# Syllabus

## *Course Description*

| | |
|---|---|
| **Course Title** | Software Engineering |
| **Course Code** | 76269 |
| **Course Title Additional** | |
| **Scientific-Disciplinary Sector** | INFO-01/A |
| **Language** | German |
| **Degree Course** | Bachelor in Computer Science |
| **Other Degree Courses (Loaned)** | |
| **Lecturers** | Prof. Dr. Claus Pahl,<br>Claus.Pahl@unibz.it<br>https://www.unibz.it/en/faculties/engineering/academic-staff/person/36376 |
| **Teaching Assistant** | |
| **Semester** | First semester |
| **Course Year/s** | 2 |
| **CP** | 6 |
| **Teaching Hours** | 40 |
| **Lab Hours** | 20 |
| **Individual Study Hours** | 90 |
| **Planned Office Hours** | |
| **Contents Summary** | 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. |
| **Course Topics** | - Software life-cycle: principles and methodologies<br>- Software processes and software project management<br>- Requirements engineering: elicitation and modeling<br>- System modeling and construction: UML, design patterns<br>- Software testing and management: principles and techniques<br>- Recent software engineering topics: DevOps, Cloud, SE and AI |
| **Keywords** | Software processes, requirements engineering, software design, |

| | |
|---|---|
| | software modeling, testing and maintenance. |
| **Recommended Prerequisites** | Students should have completed the following courses: Introduction to Programming, and Programming Project. |
| **Propaedeutic Courses** | |
| **Teaching Format** | The course includes frontal lectures, exercises, and projects. |
| **Mandatory Attendance** | 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. |
| **Specific Educational Objectives and Learning Outcomes** | Knowledge and Understanding<br>- D1.8 To have a thorough knowledge of the main fundamentals techniques and methods of software design, development and maintenance<br><br>Applying knowledge and understanding<br>- D2.5 Be able to apply the own knowledge to the , design, development and testing of information systems which satisfy given requirements<br>- 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<br>- D2.11 Be able to evaluate the quality of information systems and to identify critical aspects.<br>- D2.19 Be able to apply the own knowledge in different working contexts.<br><br>Ability to make judgments<br>- D3.1 Be able to collect and interpret useful data and to judge information systems and their applicability.<br>- D3.2 Be able to work autonomously according to the own level of knowledge and understanding.<br>- D3.3 Be able to take the responsibility for development of projects or IT consulting.<br><br>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.<br>- D4.3 Be able to negotiate with a customer for the definition of the pre-requisites and features of information systems.<br>- D4.4 Be able to structure and write technical documentation.<br>- D4.5 Be able to work in teams for the realization of IT systems.<br><br>Learning skills<br>- D5.1 Have developed learning capabilities to pursue further studies with a high degree of autonomy.<br>- D5.2 Have acquired learning capabilities that enable to carry out project activities in companies, public institutions or in distributed development communities.<br>- 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. |
| **Specific Educational Objectives and Learning Outcomes (additional info.)** | |
| **Assessment** | 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. |
| **Evaluation Criteria** | 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. |
| **Required Readings** | Lecture notes |
| **Supplementary Readings** | 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. |
| **Further Information** | - ArgoUML (https://argouml-tigris-org.github.io/tigris/argouml/)<br>- Papyrus (https://eclipse.dev/papyrus/)<br>- StarUML (https://staruml.io)<br>- UMLet (https://www.umlet.com) |
| **Sustainable Development Goals (SDGs)** | Quality education |