Ingegneria del software

Friday, 17 March 2023 10:33

Siccome ho notato di non avere compreso nulla riguardante questa materia, ora mi metto a
tutte le lezione, seguendo i miei appunti, per poterli riscrivere.

- Software è un programma per un computer **e** la documentazione associata
  Essi possono essere:
  - Generici, creato per più clienti ex office
  - Personalizzati, quindi un cliente vuole un software specifico

  E può essere creato:
  - Sviluppando nuovi programmi da 0
  - Usando software già esistenti, ex libreire ex react

- Caratteristiche di un buon software:
  - Mantenibilità: chiunque può modificare il nostro codice siccome esso è leggibile
  - Affidatezza: Ex sicurezza/protezione/no danni fisici
  - Efficienza: No spreco risorse
  - Accettabilità: User friendly, utilizzabile e compatibile con gli altri sistemi
    Questo è estremamente importante siccome, molti progetti sono falliti sic
    accettati dall'utente finale

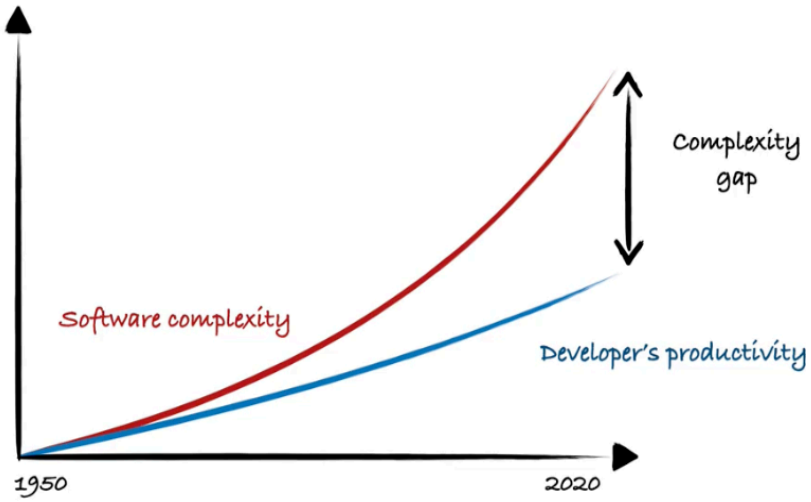
- Parole chiavi:
  - Quality, noi vogliamo un software di qualità
  - Build, costruzione
  - Needs, siccome ciò che noi sviluppiamo deve incontrare le necessità del cliente
  - People, quando si crea un software tante persone sono coinvolte
  - Processes, abbiamo tanti processi da seguire

- Significato: ingegneria che si occupa di tutti gli aspetti per la creazione di un software,
  - Disciplina ingegneristica siccome usa teorie per tenere conto dei vincoli organizzativi/finanziari
  - Non è solo processo tecnico di sviluppo, ma anche gestione progetto, gli strume per la produzione di software

- La disciplina dell'ingegneria del software ci insegna:
  - Metodologie
  - Tecniche
  - Strumenti

Affinché possiamo rispettare:
- Vincoli temporali/economiche
- Richieste clienti

- Perché è nato questo? Siccome, col tempo il software ha iniziato a diventare sempre e complicato mentre la produttività degli sviluppatori è incrementata di poco, e quindi è necessaria una disciplina che permettesse una maggiore facilità della creazione di soft



Il nostro obiettivo è quello di diminuire il gap di complessità, che è la differenza fra co software e la produttività dello sviluppatore .

- Noi abbiamo un idea, e vogliamo un software, e questo "vogliamo" si chiama "process
Processo software: spezzettiamo un problema che ci permette di arrivare da un idea a
| \-> Gestiamo la complessità
\-> Ci dice quali attività si susseguono,
E le attività sono:
  - Analisi dei requisiti, comprendere cosa dobbiamo progettare/implementa
    - Enfatizza un problema ed i requisiti -> Cosa dobbiamo fare
    - Enfatizza sull'identificazione dei concetti, o degli oggetti del dominio
      Ex. il concetto volo, concetto aereo, concetto pilota
      Qui si utilizza OO, object oriented, che si divide:
      - Noi prendiamo un concetto del dominio, Ex aereo
      - OOA: Prendiamo aereo e lo visualizziamo con campi reali, ex c
      - OOD (design): qui creiamo la classe software
      - OOP: Trasformiamo OOD in codice
  - Progettazione, noi progettiamo i componenti software
    - Enfatizza una soluzione del problema -> Come dobbiamo fare
    - Enfatizza concetti software che collaborano per i requisiti
      Prendiamo gli oggetti dell'analisi, e li trasformiamo
      Ex. L'aereo che ha attributo codiceRegistrazione e metodo getVoliEf

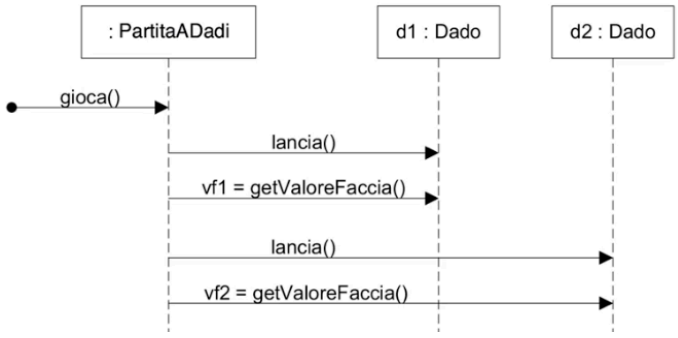
    - Implementazione, trasformiamo progetto->codice
    - Evoluzione
    - Validazione, il test che tutto è corretto

  |-> Definisce anche:
    - Artefatti, aka gli output di ogni attività
    - Ruoli, quindi le persone
    - Post/pre condizione che sono i requisiti per continuare
    - Attività

- Esempio veloce.
  - Definizione casi d'uso, che servono per definire i requisiti
    Sono delle storie scritte che ci dice come il sistema verrà usato dall'utente
    Ex. Giocatore chiede lanciare dadi, il sistema presenta il risultato: se il valore tot
    giocatore ha vinto
  - Definire modello dominio, che dobbiamo definire i concetti del dominio
    Ex.

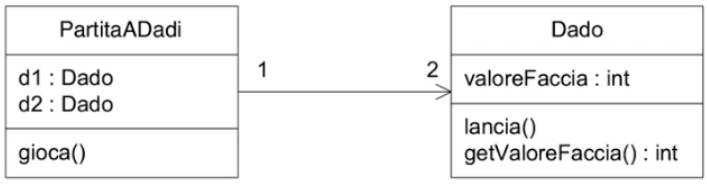
    Queste non sono classi software
    Ci mostra le associazioni ed attributi
    Aka i concetti del caso d'uso
    Questo è un modello classi in UML

  - Definire diagrammi di iterazione, come i software interagiscono per risolvere re
    Ex.



    Qui possiamo vedere come
    Tutto interagisce
    Questo è un diagramma di iterazion
    E ci mostra come interagiscono per
    Soddisfare casi d'uso

  - Definire diagrammi delle classi di progetto, come è strutturato il nostro sistema
    Ed ora finalmente possiamo creare ciò che ci piace
    Ex.



    Qui creiamo un'associazione
    Classi software in UML

    Ed ora dal progetto si implementa
    Ex.

```
public class PartitaADadi {
    private Dado d1;
```

```
    private Dado d1;
    private Dado d2;

    public PartitaADadi() {
        d1 = new Dado();
        d2 = new Dado();
    }
    public void gioca() {
        int vf1, vf2;
        d1.lancia();
         vf1 = d1.getValoreFaccia();
         d2.lancia();
         vf2 = d2.getValoreFaccia();
         ...
    }
}
```

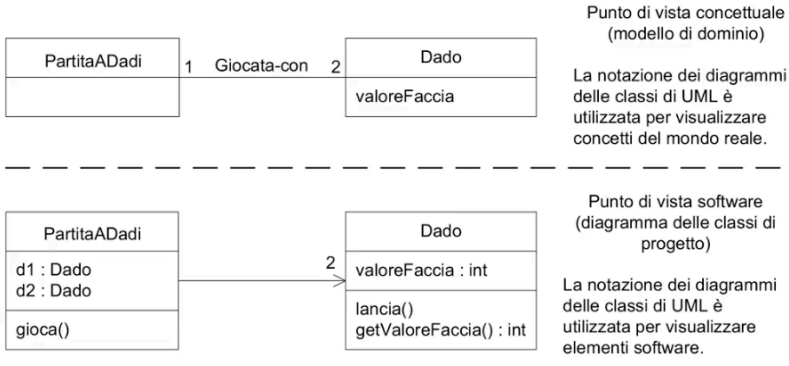
```
public class Dado {
  private int valoreFaccia;

   public Dado() {
      ...
   }

   public void lancia() {
      ...
   }

   public int getValoreFaccia() {
      return valoreFaccia;
   }
}
```

- Qui abbia misto dei quadrati, ma cosa sono questi quadrati?
  Questi quadrati sono fatti con UML:
  - Linguaggio di modellazione sistemi software
  - E' ormai uno standard, e quindi permette la divulgazione
  - NON è una metodologia, ma è un linguaggio visuale
  - Unified:
    - Può essere usato in tutti il ciclo di sviluppo
    - In qualsiasi sistema
    - Con qualsiasi linguaggio
    - Con qualsiasi processo di sviluppo
  - Esso modella i sistema come oggetti che collaborano
    E la struttura può essere:
    - Statica, quali tipi sono necessari e come sono correlati
    - Dimatica, il ciclo di vita e come collaborano per arrivare all'obiettivo
  - Ed esistono 3 tipologie per applicare:
    - Abbozzo, informali ed incompleti ma serve per dare l'idea
      Noi useremo questo visto che useremo la modellazione agile
    - Progetto
    - Linguaggio programmazione
  - L'Uml può implementare:



: Concettuale, descrivono og
del mondo reale o dominio

: Software
Descrivono astrazione o co