integrando successivamente altre specifiche. Si sviluppa con quello che si ha a disposizione, in modo da soddisfare, in poco tempo, alcune richieste del cliente: si potrebbe ad esempio sviluppare il cuore dell'applicazione, in modo da fornire una prima versione al cliente, in seguito si aggiungono ulteriori caratteristiche, ottenendo in tal modo altre versioni. Si parte dunque da un progetto piccolo e successivamente lo si incrementa. I requisiti in questo caso potrebbero aumentare procedendo nello sviluppo. Si utilizza il modello incrementale anche per riprendere lo sviluppo di software non più sviluppati. Il modello incrementale offre il vantaggio di poter sviluppare qualcosa di sicuro, basandosi su una parte di requisiti chiari.

Il modello prototyping si utilizza nel momento in cui non sono conosciuti tutti i requisiti e il cliente non ha del tutto chiare le proprie idee, in questo caso il modello prototyping prevede lo sviluppo di un prototipo, in modo da poter comprendere se ciò che si sta realizzando sia quello che vuole il cliente. Il modello prototyping utilizza per lo sviluppo di prototipi solo i requisiti rappresentativi. Lo sviluppo di prototipi viene effettuato rapidamente e ad un costo bassissimo, in modo da non perdere tempo e denaro, e tutto questo permette di chiarire passo per passo i requisiti. E' possibile sviluppare diversi prototipi: uno per l'interfaccia grafica, uno che comprenda i requisiti, ecc...

Il modello prototyping è applicato anche per testare nuove tecnologie. E' possibile intraprendere, attraverso questa tecnica, l'uso di modelli ibridi: se ad una certa fase dello sviluppo ci si accorge di dover cambiare qualcosa, attraverso l'uso di tale modello, si può effettuare un cambiamento di modello seguendone un altro, questo fa del modello prototyping il candidato ideale per delle fasi di test. I vantaggi del modello prototyping sono molteplici: si sviluppa una soluzione in poco tempo a costo minore, successivamente tal prototipo potrebbe diventare il prodotto stesso, risparmiando notevolmente. Tuttavia quest'ultimo vantaggio potrebbe rappresentare in alcuni casi anche uno svantaggio, difatti l'intrinseca natura di un prototipo ne sottintende una minore qualità. Infine, mentre nel modello incrementale è possibile aggiungere molti moduli al progetto iniziale, se si segue questa strada anche per un prototipo si potrebbe arrivare alla distruzione della struttura del progetto. Anche la visibilità del processo è scarsa, in quanto come il prototipo anche la documentazione sarà curata, in quanto la si pensa come soluzione finale.



Questa è una generalizzazione del prototyping, infatti esso potrebbe integrare altri modelli, oppure si possono fare diversi prototipi, uno per l'interfaccia, uno sui requisiti, e permetterà un prototipo per ogni fase. In tal modo c'è una scelta maggiore. Non per forza si prototipizzano tutte le fasi. Con la multimedialità è nota l'esigenza di affidare la progettazione delle interfacce grafiche a dei gruppi che studiano l'interazione del cliente con l'interfaccia.

Modello a spirale di Boehm

Un'altra famiglia di modelli è rappresentata dai modelli a spirale di Boehm. Il modello di Boehm si basa sulla valutazione dei rischi che si potrebbero presentarsi durante l'esecuzione di un processo software: si considerano rischi ad esempio la dubbia competenza del personale, il poco denaro a disposizione, i tempi ristretti, ecc... Si procede valutando i possibili rischi, e dopo attenta valutazione si sviluppano una serie di prototipi che potrebbero tutelare il processo da tali rischi. In particolare si prevede di lavorare seguendo un ciclo "a spirale" diviso in quattro parti:

- 1) nella prima parte si determina ciò a cui si deve giungere con lo sviluppo, attraverso il budget a disposizione.
- 2) nella seconda parte si analizzano i rischi da dover affrontare.
- 3) la terza fase è rappresentata dallo sviluppo.
- 4) la quarta fase è rappresentata dal collaudo.

Questa rappresenta l'idea di base, ma potrebbero bastare meno cicli, questo dipende dal tipo di progetto. Il modello a spirale di Boehm prevede l'idea di lavorare seguendo un ciclo, diviso in 4 parti. La prima parte determina ciò a cui si deve arrivare con il budget che abbiamo. La 2^a parte analizza i rischi da affrontare e si procede costruendo dei prototipi che rappresentano prototipi per la verifica in modo da rispondere a questi rischi. Si ha infine la parte di sviluppo e di collaudo in cui si fa la verifica sulle cose che si devono fare. Questa è l'idea di base e potrebbero bastare meno cicli a seconda del progetto. Si sceglie questa soluzione quando i rischi da affrontare sono alti.

