Introduzione

Leggere Cap. 1 Ghezzi et al.

università di ferrara
DA SEICENTO ANNI GUARDIAMO AVANTI.

Sommario

- Definizione
- Nascita dell'ingegneria del software
- Ruolo
- Relazione con altre discipline

Il software

- Il ***software*** e` definito come:

  i programmi, le procedure, e l'eventuale documentazione associata e i dati relativi all'operatività di un sistema di elaborazione.

Definizione di Ingegneria del software

- Nel glossario dell'IEEE ("IEEE Standard Glossary of Software Engineering", 1990), l'***ingegneria del software*** e` definita come:

  applicazione di un approccio sistematico, disciplinato e quantificabile allo sviluppo, all'operatività e alla manutenzione del software.

Campo di applicazione

- Costruzione di sistemi software
  - grandi e complessi
  - prodotti da squadre
  - esistenti in diverse versioni
  - in funzione per anni o decenni
  - che subiscono evoluzioni

Storia: anni Cinquanta

- Il software doveva risolvere problemi relativamente semplici e ben compresi
- L'utente e il programmatore erano la stessa persona
- Ingegneria del software = programmazione

Storia: anni Sessanta

- Calo dei costi dell'hardware, maggiore diffusione dei computer
- Software applicato a problemi vari, più complessi, non sempre ben compresi
- Utente != programmatore

Storia: anni Sessanta

- Problemi:
  - difficoltà di comunicazione fra utenti e programmatori, requisiti del software non chiari
  - tecniche di programmazione non adatte a grossi sistemi
  - parti dei sistemi software fortemente accoppiate:
    - necessità di coordinazione continua fra i programmatori
    - difficoltà nella sostituzione di chi abbandonava i progetti
    - la modifica di una parte influenzava l'intero sistema

Storia: anni Sessanta

- Esempio di progetto: sviluppo del sistema operativo OS 360 per i mainframe IBM della famiglia 360
- Viene coniata l'espressione "crisi del software"

Storia: anni Sessanta

- Risultato:
  - sistemi poco soddisfacenti, poco affidabili, che non rispettavano i tempi e i costi previsti
- Si comprende la necessità di un approccio sistematico alla produzione del software
- L'espressione "ingegneria del software" nasce in questo periodo

Storia: dagli anni Sessanta a oggi

- L'ingegneria del software ha prodotto:
  - Linguaggi e strumenti
  - Approcci rigorosi a
    - specifica
    - verifica
  - Standard
  - Metodologie pratiche, organizzative e gestionali

L'ingegneria del software è una disciplina relativamente giovane

- I problemi
  - non esistono parametri universalmente accettati per definire i requisiti
  - le tecniche formali sono poco sviluppate e di applicabilità incerta
  - (quasi) ogni nuovo prodotto pone problemi nuovi
- Intuito ed esperienza giocano ancora un ruolo fondamentale

“No silver bullet”

- In un articolo del 1987 Brooks afferma che non esiste un “proiettile d’argento”, ovvero una soluzione magica ai problemi del software (dal fatto che i proiettili d’argento uccidono i lupi mannari)
- La progettazione e lo sviluppo del software richiedono sforzo intellettuale, creatività e tempo

Dimensione economica

- Sempre più sistemi sono sotto controllo software
- Spese per il software passate da $140*10^9$ $ nel 1985 a $800*10^9$ $ nel 2000
- Necessità di
  - prodotto sempre più affidabili
  - contenimento dei costi

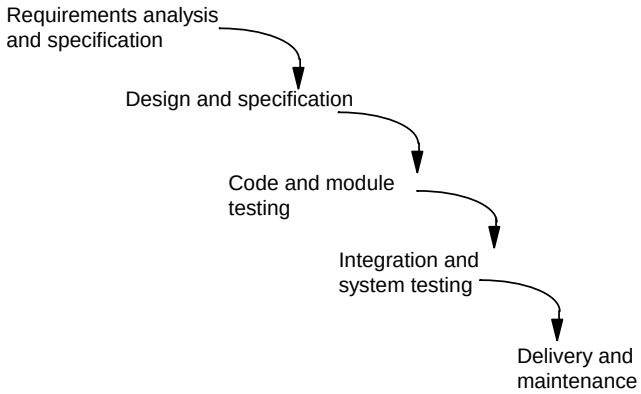
Ruolo dell’ingegnere del software

- Non solo programmare:
  - comprendere i requisiti e tradurli in specifiche precise
  - progettare
  - operare a diversi livelli di astrazione
  - comunicare con la squadra, gli utenti, i clienti
  - rispettare tempi e costi

Attività nell’ingegneria del software

- Analisi e specifica dei requisiti
- Progettazione e specifica di sistema
- Codifica e verifica di modulo
- Integrazione e verifica di sistema
- Consegna e manutenzione

Modello a cascata del ciclo di vita del software



Analisi e specifica dei requisiti

- Dopo studio di fattibilità
- Obiettivo: identificare e documentare i requisiti che il sistema software dovrà rispettare (*che cosa*)
- In caso di sistemi innovativi, richiede ampia interazione fra cliente e ingegnere
- Il documento dei requisiti dovrebbe
  - essere comprensibile per l'utente finale
  - poter portare alla creazione del manuale utente e di casi di test

Progettazione e specifica di sistema

- Obiettivo: progetto di un particolare sistema che soddisfi i requisiti (*come*)
- Divisa in due fasi:
  - Progetto architetturale: organizzazione globale del sistema in componenti di alto livello e loro interazioni
  - Progetto dettagliato: scomposizione a livelli di dettaglio sempre maggiori, fino a una specifica diretta della codifica

Codifica e verifica di modulo

- Produzione del codice sorgente secondo il progetto dettagliato del sistema
- Verifica che i singoli moduli prodotti rispettino la loro specifica

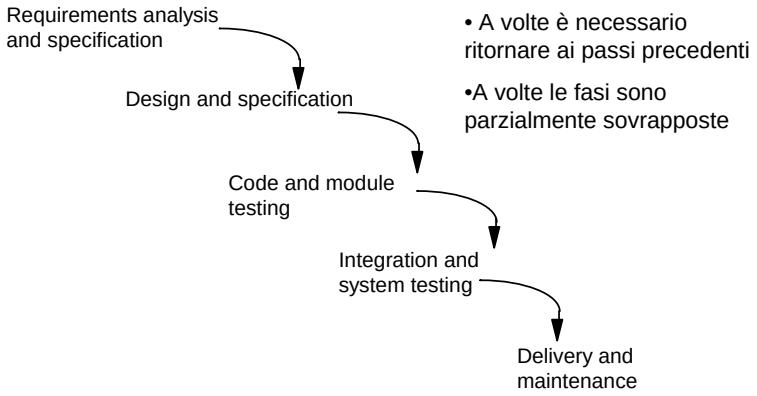
Integrazione e verifica di sistema

- I singoli moduli prodotti vengono
  - assemblati
  - integrati
  - testati come sistema unico

Consegna e manutenzione

- Quando il sistema ha superato tutti i test viene consegnato all'utente
- Successivamente, si rende necessaria la manutenzione del sistema:
  - per correggere i difetti che si manifestano
  - per adattare il sistema a cambiamenti dei requisiti (evoluzione)

Modello a cascata del ciclo di vita del software



- A volte è necessario ritornare ai passi precedenti
- A volte le fasi sono parzialmente sovrapposte

Relazione con altri campi dell'informatica

- Linguaggi di programmazione
- Sistemi operativi
- Basi di dati
- Intelligenza artificiale
- Metodi formali

Relazione con linguaggi di programmazione

- Forte influenza dell'ingegneria del software sui linguaggi di programmazione: ad esempio,
  - Costrutti per programmazione modulare
  - Costrutti per la gestione delle eccezioni
  - Separazione tra interfaccia e implementazione (Java, Ada 95)

Relazione con linguaggi di programmazione

- Influenza inversa:
  - Specifiche di progetto rigorose, adatte all'implementazione in un linguaggio di programmazione
  - Formalizzazione del linguaggio con cui si scrive l'input di un software (da Job Control Language di OS 360 a shell di Unix)
  - Formalizzazione per favorire l'automazione

Sistemi operativi

- Hanno avuto influenza notevole sull'ingegneria del software:
  - primi sistemi di grandi dimensioni
  - macchine virtuali, livelli di astrazione, separazione fra politiche e meccanismi
- Influenza inversa:
  - modularità e portabilità dei sistemi operativi moderni
    - ad es., separazione dell'interprete dei comandi
  - architetture a micro-kernel

Basi di dati

- Hanno influenzato l'ingegneria del software:
  - indipendenza dalla rappresentazione dei dati (astrazione, separazione degli interessi)
  - uso di DBMS come componente risolve "gratis" i problemi legati all'accesso concorrente a grosse quantità di dati

Basi di dati

- Influenza inversa:
  - necessità di nuovi modelli per usare DBMS come strumenti per l'ingegneria del software
    - memorizzazione di codice sorgente, documenti, file binari
    - memorizzazione di diverse versioni dello stesso oggetto
    - transazioni lunghe

Intelligenza artificiale

- Influenza su ingegneria del software
  - Tecniche per produrre sistemi software grossi e complessi con requisiti incerti (sviluppo esplorativo)
  - Utilizzo della logica
- Influenza inversa
  - Sistemi di IA modulari (a regole o shell)
  - Concezione di tecniche di intelligenza artificiale da applicare a ingegneria del software
    - Assistenti di programmazione
    - Elaborazione del linguaggio naturale per le interfacce

Sistemi formali

- Modelli formali utilizzati in ingegneria del software (specifiche e modelli)
  - automi a stati finiti
  - automi a pila
  - reti di Petri
  - logica matematica
- Inversamente: sviluppo di nuovi metodi formali:
  - specifiche algebriche
  - tipi di dato astratti
  - logica temporale

Relazione con altre discipline

- Scienze organizzative
  - Modelli gestionali applicati a produzione del software per effettuare stime, gestire le risorse umane, monitorare il processo
  - Necessità di nuovi modelli
- Ingegneria dei sistemi
  - Software come componente di un sistema più complesso