Study Program: Informatics

Type and level of studies: Bachelor studies

Course name: Software Engineering

Lecturer: Савић С. Милан

Status: Compulsory

ECTS: 6

Attendance Prerequisites: Basic, high-school level informatics knowledge.

Course aims

Acquisition of basic knowledge in the field of software engineering, as well as learning to use the tools for software analysis, design and modeling based on the established requirements of users.

Course outcome

The student's are able to design, model, develop and deliver software products and analyze their quality. They have learned the basics which will allow them to perform better in programming, architecture and software development courses.

Course content

Theoretical part

Software process models. Software project management. The principles of software engineering. Requirements collection and analysis. System modeling. Use of prototypes. Formal specification. Design, construction, testing and maintenance. Project forms. The applicability and significance of project forms. Learning the basic patterns.
User-defined features and user-oriented design. Client - server. Introduction to client - server architecture as well as client - server systems with an emphasis on class diagrams and state diagrams. Software Design. General design principles: decomposition, coupling, cohesion, reuse, portability, flexibility. Structural and object-oriented analysis and design. Formal specifications. Design evaluation. Introduction to Software Testing. Software quality criteria. Implementation principles and methods. Design implementation via appropriate data structures, environments, and APIs. Analysing, designing and programming simpler applications. Application of technology in implementing user interface. Reverse engineering. Standardization of software development process. Configuration management. Software reuse.

Practical part

Practicing the methods of analyzing and improving the specified requirements, as well as the methods of estimating the cost of software development of object-oriented analysis and the description of the software product through methods of formal specification. Systematic and functional testing in practice, practicing software quality measurement methods.Other forms of teaching, student research work.

Literature:

1. S. L. Pfleeger, J. M. Atlee, Софтверско инжењерство, теорија и пракса, ЦЕТ, 2006.
2. Ian Sommerville: *Software Engineering*, 7th Edition, Pearson Education Limited, 2004.
3. S.R.Schach: Object-Oriented and Classical Software Engineering, Seventh Edition. McGraw-Hill, New York, 2006.
4. R.S.Pressman: Software Engineering - A Practitioner's Approach, Sixth Edition. McGraw-Hill, New York, 2005.

Number of active classes				Other classes
Lectures: 3	Practical classes: 3	Other forms of teaching:	Students' research work	

Teaching methods

Lectures are in accordance with the topic in *course content*, computer practice and independent student research work.

Assessment (maximum 100 points)			
Course assignments	**points**	Final exam	**Points**
activity during lectures	**10**	written exam	**20**
practical classes	**10**	oral exam	**30**
term test(s)	**20**		
seminar(s)	**10**		
Total	**50**		**50**