I052	**Software Engineering**	L	S	E	ECTS
		2	0	2	6

Course objectives. Course objectives are to help students to synthetize, systematize and deepen the knowledge they already acquired on previous Computer Science courses. In other words, the main objective of this course is to help students to link already acquired knowledge into a whole, in order to get a global image of software development process, as well as to provide them with the skills necessary for working on large-scales software projects.

Course prerequisites. Introduction to Computer Science. Object Oriented Programming.

Syllabus.

1. *Introduction*. Basic notions in software engineering. Models for software engineering. Classic and agile methods in software development. Software project management.
2. *Requirements*. Requirement analysis. Detection of requirement. System modelling.
3. *Modelling and implementation*. Architecture modelling. Object-oriented approach in modelling. Modelling of distributed systems. Modelling and implementation of Modelling and implementation of embedded systems.
4. *Verification and validation*. Static verification. Software testing.
5. *Maintenance and evolution*. Configuration management. Legacy systems and its evolution.
6. *Software reuse*. Traditional forms of software reuse. Component based software development. Application development with the help of web services.

EXPECTED LEARNING OUTCOMES

No.	LEARNING OUTCOMES
1.	To demonstrate the knowledge and understanding which can serve as the foundation for developing and applying original ideas.
2.	To apply knowledge, understanding and skills in a broad variety of problems in the field of software engineering.
3. 4.	To integrate new knowledge to solve problems in software development. To be able to present conclusions and findings to experts and laymen based on the knowledge and experience.
5.	To apply the acquired skills onto further education in this field.

COUPLING OF THE EXPECTED LEARNING OUTCOMES, TEACHING PROCESS ORGANIZATION AND THE EVALUATION OF THE TEACHING OUTCOMES

TEACHING PROCESS ORGANIZATION	ECTS	EXPECTED LEARNING OUTCOMES **	STUDENT ACTIVITY *	EVALUATION METHOD	SCORE	
					min	max

Lecture attendance	1	1-5	Class atendance, discussion, solving the problems individually and in a team	Lists with signatures, observing the activity during the lectures	0	4
Homework	1	1-4	Solving the problems individually	Grading	12	20
Repeated exams	2	1-4	Preparation for the written exam	Grading	19	38
Final exam	2	1-4	Revising	Oral exam	19	38
TOTAL	6				50	100

Teaching methods and student assessment. Lectures will contain many examples with in-depth explanations. Lectures and exercises are obligatory. Exercises will be held in computer laboratories. The final exam will be held after the completion of lectures and exercises and it will contain theoretical and practical tasks each student will have to complete independently. Students can influence their final grade by successful completion of homework that will be given to students on a regular basis.

Can the course be taught in English: Yes

Basic literature:

1. F. Tsui, O. Karam, B. Bernal: Essentials of Software Engineering, 3rd Edition. Jones & Bartlett Publishers, Sudbury MA, 2013.

Recommended literature:

1. R. Manger, "Softversko inženjerstvo", Element, Zagreb, 2016
2. J.Arlow, I.Neustadt: UML 2 and the Unified Process, 2nd Edition. Addison-Wesley, Upper Saddle River, NJ, 2005.
3. E.J. Braude, M.E. Bernstein: Software Engineering – Modern Approaches. John Wiley and Sons, New York, 2010.
4. S.L. Pfleeger, J.M. Atlee: Software Engineering - Theory and Practice, 4th Edition. Pearson Education Inc, Boston MA, USA, 2013.
5. I.Sommerville: Software Engineering, 10th Edition. Pearson Education Inc, Boston MA, USA, 2016.
   H. Van Vliet: Software Engineering - Priniples and Practice, 3rd Edition. John Wiley and Sons, Chichester UK, 2008.