

## SYLLABUS COURSE DESCRIPTION

<b>COURSE TITLE</b>	<b>Software Engineering</b>
<b>COURSE CODE</b>	76215
<b>SCIENTIFIC SECTOR</b>	INF/01
<b>DEGREE</b>	Bachelor in Computer Science
<b>SEMESTER</b>	2nd Semester
<b>YEAR</b>	2nd year
<b>CREDITS</b>	6
<b>TOTAL LECTURING HOURS</b>	40
<b>TOTAL LAB HOURS</b>	20
<b>PREREQUISITES</b>	Students should have done the following courses: Computer Programming, Programming Project
<b>COURSE PAGE</b>	ole.unibz.it
<b>SPECIFIC EDUCATIONAL OBJECTIVES</b>	Type of course: "caratterizzanti" Scientific area: "discipline informatiche"  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.
<b>LECTURER</b>	<a href="#">Claus Pahl</a>
<b>SCIENTIFIC SECTOR OF THE LECTURER</b>	INF/01
<b>TEACHING LANGUAGE</b>	English
<b>OFFICE HOURS</b>	After the lecture/lab times or by appointment (email). Faculty of CS, Piazza Domenicani 3, Office 1.11
<b>TEACHING ASSISTANT</b>	Claus Phal
<b>OFFICE HOURS</b>	TBA

<b>LIST OF TOPICS COVERED</b>	<ul style="list-style-type: none"> <li>• Software life-cycle: principles and methodologies</li> <li>• Software processes and software project management</li> <li>• Requirements engineering: elicitation and modeling</li> <li>• System modeling and construction: UML, design patterns</li> <li>• Software testing: principles and techniques</li> <li>• Software management and evolution</li> </ul>
<b>TEACHING FORMAT</b>	Frontal lectures, exercises, projects.
<b>LEARNING OUTCOMES</b>	<p><b>Knowledge and understanding</b></p> <ul style="list-style-type: none"> <li>• Know in detail principles, techniques and methods of planning, designing, developing and maintaining software;</li> </ul> <p><b>Applying knowledge and understanding</b></p> <ul style="list-style-type: none"> <li>• Be able to apply the own knowledge to the analysis, design, development and testing of information systems which satisfy given requirements;</li> <li>• be able to solve typical problems in computer science, such as the definition of requirements, the analysis of possible methods for a solution, the selection of methods and tools as well as their application;</li> <li>• be able to evaluate the quality of information systems and to identify critical aspects;</li> <li>• be able to apply the own knowledge in different working contexts;</li> </ul> <p><b>Making judgments</b></p> <ul style="list-style-type: none"> <li>• be able to take the responsibility for software development projects</li> </ul> <p><b>Communication skills</b></p> <ul style="list-style-type: none"> <li>• be able to explain a project activity or a scientific study, also to non-experts</li> <li>• be able to work in teams to implement software systems</li> </ul> <p><b>Ability to learn</b></p> <ul style="list-style-type: none"> <li>• have acquired learning capabilities that enable them to carry out project activities in companies, public institutions or in distributed development communities</li> <li>• be able to learn cutting edge IT technologies and their strengths and limitations</li> </ul>
<b>ASSESSMENT</b>	<p>Written and project work: written exam with verification questions and written project report done in groups.</p> <p>In case of a positive mark the project will count for all 3 regular exam sessions.</p> <p>Projects have to be submitted BEFORE the final exam at the end of the semester, otherwise the exam cannot be registered.</p>
<b>ASSESSMENT LANGUAGE</b>	English
<b>EVALUATION CRITERIA AND</b>	Weighting of parts: <ul style="list-style-type: none"> <li>• 70% written exam</li> </ul>

<b>CRITERIA FOR AWARDING MARKS</b>	<ul style="list-style-type: none"> <li>• 30% exercises/project.</li> </ul> <p>Criteria:          Relevant for assessment of project and exam:</p> <ul style="list-style-type: none"> <li>• clarity of answers,</li> <li>• mastery of language,</li> <li>• skills in critical thinking</li> <li>• ability to summarize, evaluate, and establish relationships between topics,</li> <li>• technical competence</li> </ul> <p>Relevant for project assessment:</p> <ul style="list-style-type: none"> <li>• ability to work in a team,</li> <li>• creativity,</li> <li>• development skills</li> </ul>
<b>REQUIRED READINGS</b>	The course will be based on lecture notes
<b>SUPPLEMENTARY READINGS</b>	I. Sommerville. Software Engineering. Addison Wesley.
<b>SOFTWARE USED</b>	Software Modelling (e.g. Argo UML, Papyrus, StarUML, UMLet)