

Study Programme: Applied Mathematics – Data Science		
Course Unit Title: Software engineering		
Course Unit Code: MDS28		
Name of Lecturer(s): Zoran D. Budimac		
Type and Level of Studies: Master Academic Degree		
Course Status (compulsory/elective): elective		
Semester (winter/summer): Winter		
Language of instruction: English		
Mode of course unit delivery (face-to-face/distance learning): Face-to-face		
Number of ECTS Allocated: 5		
Prerequisites: None		
Course Aims: 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.		
Learning Outcomes: <i>Minimal:</i> Students should be able to apply the obtained knowledge, and be able to work as a team member on the development and delivery of software products of high quality. <i>Optimal:</i> Students should have good knowledge, ability for critical analysis and application of knowledge in the field, ability to work both individually and as a team member on the development and delivery of high quality software products, as well as the ability to analyze their quality level.		
Syllabus: <i>Theory</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>Practice</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.		
Required Reading: 1. Eric J. Braude, Michael E. Bernstein, Software Engineering: Modern Approaches, John Wiley and sons, 2010 2. R. Pressman: Software Engineering, A Practitioner's Approach, 7th edition, McGraw-Hill, 2009 3. I. Sommerville: Software Engineering, 9th Edition, Addison-Wesley, 2010 4. G. Booch, I.Jacobson, J. Rumbaugh: The Unified Modeling Language User Guide. Addison-Wesley, 2005		
Weekly Contact Hours:	Lectures: 2	Practical work: 2

Teaching Methods:

Classical methodology is applied in lectures including the use of the video-beam. During exercises, case studies are analyzed in-depth. Some aspects and principles are practically covered by software tools. Furthermore, students study some of the covered topics and report on their findings in written papers in an individual and more thorough manner.

Knowledge Assessment (maximum of 100 points):

Pre-exam obligations	points	Final exam	points
Active class participation		written exam	
Practical work	20	oral exam	40
Preliminary exam(s)	40	
Seminar(s)			