

1.	Course title	Advanced software engineering		
2.	Course code	SI-3-01		
3.	Study program	Master Studies in Computer Science and Engineering - Software engineering		
4.	Unit offering the course	FCSE		
5.	Undergraduate/master/PhD	Master		
6.	Year/semester 1(2)/winter/compulsory	7. ECTS: 6		
8.	Teacher(s)	Assoc. prof. d-r Dejan Gjorgjevikj, assist. prof. d-r Ivan Chorbev		
9.	Course prerequisites	None		
10.	Goals (competences): The goal of the course aims for the students to acquire advanced knowledge in software engineering for developing big software projects. After the completion of the course, the candidates will be able to: specify, model, implement and test software systems, to define, plan and execute a software project while working in a team, to cooperate with other teams while working on joint projects, to extract, analyse and document experiences out of development projects, to constructively apply the general principles of software engineering in development of software intensive systems, to be fully familiarised with the general principles and techniques for quality control of software systems (security, reliability, robustness and features specific to various kinds of software systems).			
11.	Course content: Software lifecycle, methodologies and techniques. Managing the software development process. Managing requirements. Methods for systematic and detailed design. Managing software projects, managing resources. Quality control through revisions and testing. Delivery strategies. Management of configurations and changes. Agile software development methods. Formal methods for software development. Development of web based software, distributed software and systems in the cloud. Software for service oriented systems. Software for mobile devices and embedded systems. Software as a service.			
12.	Teaching methods: Lectures supported by slide presentations, interactive lectures, trainings (using lab equipment and software packages), team work, case studies, invited guests and lectures, individual practical assignments presentations, seminar paper, e-learning (forums, consultations).			
13.	Total available time	6 ECTS x 30 hours = 180 hours		
14.	Distribution of the available time	60 + 0 + 120 = 180 hours		
15.	Teaching activities	15.1.	Lectures	60 hours
		15.2.	Training (labs, problem solving), seminar and team work	0 hours
16.	Other activities	16.1.	Project work	45 hours
		16.2.	Self study	45 hours
		16.3.	Home work	30 hours

17.	Grading					
	17.1.	Tests			45 points	
	17.2.	Seminar work/project (written or oral presentation)			45 points	
	17.3.	Active participation			10 points	
18.	Grading criteria		to 59 points		5 (five) (F)	
			from 60 to 68 points		6 (six) (E)	
			from 69 to 76 points		7 (seven) (D)	
			from 77 to 84 points		8 (eight) (C)	
			from 85 to 92 points		9 (nine) (B)	
from 93 to 100 points		10 (ten) (A)				
19.	Final exam prerequisites		Successfully completed activities 15.1 and 15.2			
20.	Course language		Macedonian and English			
21.	Quality assurance methods		Internal evaluation and student questionnaires			
22.	Literature					
	22.1.	Compulsory				
		No.	Authors	Title	Publisher	Year
		1.	Ian Sommerville	Software Engineering, 9 th Ed.	Addison Wesley	2010
		2.	Roger S. Pressman	Software Engineering: A Practitioner's Approach 7 th Ed.	McGraw-Hill Science	2009
	3.					
	22.2.	Additional				
		No.	Authors	Title	Publisher	Year
		1.	Capers Jones	Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies	McGraw-Hill Osborne Media	2009
		2.	Frederick P. Brooks	The Mythical Man-Month: Essays on Software Engineering	Addison-Wesley Professional	1995
3.		Alain Abran, James W. Moore, Pierre Bourque, Robert Dupuis	Guide to the Software Engineering Body of Knowledge (SWEBOK): 2004 Version	IEEE Computer Society Press	2005	