



COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Software Engineering II	

Lecturer(s)	Department where the course unit is delivered
Coordinator: Karolis Petrauskas Other lecturers: Saulius Ragaišis	Department of Software Engineering Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the course unit
1 st (BA)	Compulsory

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	4 th semester	Lithuanian

Prerequisites
Prerequisites: Software Engineering I

Number of credits allocated	Student's workload	Contact hours	Individual work
10	270	84	186

Purpose of the course unit: programme competences to be developed		
Purpose of the course unit – to introduce students with software engineering and its application in software maintenance, apply defined methods in practice, develop skills in testing and maintaining software. Also to introduce students with basics of project management, quality assurance, software engineering process, and enterprise architecture.		
Generic competences: <ul style="list-style-type: none"> • Communication and collaboration (<i>GK1</i>). 		
Specific competences: <ul style="list-style-type: none"> • Knowledge and skills of underlying conceptual basis (<i>SK4</i>). • Software development knowledge and skills (<i>SK5</i>). • Technological and methodological knowledge and skills, professional competence (<i>SK6</i>). 		
Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
Will be able to chose and apply different software design methods. Apply knowledge on software engineering to maintain software, to present and defend proposed solutions. Select life-cycle model for software and to design systems with respect to maintenance requirements. Define software maintenance plans and plans for software development projects, identify risks. Work according to defined methods for software development and maintenance. Define enterprise architecture and develop road-maps based on it.	Problem oriented teaching, case analysis, group discussion, individual reading.	Laboratory assignments, examination in written form.

Course content: breakdown of the topics	Contact hours						Individual work: time and assignments		
	Lectures	Tutorials	Seminars	Practice	Laboratory work (LW)	Practice	Contact hours	Individual work	Assignments
Overview of the Iconix process: core principles, initial requirements, requirements definition, analysis, conceptual modelling, design, coding, traceability and testing. Structure of the course, requirements for the student evaluation.	9				12		21	42	Individual reading, preparation of 1st laboratory assignment.
Software construction: main objectives and issues, management, techniques used, technologies and tools.	3				10		13	32	Individual reading, preparation of 2nd laboratory assignment.
Software testing: terminology, main issues, test levels, objectives, techniques, measurements and tools. Testing process and related activities.	6				3		9	14	Individual reading, preparation of 3rd laboratory assignment.
Software maintenance: strategies for software maintenance, main issues, change management, impact analysis and scope estimation. Software release management and architecture management.	9				7		16	24	
Configuration management: main concepts, roles and activities, relation to the software development and maintenance.	3						3	6	Individual reading.
Introduction to project management: main concepts and roles, its role in the context of software engineering. Project management process groups: initiation, planning, executing, monitoring and control, closing.	6						6	12	Individual reading.
Software engineering process: definition of the software engineering process, assessment and improvement models, software life-cycle and its models.	4						4	12	Individual reading.
Software quality assurance: main concepts, quality characteristics. Software quality management process.	2						2	8	Individual reading.
Enterprise architecture: main concepts, context and domain. Overview of enterprise architecture frameworks: Zachman framework, TOGAF.	6						6	12	Individual reading.
Exam in written form		2					4	24	2 hours for tutorial, 24 hours for preparation, and 2 hours for the exam.
Total	48	2			32		84	186	

Assessment strategy	Weight %	Deadline	Assessment criteria
1 st laboratory assignment	15%	Week 6	Students should organize small teams, which will choose one of the projects previously developed by other team in the course of Software engineering I or other course. Define new requirements for the selected system. Results of the assignment should be provided in the written form and defended during laboratory assignments class. The work is evaluated in the ten points system. A penalty for each delayed week is 1 point. A progress of the assignment must be discussed with the laboratory assignments class teacher on each lecture.
2 nd laboratory assignment	10%	Week 10	Students should perform analysis of requirements, described in the first laboratory assignment, and define architecture for the solution. The work should be done in the same teams. Results of the assignment should be provided in the written form and defended during laboratory assignments class. The work is evaluated in the ten points system. A penalty for each delayed week is 1 point. A progress of the assignment must be discussed with the laboratory assignments class teacher on each lecture.
3 rd laboratory assignment	10%	End of the semester	Students should specify in more details system design that was presented in the second laboratory assignment, implement system prototype, prepare test plan and test scenarios. The work should be done in the same teams. Results of the assignment should be provided in the written form and defended during laboratory assignments class. The work is evaluated in the ten points system. A penalty for each delayed week is 1 point. A progress of the assignment must be discussed with the laboratory assignments class teacher on each lecture.
Practical assignments	10%	During the semester	Each student should solve 4 small practical problems during the semester. The problems are formulated to check skills on a particular topic. Each assignment is evaluated in the ten points system.
Exam in written form	55%	During exam session	For the right to take the exam a student must submit all the laboratory assignments and each of them must be evaluated not less than 5 in the ten points system.

Author	Publishing year	Title	Number or volume	Publisher or URL
Required reading				
R.S. Pressman	2004	Software Engineering: A Practitioner's approach		McGraw-Hill
P. Bourque, R. E. Fairley	2014	Guide to the Software Engineering Body of Knowledge Version 3.0		IEEE, http://www.computer.org/portal/web/swebok .
D. Rosenberg, M. Stephens	2007	Use Case Driven Object Modeling with UML: Theory and Practice		Apress
Recommended reading				
A. Čaplinskas	1996	Programų sistemų inžinerijos pagrindai, I dalis		MII, Vilnius.
A. Čaplinskas	1998	Programų sistemų inžinerijos pagrindai, II dalis		MII, Vilnius.