

<b>Study programme(s):</b> Information Technologies			
<b>Level:</b> Bachelor			
<b>Course title:</b> Software Engineering			
<b>Lecturer:</b> Zoran D. Budimac			
<b>Status:</b> obligatory			
<b>ECTS:</b> 8			
<b>Requirements:</b> None			
<b>Learning objectives</b> Overview of elementary and advanced phases and techniques of software development. Preparation of students for teamwork in characteristic phases of software development: requirements, analysis, design, implementation, elements of management and quality control.			
<b>Learning outcomes</b> <i>Minimal:</i> By the end of the course, it is expected that student presents knowledge and ability of its application, and to be able to work as a team member on the development and delivery of software products of high quality. <i>Desirable:</i> By the end of the course, it is expected that student presents good knowledge, ability for critical analysis and application of knowledge from the field, ability to work both individually and as a team member on the development and delivery of high quality software products, and ability to analyze their quality level.			
<b>Syllabus</b> <i>Theoretical instruction</i> Basic notions and definitions. Software quality criteria. Models of software development process and basic concepts of the development description. Possible views on the software development process: functional, data oriented, rule oriented, state oriented, scenario based. Structure and object-oriented analysis and design. Formal specification. Principles and methods of implementation. Reverse engineering. Standardization of a software development process. <i>Practical instruction</i> Analysis and practical improvement of requirements specification. Training in methods of software cost estimation. Training in object-oriented analysis. Training in description of software product by methods of formal specification. Practical work on system and functional testing. Principles of software metrics and practicing of methods of software quality measurement.			
<b>Literature</b> <i>Recommended:</i> Zoran Budimac, Mirjana Ivanovic, Zoran Putnik: <i>Advanced Topics in Software Engineering</i> , University of Novi Sad, Faculty of Science, Department of Mathematics and informatics, Novi Sad, 2007. <i>Alternative:</i> Ian Sommerville: <i>Software Engineering</i> , 7th Edition, Pearson Education Limited, 2004.			
<b>Weekly teaching load</b>			Other:
Lectures: 4	Exercises: 2	Other forms of teaching:	
<b>Teaching methodology</b> During the lectures, classic methods of teaching are used assuming usage of video-beam and slides. All of the presentations are also available on a web-site of the Department as a static PDF files, but also as dynamic electronic lessons. At theoretical exercises, applicable methods for individual phases of software development are presented and explained in details. Through practical exercises, presented methods are practiced by students using teamwork. Assignment solutions are, within a given time-frame, submitted either in written or electronic form. During exercises, students knowledge is tested with four tests, covering the material presented at lectures.			
<b>Grading method (maximal number of points 100)</b>			
<b>Pre-exam obligations</b>	<b>points</b>	<b>Final exam</b>	<b>points</b>
Practical assignments	<b>60</b>	Oral exam	<b>40</b>