Syllabus

Descrizione corso

Titolo insegnamento	Software Engineering
Codice insegnamento	76269
Titolo aggiuntivo	
Settore Scientifico-Disciplinare	INFO-01/A
Lingua	Tedesco
Corso di Studio	Corso di laurea in Informatica
Altri Corsi di Studio (mutuati)	
Docenti	prof. dr. Claus Pahl, Claus.Pahl@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/36376
Assistente	
Semestre	Primo semestre
Anno/i di corso	2
CFU	6
Ore didattica frontale	40
Ore di laboratorio	20
Ore di studio individuale	90
Ore di ricevimento previste	
Sintesi contenuti	The course introduces the state-of-the-art in software engineering. It aims to demonstrate how this is transferred into practically applicable knowledge and skills for software development.
Argomenti dell'insegnamento	- Software life-cycle: principles and methodologies - Software processes and software project management - Requirements engineering: elicitation and modeling - System modeling and construction: UML, design patterns - Software testing and management: principles and techniques - Recent software engineering topics: DevOps, Cloud, SE and AI

Parole chiave	Software processes, requirements engineering, software design, software modeling, testing and maintenance.
Prerequisiti	Students should have completed the following courses: Introduction to Programming, and Programming Project.
Insegnamenti propedeutici	
Modalità di insegnamento	The course includes frontal lectures, exercises, and projects.
Obbligo di frequenza	Generally, attendance is not compulsory, but non-attending students have to contact the lecturer at the start of the course to agree on the modalities of the independent study. Non-attending students will still be assessed under the same modalities as regularly attending students regarding the project organization/assessment and final exam.
Obiettivi formativi specifici e risultati di apprendimento attesi	Knowledge and Understanding - D1.8 To have a thorough knowledge of the main fundamentals techniques and methods of software design, development and maintenance Applying knowledge and understanding - D2.5 Be able to apply the own knowledge to the , design, development and testing of information systems which satisfy given requirements - D2.10 Be able to solve typical problems in computer science based on software engineering methodologies, such as the definition of requirements, the of possible methods for a solution, the selection of the most appropriate methods and tools as well as their application - D2.11 Be able to evaluate the quality of information systems and to identify critical aspects. - D2.19 Be able to apply the own knowledge in different working contexts. Ability to make judgments - D3.1 Be able to collect and interpret useful data and to judge information systems and their applicability. - D3.2 Be able to work autonomously according to the own level of knowledge and understanding. - D3.3 Be able to take the responsibility for development of projects or IT consulting.

	Communication skills - D4.1 Be able to use one of the three languages English, Italian and German, and be able to use technical terms and communication appropriately. - D4.3 Be able to negotiate with a customer for the definition of the pre-requisites and features of information systems. - D4.4 Be able to structure and write technical documentation. - D4.5 Be able to work in teams for the realization of IT systems. Learning skills - D5.1 Have developed learning capabilities to pursue further studies with a high degree of autonomy. - D5.2 Have acquired learning capabilities that enable to carry out project activities in companies, public institutions or in distributed development communities. - D5.3 Be able to follow the fast technological evolution and to learn cutting edge IT technologies and innovative aspects of last generation information systems.
Obiettivi formativi specifici e risultati di apprendimento attesi (ulteriori info.)	
Modalità di esame	The assessment consists of a written exam and a group project. The written exam includes verification questions, and the project requires the submission of a written report. A positive evaluation of the project is valid for all three regular exam sessions. Projects must be submitted before the final exam at the end of the semester; otherwise, the exam cannot be registered.
Criteri di valutazione	The final grade is composed of 60% from the written exam and 40% from exercises or project work. Both the written exam and the project will be assessed based on the clarity of answers, mastery of language, critical thinking skills, ability to summarize, evaluate, and establish connections between topics, and technical competence. The project will also be evaluated on the ability to work effectively in a team, creativity, and development skills.
Bibliografia obbligatoria	Lecture notes
Bibliografia facoltativa	Ian Sommerville. Software Engineering. Pearson, 10th edition, 2016. ISBN 978-1-292-09613-1. URL:

	https://www.pearson.com/us/higher-education/program/PGM35255.html.
Altre informazioni	- ArgoUML (https://argouml-tigris-org.github.io/tigris/argouml/) - Papyrus (https://eclipse.dev/papyrus/) - StarUML (https://staruml.io) - UMLet (https://www.umlet.com)
Obiettivi di Sviluppo Sostenibile (SDGs)	Istruzione di qualità