

# Guida rapida all'insegnamento

## Ingegneria del software

Docente: Giuseppe Scollo

Università di Catania, sede di Comiso (RG)  
Facoltà di Scienze Matematiche, Fisiche e Naturali  
Corso di Studi in Informatica applicata, AA 2008-9

### Indice

1. Guida rapida all'insegnamento
2. Organizzazione dell'insegnamento
3. Obiettivi formativi
4. Attività formative
5. Relazione con altri insegnamenti
6. Programma delle lezioni (1)
7. Programma delle lezioni (2)
8. Modalità di valutazione
9. Testi consigliati
0. Strumenti per l'interazione formativa
- .1. Strumenti open-source

## Organizzazione dell'insegnamento

- **modalità:** *blended e-learning*
  - *strumenti per l'interazione in rete: sito dell'insegnamento, Forum, Wiki, ...*
- **sviluppo progettuale cooperativo**
  - *non solo di progetti di laboratorio, ma anche dello stesso progetto formativo dell'insegnamento*
- **condivisione di obiettivi formativi**
- **"imparare a non programmare"**

## Obiettivi formativi

Acquisizione di:

nozioni introduttive di base, principi e metodi dell'ingegneria del software, e di competenze professionali nella disciplina, relative agli aspetti più salienti di

*pianificazione,*

*specificazione e analisi dei requisiti,*

*progettazione, collaudo e manutenzione, nei*

processi di sviluppo e produzione del software

metodi di progettazione e specificazione di architetture software e siti Web

*con particolare attenzione alla progettazione orientata agli oggetti*

competenze professionali nell'uso dei linguaggi standard UML e OCL

*per la modellazione, specificazione e sviluppo di tali architetture*

competenze professionali nell'uso di strumenti di laboratorio

*di supporto a tali attività*

## **Attività formative**

L'organizzazione didattica dell'insegnamento prevede

48 ore di lezione

L'acquisizione di metodi e competenze professionali nella disciplina è sostenuta dallo

studio delle note delle lezioni, e di eventuali altre fonti di approfondimento e dalla sua applicazione nella

soluzione di problemi ed esercizi proposti

e nella

sperimentazione dell'uso di strumenti di laboratorio

per la

realizzazione di progetti autogestiti

## **Relazione con altri insegnamenti**

Questo insegnamento non ha prerequisiti. La conoscenza di un linguaggio di programmazione qualsiasi e un po' di esperienza di programmazione sono desiderabili, ma non strettamente necessarie.

## **Programma delle lezioni (1)**

1. Ingegneria del software nel progetto di siti Web
2. Progettazione di qualità
3. Qualità di prodotti software e di siti Web
4. Usabilità di prodotti software e di siti Web
5. Metodologie di progettazione del software
6. Pianificazione di risorse per la produzione di software
7. Evoluzione del software e gestione del cambiamento
8. Analisi, modellazione e specifica di requisiti
9. Progettazione di architetture software
10. Progettazione di siti web con sistemi di gestione dei contenuti
11. Collaudo del software
12. Metriche e modelli di stima dei costi del software

## **Programma delle lezioni (2)**

13. Modellazione di architetture software, introduzione a UML e MDA
14. Diagrammi delle classi: concetti fondamentali
15. Diagrammi delle classi: concetti avanzati
16. Costruzione di un modello di rete di servizi
17. Diagrammi di interazione
18. Diagrammi di macchine a stati
19. Diagrammi di attività e di interazione generale
20. Altri diagrammi strutturali
21. Specifica di vincoli in modelli UML con OCL
22. Costruzione di modelli con OCL
23. Tipi e costrutti avanzati di OCL
24. Validazione di modelli UML e vincoli OCL

## **Modalità di valutazione**

La verifica del profitto avviene con un colloquio individuale sugli argomenti del programma,

- a partire dal lavoro svolto dallo studente in un progetto,
- preferibilmente realizzato in collaborazione con altri studenti.

Il superamento della prova porta all'acquisizione di 6 crediti.

La valutazione di contributi prodotti dagli studenti in risposta a problemi ed esercizi proposti viene effettuata *in itinere* e dà luogo all'acquisizione di *bonus*, cioè di un incremento del voto relativo al colloquio di esame.

## **Testi consigliati**

### **Testi di riferimento**

G. Scollo, *Note delle lezioni di Ingegneria del software*  
Università di Catania, DMI (2008)

M. Aspeli, *Professional Plone Development*  
Packt Publishing (2007)

H. Sedehi, *Ingegneria economica del software*  
EUCOS, Libreria Italiana (2003)

M. Fowler, *UML Distilled, Terza Edizione*  
Addison-Wesley, Pearson Education Italia (2004)

J.B. Warmer & A.G. Kleppe, *The Object Constraint Language, Second Edition*  
Addison-Wesley (2003)

### **Altri testi consigliati (per consultazione)**

v. Bibliografia

## Strumenti per l'interazione formativa

meno Forum, più Wiki: perché?

meno Forum: non ci sono esercizi specifici per le singole lezioni,  
mentre restano le discussioni di

*organizzazione dell'insegnamento*

*argomenti delle lezioni*

*idee di progetto*

*uso di strumenti di laboratorio*

più Wiki: sono incoraggiati lo sviluppo di **esercizi** e la documentazione *in corso d'opera* dello sviluppo di **progetti**, modelli e specifiche, facendo pieno uso della funzionalità Wiki per il controllo di versione e per la gestione di discussioni

## Strumenti open-source

CMS a confronto:

<http://en.wikipedia.org>

[/wiki/Comparison\\_of\\_content\\_management\\_systems](http://en.wikipedia.org/wiki/Comparison_of_content_management_systems)

<http://www.cmsmatrix.org>

**Plone:** <http://plone.org>

**AndromDA:** <http://www.andromda.org>

**ArgoUML:** <http://argouml.tigris.org>

**BOUML:** <http://bouml.free.fr>

**Gaphor:** <http://gaphor.sourceforge.net>

**Dresden OCL Compiler:** <http://dresden-ocl.sourceforge.net>

**USE - A UML-based Specification Environment:**

<http://www.db.informatik.uni-bremen.de/projects/USE>