

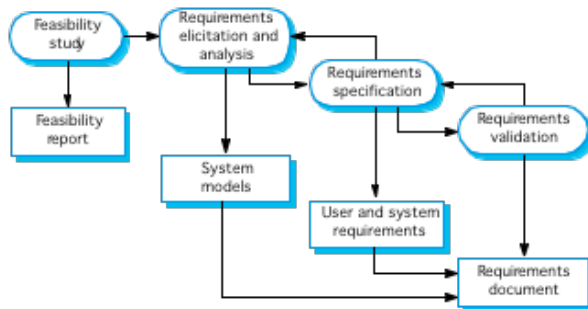
Processi (di sviluppo del) software

- Un *processo software* descrive le *attività* (o task) necessarie allo sviluppo di un prodotto software e come queste attività sono collegate tra loro
- Attività di sviluppo
 - Analisi requisiti (specifiche)
 - Progettazione (design)
 - Codifica o implementazione (codice)
 - Validazione
 - Manutenzione

Riferimenti
Pressman, capitoli 3.1, 3.2, 3.3.1, 3.4.1, 3.4.2, 3.5.1, 3.6, 4.1, 4.2, 4.3.1, 5.1, 5.4, 5.5, 8.7.4
Sommerville, capitolo 3 #3.7 +RUP

Web
www.agilealliance.com
www.extremeprogramming.org

Ingegneria dei requisiti



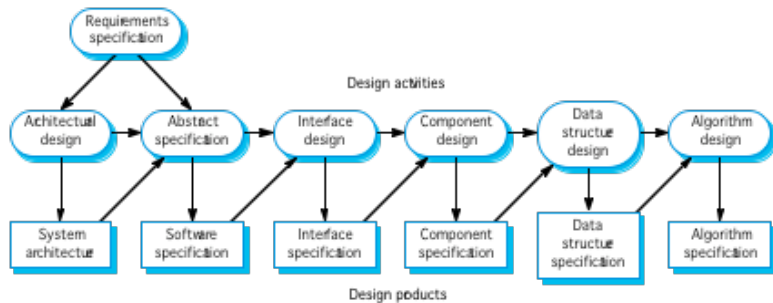
Analisi dei Requisiti

- L'analisi dei requisiti è il processo che porta a definire le specifiche, stabilisce i servizi richiesti ed i vincoli del software
 - Requisito = ciascuna delle caratteristiche del software
 - Specifica = descrizione rigorosa del software
- Fasi per l'*ingegneria dei requisiti* (vedi lezioni 4 e 6)
 - Studio di fattibilità
 - Analisi requisiti
 - Specifica requisiti
 - Validazione requisiti
- Requisiti funzionali
 - **Cosa** il sistema deve fare (funzionalità)
- Requisiti non funzionali
 - **Come** il sistema lo fa (es. affidabilità, efficienza, prestazioni, manutenibilità. etc.)

Progettazione ed implementazione

- Progettazione (vedi lezioni da 7 a 13)
 - Il processo che stabilisce la struttura software che realizza le specifiche
 - Attività
 - Progettazione architettura, Specifiche delle respons. dei sottosistemi
 - Progettazione interfacce, Progettazione componenti
 - Progettazione strutture dati, Progettazione algoritmi
- Implementazione
 - Produce un programma eseguibile a partire dalla struttura
- Progettazione ed implementazione sono attività correlate e spesso sono alternate

Progettazione Software



- Il progetto è documentato da un insieme di modelli
 - Modello degli oggetti, di sequenza, di transizione di stati, strutturale, data-flow

Programmazione

- Tradurre i modelli del progetto in un programma (codice) e rimuovere gli errori dal programma
- I programmatori effettuano alcuni test sul programma prodotto per scoprire bug e rimuoverli

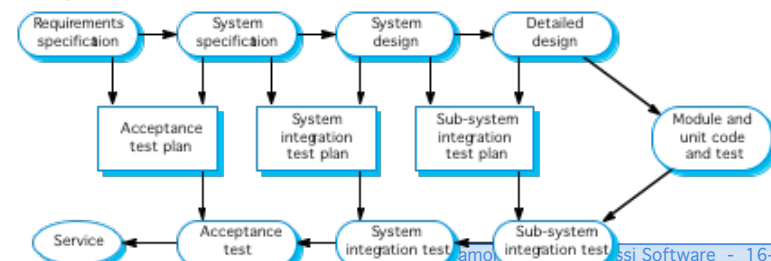


Validazione

- La fase di Verifica e Validazione (V & V) intende mostrare che il sistema è conforme alle specifiche e che soddisfa le richieste (aspettative) del cliente
 - Viene condotta tramite processi di revisione e test del sistema
 - I test mirano ad eseguire il sistema software in condizioni derivate dalle specifiche di dati reali che il sistema dovrà elaborare

Test

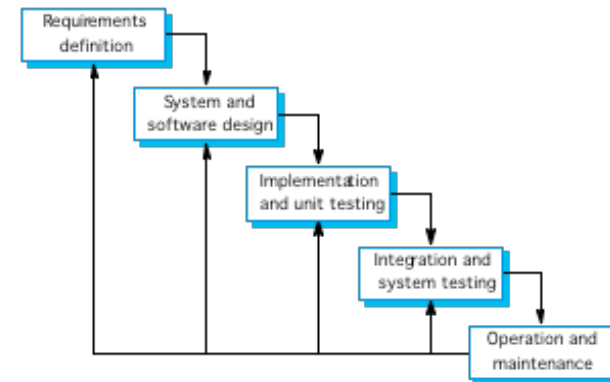
- Test di componenti o unità
 - I singoli componenti sono testati indipendentemente
 - Componenti potranno essere funzioni o oggetti o loro raggruppamenti
- Test di sistema
 - L'intero sistema è testato, dando speciale importanza alle proprietà emergenti
- Test di accettazione (alpha testing)
 - Test condotti con dati del cliente per verificare che il sistema soddisfa le esigenze del cliente



Evoluzione

- Il software è intrinsecamente flessibile e può cambiare
- Al cambiare dei requisiti per cambiamenti dell'ambiente a cui è rivolto (business, hardware, etc.), il software deve evolvere se deve rimanere ad essere utile

Cascata (Waterfall) [Royce 1970]



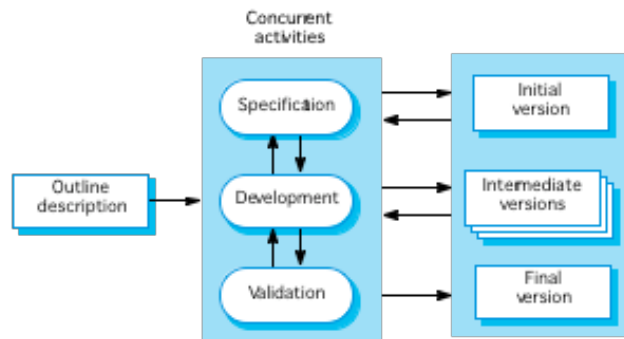
Cascata (Waterfall)

- Il primo dei processi (anni '70), derivato da altri processi di ingegneria
- Focalizza sul prodotto completo
- Si comincia la fase successiva solo se la fase precedente è completa
 - Prima specifica tutto, poi produci tutto, poi testa tutto, ...
- Processo statico con tanta documentazione
 - Lungo tempo per ottenere il prodotto
 - Poche interazioni con i clienti (solo nella fase iniziale)
 - Difficoltà ad introdurre cambiamenti richiesti dal cliente
 - + Consistenza tra artefatti
 - + Utile se i requisiti sono stabili e chiaramente definiti

Evolutivo

- Sviluppo per esplorazione
 - L'obiettivo è di lavorare con i clienti ed evolvere fino ad un sistema finale uno schema di specifiche iniziali
 - Dovrebbe partire da requisiti ben chiari ed aggiungere nuove caratteristiche definite dal cliente
- Sviluppo Build and Fix
 - Documentazione inesistente o quasi
 - Comprensione limitata del sistema da produrre
 - Costruzione della prima versione e modifica fino a quando il cliente è soddisfatto

Evolutivo



Evolutivo

- Problemi
 - Tempi lunghi
 - Sistemi difficilmente comprensibili e modificabili, probabilmente non corretti
 - Mancanza di visione d'insieme del progetto
- Applicabilità
 - Sistemi di piccole dimensioni
 - Parti di sistemi grandi (es. interfaccia utente)
 - Sistemi con vita breve (es. prototipi)

Altri Processi

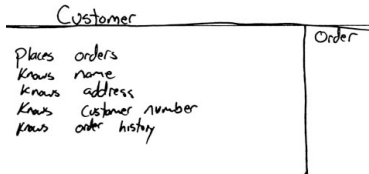
- Sviluppo Incrementale
 - Sono implementate prima le funzionalità di base (o prioritarie)
 - Al codice sviluppato in precedenza è aggiunto altro codice per un altro gruppo di funzionalità
 - Si ripete il passo precedente, fino a completamento
- Processo basato su COTS (componenti esistenti)
 - Analisi dei componenti
 - Modifica requisiti
 - Progettazione tramite riuso
 - Sviluppo ed integrazione

Sviluppo Agile [Cockburn 2002]

- Sono più importanti auto-organizzazione, collaborazione, comunicazione e adattabilità rispetto a ordine e coerenza del progetto
- Privilegiare
 - Individui rispetto a processi e strumenti
 - Disponibilità di software funzionante rispetto alla documentazione
 - Collaborazione con il cliente rispetto alla negoziazione dei contratti
 - Pronta risposta ai cambiamenti rispetto all'esecuzione di un piano
- Agilità
 - Considerare positivamente le richieste di cambiamento anche in fase avanzata di sviluppo
 - Fornire software funzionante frequentemente
 - Costruire progetti con gruppi di persone motivate
 - Continua attenzione all'eccellenza tecnica

eXtreme Programming (XP) [Beck 2000]

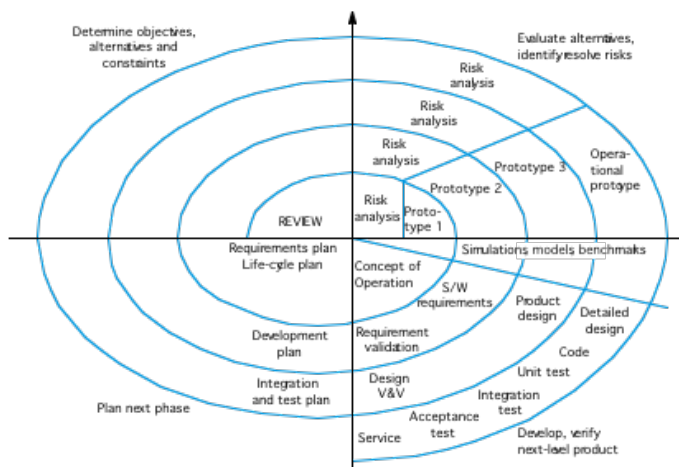
- Approccio basato sullo sviluppo e la consegna di piccoli incrementi di funzionalità
 - Solo 2 settimane per lo sviluppo degli incrementi
 - Piccoli gruppi di sviluppatori (da 2 a 12 persone)
 - Costante miglioramento del codice
 - Poca documentazione (uso di CRC), enfasi su comunicazione diretta tra persone
 - Iterazioni corte e di durata costante
 - Coinvolgimento di sviluppatori, clienti e manager
 - Testabilità dei prodotti e prodotti testati sin dall'inizio
- Adatto per progetti in cui
 - I requisiti non sono stabili, XP è fortemente adattativo
 - I rischi sono grandi, es. tempi di consegna brevi, software innovativo per gli sviluppatori



A Spirale [1988]

- Focalizza su tanti prodotti parziali (sottosistemi funzionali)
- Ogni loop è una fase, con i seguenti obiettivi
 - Identifica obiettivi specifici
 - Valuta rischi
 - Produce una parte e la valida
 - Revisiona progetto e pianifica
- Processo agile
 - + Poco tempo per la prima versione del prodotto
 - + Opportunità di interagire con il cliente
- Ogni fase produce un codice testato ed integrato nel sistema complessivo

A Spirale



Settori del processo a Spirale

- Stabilire obiettivi
 - Gli obiettivi per la fase corrente sono identificati
- Valutare il rischio e ridurlo
 - I rischi sono valutati ed attività sono intraprese per ridurre quelli più importanti
- Sviluppo e validazione
 - Secondo uno dei modelli precedenti
- Pianificazione
 - Il progetto è revisionato e la prossima fase della spirale è pianificata

Rational Unified Process (RUP)

- Cinque attività: avvio, elaborazione, costruzione, transizione e produzione
 - Ripetute più volte e possibilmente condotte in parallelo
- Fase di avvio
 - Comunicazione con il cliente
 - Identificazione dei requisiti del sistema
 - I requisiti vengono descritti da un insieme di casi d'uso, ovvero da ciò che è richiesto da ogni classe di utenti
 - Identificazione di una architettura di base
 - Indicazione dei principali sottosistemi e relative funzioni e caratteristiche
 - Pianificazione
 - Identificazione di risorse, principali rischi, un piano dei tempi e le fasi da seguire per lo sviluppo di parti del software

Rational Unified Process (RUP)

- Fase di elaborazione
 - Comunicazione con il cliente
 - Raffinamento ed ampliamento dei casi d'uso
 - Espansione della rappresentazione dell'architettura
- Fase di costruzione
 - Sviluppa o acquisisce i componenti software che serviranno gli utenti nei vari casi d'uso
 - Test sui componenti
 - Integrazione dei componenti
- Fase di transizione
 - Fornisce una versione del sistema su cui sono condotti i beta test
- Fase di Produzione
 - Viene monitorato l'utilizzo del software

Confronto tra Processi

Processo	Punti di forza	Punti di debolezza
Cascata	Approccio regolato da documenti; appropriato per sistemi con requisiti stabili	Non soddisfacente per i clienti per il poco feedback permesso; non permette l'adattamento a requisiti variabili
Spirale	Approccio disciplinato regolato da documenti	Può essere usato solo per sistemi grandi
RUP	Approccio iterativo; uso di componenti; modellazione visuale	Poco chiaro (?)
Build and fix	Approccio per piccoli sistemi che non necessitano manutenzione	Non soddisfacente per sistemi non banali
XP	Permette l'adattamento dei requisiti	Poca documentazione