

Sistemi Software
(6 CFU – docente: Andrea D'Ambrogio)

Obiettivi dell'insegnamento:

Gli allievi acquisiscono le conoscenze relative agli aspetti metodologici ed applicativi per inquadrare la produzione del software all'interno di una disciplina ingegneristica. Vengono in particolare presentati il processo software e i principali metodi di analisi e progettazione del software; al termine del corso lo studente avrà acquisito le competenze necessarie per conoscere gli aspetti accidentali ed essenziali dei prodotti software (*conoscenza e capacità di comprensione*). In particolare, lo studente avrà avuto modo di apprendere gli strumenti qualitativi e quantitativi per l'analisi e la progettazione di sistemi software, e per la gestione efficace di progetti software (*capacità di applicare conoscenza e comprensione*). Il riferimento a contesti applicativi e casi di studio reali stimolano *autonomia di giudizio* e *abilità comunicative*. Infine, le conoscenze di base dell'ingegneria del software apprese nel corso contribuiscono a sviluppare *capacità di apprendimento* da parte dello studente mettendolo nelle condizioni di poter applicare in maniera autonoma le tematiche affrontate.

Prerequisiti:

Non esistono propedeuticità obbligatorie da rispettare. È tuttavia vivamente consigliato come prerequisito avere sostenuto l'esame di Fondamenti di Informatica.

Metodi di insegnamento: A distanza. Il corso è erogato utilizzando la piattaforma di didattica online Moodle disponibile al link: <http://iol.uniroma2.it/moodle/>. Le lezioni sono erogate attraverso l'utilizzo di strumenti multimediali quali audiofile disponibili sulla piattaforma Moodle che integrano il materiale messo a disposizione dal docente sotto forma di presentazioni/slide delle lezioni e commenti alle stesse. Gli studenti sono invitati a partecipare inoltre ai forum di discussione aperti dal docente sui principali argomenti delle singole lezioni.

Metodi di valutazione: Scritto. La prova scritta tipicamente ha una durata di 2 ore.

Contenuti (programma):

Introduzione: caratteristiche essenziali del software, scopo dell'ingegneria del software e sua evoluzione. Processo software e sue macro-fasi: analisi, progettazione, codifica e manutenzione. Fasi di verifica e convalida. Modelli di processo: build&fix, waterfall, rapid prototyping, incremental, spiral, synch-and-stabilize. Pianificazione e gestione di progetti software. Qualità del software e fattori di qualità. Principi, metodi e linguaggi di analisi e progettazione: approccio strutturato, approccio object-oriented, approccio component-based e approccio model-driven. Illustrazione di casi di studio.

Attività didattica interattiva:

Durante il corso sono previste attività di didattica interattiva (DI) oltre alla didattica erogata (DE) secondo le linee guida ANVUR. Agli studenti è richiesto in particolare di partecipare attivamente a web forum (dimostrazione o suggerimenti operativi su come si affronta problemi di analisi e progettazione di sistemi software in domini applicativi introdotti dal docente). Alla didattica interattiva sono dedicate almeno 6 ore, ovvero minimo 1 ora per ogni CFU del corso. Nello specifico sono previste le seguenti attività: svolgimento a titolo individuale di esercizi di analisi e progettazione software su indicazione del docente. Il docente fornirà opportuni feedback agli esercizi svolti sulla piattaforma Moodle, attraverso strumenti di chat e forum.

Software Systems

(6 Credits – instructor: Andrea D'Ambrogio)

Aim of the Course:

Providing methodological and application-oriented aspects to build software systems according to a disciplined and rigorous approach. Specifically, the course presents the software process, as well as software analysis and design methods. At the end of the course the students get the necessary skills to learn about the basic aspects of software products (*knowledge and understanding*). In particular, the student will have had the opportunity to learn the qualitative and quantitative tools for the analysis and design of software systems, and for the effective management of software projects (*ability to apply knowledge and understanding*). The reference to application contexts and real case studies stimulate *judgment autonomy* and *communication skills*. Finally, the basic knowledge of software engineering learned in the course contributes to developing *learning skills* by the student, enabling him/her to autonomously apply the topics addressed.

Prerequisites:

There is no mandatory prerequisite to meet. However, it is highly recommended to have passed the exam of Foundations of Computer Science (Fondamenti di Informatica).

Teaching methods: On line. The course is taught using the Moodle e-learning platform available at the link: <http://iol.uniroma2.it/moodle/>. Lessons are provided through the use of multimedia tools such as audio and/or video files of the lessons available on the Moodle platform which integrate the material made available by the teacher in the form of presentations/slides and lecture notes and comments. Students are also invited to participate in the discussion forums opened by the teacher on the main topics of the individual lessons.

Examination procedures: Written. The duration of the written test is about 2 hours.

Contents:

Introduction: accidental and essential aspects of software products, software engineering evolution. Software lifecycle models: build & fix, waterfall, rapid prototyping, incremental, spiral. Software lifecycle phases: requirements, specification, preliminary design, detailed design, implementation, integration, operation and maintenance. Quality and quality factors. Software project management, development time and effort. Principles, methods and tools to support the specification, design and documentation of software products and services. Application to case studies.

Interactive teaching:

The course involves some interactive teaching activities (DI), beyond the classroom teaching (DE). Specifically, students are required to participate in open discussions on the Moodle platform on topics suggested by the teacher. Specifically, the following e-tivity are provided: carry out individual activity for the resolution of exercises provided by the teacher. The teacher will provide appropriate feedback to the exercises carried out on the Moodle platform, through chat tools and forums. At least 6 hours are dedicated to interactive teaching (DI), or at least 1 hour for each one of the six ECTS (credits) of the course.