

introduzione al corso di ingegneria del software

a.a. 2003-2004

contatti con i docenti

Maurizio Pizzonia

pizzonia@dia.uniroma3.it

orario ricevimento:

mercoledì 17:30

(presentarsi entro le 18:00)

Valter Crescenzi (esercitazioni)

crescenz@dia.uniroma3.it

l'ingegneria del software

- www.pcwebopedia.com definisce l'ingegneria del software nel seguente modo

“The computer science discipline concerned with **developing large applications**. Software engineering covers not only the **technical aspects** of building software systems, but also **management issues**, such as directing **programming teams**, **scheduling**, and **budgeting**.”

obiettivi del corso

- L'obiettivo di questo corso è...

fornire competenze pratiche su metodologie e tecnologie informatiche e di supporto al lavoro di gruppo per progetti software di dimensioni medie e grandi.

una scienza non esatta

- l'ingegneria del software non è matematica
 - niente teoremi
 - niente certezze
 - molte linee guida

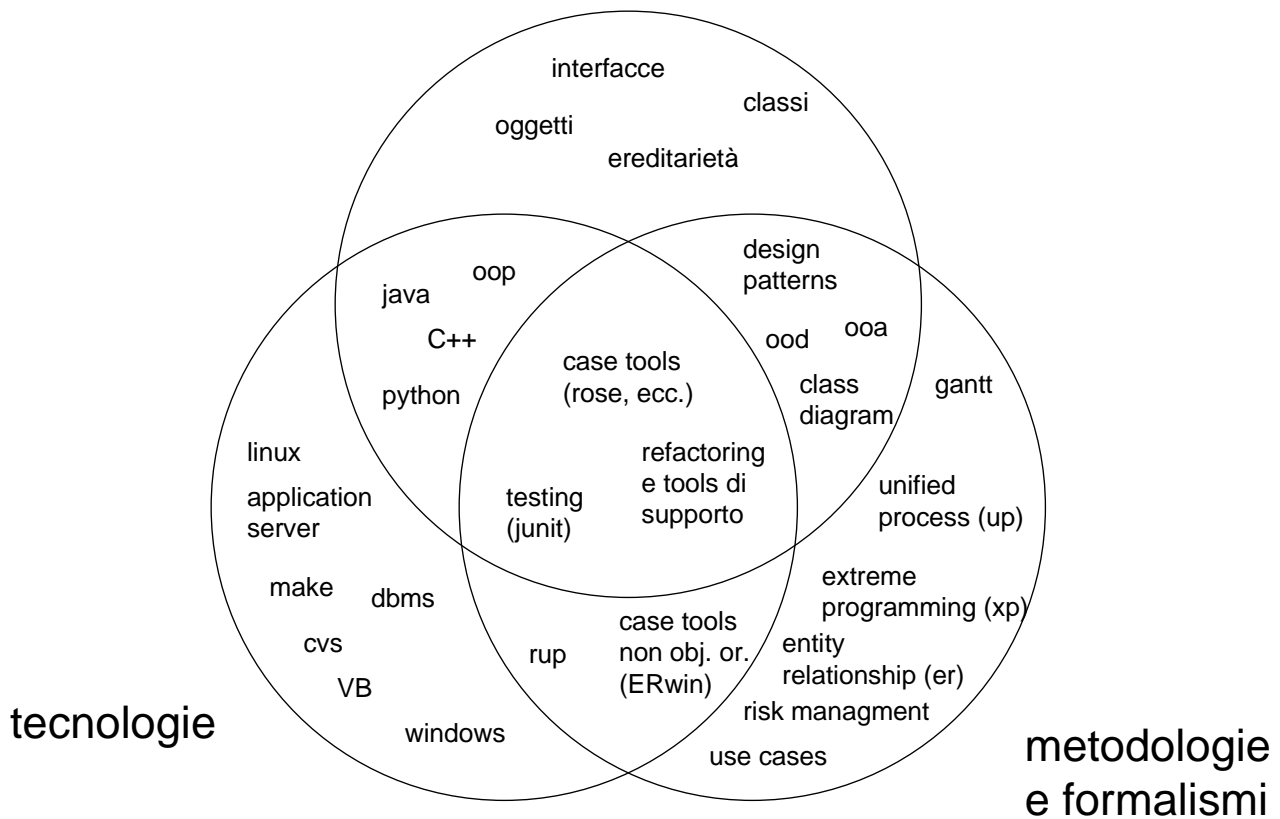
una vera competenza su progetti software di *grandi dimensioni* si acquista solo in anni di esperienza!

- noi studiamo metodologie e tecnologie *applicabili* a sw di grandi dimensioni,
 - 50 ore di corso non sono sufficienti per diventare ottimi project manager!

programma del corso in sintesi

- principali metodologie
 - UP, XP
- object oriented analysis/design/programming
 - design patterns, refactoring, testing
- tecnologie
 - cvs, make, JUnit
 - linguaggi (quelli che servono)
 - come scegliere
 - come apprendere nuove tecnologie
- sviluppo di un progetto

orientazione agli oggetti



il metodo didattico

- limitato studio del materiale
- molta applicazione pratica di
 - metodologie
 - tecnologie
- molta discussione in aula
 - su casi e problemi concreti
 - mediamente un giorno a settimana
- lavoro di gruppo
 - pair programming

i progetti

- presentati il terzo giorno di corso
 - altre proposte sono benvenute
- sviluppo di software con approccio metodologico
 - es. risk management, UML, testing
- svolti in gruppi di 4 persone
- uso metodico di cvs
- valutazione in base al corretto approccio metodologico
 - il lavoro non necessariamente deve/può dar luogo a software completamente funzionante
 - ma deve almeno chiarire inequivocabilmente che sareste in grado di portarlo a termine e in che tempi

i progetti

- stato di avanzamento lavori
 - una lezione a settimana dedicata ad un report sui progetti
 - circa 7 progetti ⇒ circa 10 minuti per ciascun progetto
- supporto dei docenti
 - pochissimo supporto sulla tecnologia
 - fa parte del lavoro metodologico capire i rischi legati alla singola tecnologia e fare le scelte corrette
 - supporto sulle metodologie
 - supporto sulla progettazione

valutazione

- contributo al progetto
basata su
 - contributi tracciati con cvs
 - i contributi vengono tracciati per studente tramite i log
 - contributi dati in coppia sono ammessi e benvenuti purché le coppie non siano sempre le stesse
 - interventi nei rapporti settimanali e a lezione
- prova scritta
 - tema o domande
- project management
 - ciascun gruppo avrà una persona di riferimento che verificherà il corretto svolgimento del lavoro

on-line

- <http://www.dia.uniroma3.it/~pizzonia/swe>
- newsgroup: dia.corsi.swe
- registrazione obbligatoria entro il 4 dicembre
 - per questioni organizzative registratevi subito (oggi!)
 - vi si chiede
 - username e password (per il repository cvs)
 - competenze su vari argomenti
 - per organizzare progetti e didattica
- parte del materiale on-line richiede una password
 - **non** è quella data per la registrazione
 - chiedere al docente

materiale didattico

- slides scaricabili dal sito
- altre fonti
 - approccio didattico
 - C. Larman, "Applying UML and Patterns". Prentice Hall.
 - K. Beck, "Extreme Programming Explained". Addison-Wesley.
 - J. Link, P. Frohlich, "Unit Testing in Java". Morgan Kaufmann.
 - A. Shalloway, J. R. Trott, "Design Patterns Explained". Addison-Wesley.
 - approccio didattico/reference
 - P. Kruchten, "Rational Unified Process - Introduzione". Addison-Wesley.
 - M. Fowler, K. Scott, "UML Distilled". Addison-Wesley.
 - www.extremeprogramming.org
 - approccio reference
 - E. Gamma, R. Helm, R. Johnson, J. Vlissides, "Design Patterns". Addison-Wesley.
 - M. Fowler, "Refactoring". Addison-Wesley.