

# 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 o fasi dello sviluppo
  - Analisi dei requisiti (specifiche)
  - Progettazione (design)
  - Codifica o implementazione (codice)
  - Convalida o testing (approvazione)
  - Manutenzione

# Fase di 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 che il software deve avere
  - Specifiche: descrizione rigorosa delle caratteristiche del software
    - I requisiti tendono ad essere granulari (ovvero: molti e piccoli)
  - Feature: un set di requisiti correlati tra loro
    - Una feature permette all'utente di soddisfare un obiettivo
- Fasi per l'*ingegneria dei requisiti* [vedi lezioni da 5 a 7]
  - (1) Studio di fattibilità, (2) Analisi dei requisiti, (3) Specifica dei requisiti, (4) Convalida requisiti
- Requisiti
  - Funzionali: Cosa il sistema deve fare (funzionalità)
  - Non-funzionali: Come il sistema lo fa (es. affidabilità, efficienza, prestazioni, manutenibilità, etc.)

# Esempi di Feature e Requisiti

- Feature di Firefox 3.6
  - Browsing privatamente: navigazione del web senza lasciare tracce
  - Password manager: ricordare le password dei siti, senza usare pop-up
  - Awesome Bar: trovare i siti preferiti in pochi secondi
  - One-click bookmark: bookmark, cerca e organizza siti web velocemente e facilmente
- Requisiti (sintetici) di Firefox 3.7
  - Eseguire i plug-in in un processo separato per migliorare la stabilità dell'applicazione e diminuire i tempi di risposta
  - Migliorare i tempi di startup
  - Ottimizzare caricamento delle pagine

# Progettazione ed implementazione

- Fase di Progettazione [vedi lezioni da 8 a 18]
  - Il processo che stabilisce la struttura software che realizza le specifiche
  - Attività della progettazione
    1. Suddivisione dei requisiti
    2. Identificazione sottosistemi, ovvero progettazione architettura software
    3. Specifica delle responsabilità dei sottosistemi
    4. Progettazione di: interfacce, componenti, strutture dati, algoritmi
  - Ognuna delle attività suddette produce un documento corrispondente (o integra un documento già esistente) che descrive un modello
    - Modello degli oggetti, di sequenza, di transizione stati, strutturale, data-flow
- Fase di Implementazione
  - Produce un programma eseguibile a partire dalla struttura stabilita
- Progettazione ed implementazione sono attività correlate e spesso sono alternate

## Fase di Implementazione

- Consiste nella programmazione ovvero nella traduzione dei modelli del progetto in un programma (codice) e della rimozione degli errori dal programma
- I programmatori effettuano alcuni test sul programma prodotto per scoprire bug e rimuoverli
- Per rimuovere i bug
  1. Localizzare l'errore nel codice
  2. Rimuovere l'errore nel modello e poi nel codice
  3. Effettuare nuovamente il test nel programma



## Fase di Convalida (Verifica & Validazione)

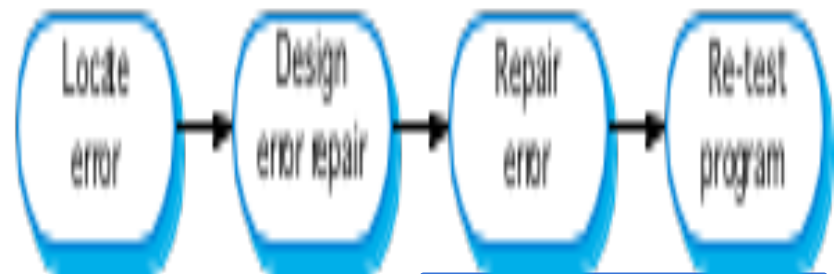
- La fase di convalida o Verifica e Validazione (V & V) del sistema software intende mostrare che il sistema software è conforme alle specifiche e che soddisfa le richieste (aspettative) del cliente
  - Viene condotta tramite processi di revisione e test del sistema software
  - I test mirano ad eseguire il sistema software in condizioni derivate dalle specifiche di dati reali che il sistema software dovrà elaborare
- [vedi lezione 21]

## Fase di Test

- Test di componenti o unità (unit test)
  - 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 dagli sviluppatori con dati del cliente per verificare che il sistema soddisfi le esigenze del cliente
- Beta test: test condotti da alcuni clienti sul prodotto quasi completo
- Prodotti software corrispondenti alle varie fasi di test
  - versione alfa, versione beta, versione golden

## Quadro riassuntivo

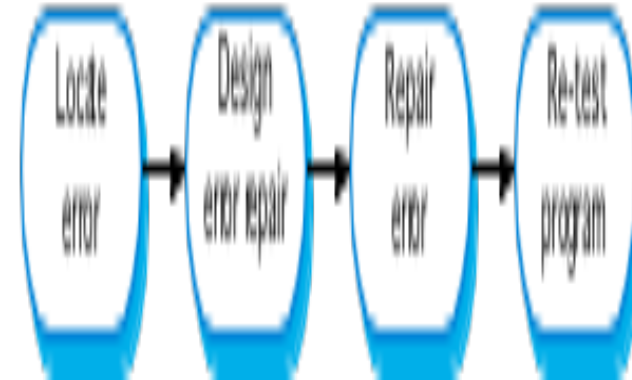
- Dai requisiti (R) otteniamo il documento della specifica dei requisiti (SRS)
- Dall'SRS ricaviamo il design del sistema (DS)
- Dal DS ricaviamo il design dettagliato (DD)
- Da DD ricaviamo codice e test
- Da DS e da DD ricaviamo come integrare i sottosistemi e come fare i test di sistema
- Da R e SRS ricaviamo come fare i test di accettazione



# 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
- [vedi lezione 19]

# Cascata (Waterfall) [Royce 1970]



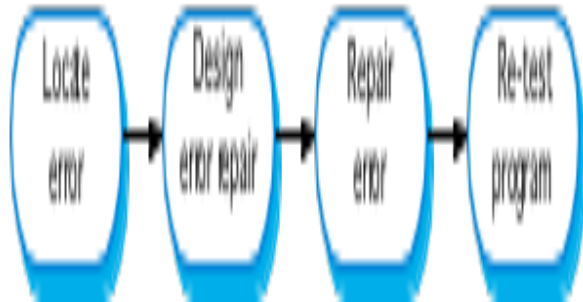
## Cascata (Waterfall)

- 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 i cambiamenti richiesti dal cliente
  - + Consistenza tra artefatti
  - + Ampia documentazione
  - + Utile se i requisiti sono stabili e chiaramente definiti
  - + Usato principalmente per sistemi grandi, complessi, critici, per gestire team numerosi
  - + Alta qualità del codice prodotto

## Processo Evolutivo

- Il processo evolutivo ha due varianti: esplorazione e Build and Fix
- Sviluppo per esplorazione
  - Gli sviluppatori lavorano con i clienti
  - Dalle specifiche iniziali si arriva per mezzo di trasformazioni successive (evoluzione) fino al sistema software finale
  - 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
    - Costruire la prima versione e modificarla fino a che il cliente è soddisfatto
    - Fase di design pressoché inesistente
    - Codice prodotto di bassa qualità

## 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
  - Singole parti di sistemi grandi (es. interfaccia utente)
  - Sistemi con vita breve (es. prototipi)

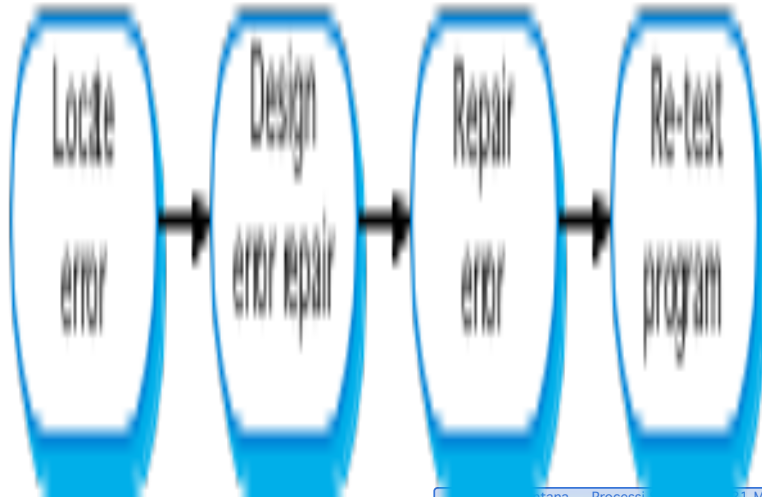
## Altri Processi

- Processo di 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 CBSE o basato su COTS
  - COTS = componenti esistenti (Components Off The Shelf)
  - Analisi dei componenti esistenti
  - Modifica dei requisiti (?)
  - Progettazione tramite riuso
  - Sviluppo ed integrazione

## A Spirale [Boehm 1988]

- Focalizza su tanti prodotti parziali (sottosistemi funzionali)
- Ogni loop (giro della spirale) è una fase (es. loop per requisiti)
- Ogni loop consiste dei seguenti settori
  1. Identificazione obiettivi specifici per la fase corrente
  2. Valutazione rischi del progetto
    - Terminologia: Rischio = qualcosa che può impedire il *successo* e che è sconosciuta
    - Successo: soddisfare tutti i requisiti. Attributi del rischio: (i) probabilità di occorrenza; (ii) impatto sul progetto (ovvero gravità, danno peggiore)
  3. Produzione di una parte e convalida della parte
  4. Revisione del progetto e pianificazione fase successiva
- 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 convalida
  - Secondo uno dei modelli precedenti
- Pianificazione
  - Il progetto è revisionato e la prossima fase della spirale è pianificata