COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Software Engineering I	

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
1st (BA)	Compulsory

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	3rd semester	Lithuanian

Prerequisites
Prerequisites: Procedural programming, Object-oriented programming, Discrete mathematics.

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 the software engineering, its application in development of software systems, including domain modelling, requirements analysis and software design.

Generic competences:

- Communication and collaboration *(GK1)*.
- Social responsibility *(GK3)*.

Specific competences:

- Knowledge and skills of underlying conceptual basis *(SK4)*.
- Software development knowledge and skills *(SK5)*.
- Technological and methodological knowledge and skills, professional competence *(SK6)*.

Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
Know main concepts, application domains and a scope of the software engineering.	Problem oriented teaching, case analysis, group discussion, individual reading.	Laboratory assignments, examination in written form.
Apply knowledge of software engineering to development of software systems, present and defend proposed solutions.		
Make analysis of a business domain; use BPMN for modelling business processes.		
Assess a solution in technology, economic and social aspects.		
Define functional and non-functional requirements for a software system and assess quality of the requirements.		
Design software systems according to defined requirements, select appropriate architectural solution, use UML for modelling and specifying the system.		

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 software engineering, its scope and definition. Structure of the course, requirements for the student evaluation.	3						**3**	**18**	Individual reading.
Domain modelling: information model, agents, use-cases and use case scenarios.	6				4		**10**	**12**	
Object-oriented paradigm and UML: definition of the object-oriented paradigm, structure of the UML, UML diagrams and models. Model abstraction levels, UML 4+1 model, UML tools.	9				3		**12**	**18**	
Software design: structural and behavioural models of a system, decomposition, views, viewpoints and abstraction levels of a system design. Applying UML for design of software systems.	6				5		**11**	**24**	
Software requirements: functional and non-functional requirements, types of non-functional requirements, methods for requirement gathering, requirement quality characteristics and annotations. Requirements management and specification.	6				8		**14**	**38**	Individual reading, preparation of 2nd laboratory assignment.
Business analysis: internal and external analysis of a business process, black-box and white-box principles, measurements, SWOT analysis, five Porter's forces, feasibility analysis.	9				6		**15**	**25**	
Main issues in software design: concurrency, event handling, distribution of components, error handling, interaction, data persistence, reuse. Typical tactics for solving main issues in software design.	6				3		**9**	**10**	Individual reading, preparation of 3rd laboratory assignment.
Software architectural styles: object-oriented architecture, event-driven architecture, service oriented architecture. Layered models, data-flow and control-flow oriented designs. Distributed systems, the CAP theorem.	3				3		**6**	**17**	
Exam in written form		2					**4**	**24**	2 hours for tutorial, 24 hours for preparation, 2 hours for the exam.
Total	**48**	**2**			**32**		**84**	**186**	

Assessment strategy	Weight %	Deadline	Assessment criteria
1st laboratory assignment	10%	Week 6	Students should organize in small teams, which will choose a domain and a problem related to it, design an architecture for a system solving the chosen problem and implement its prototype. 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.

2nd laboratory assignment	10%	Week 10	Students should specify requirements for the software designed during the first laboratory assignment, update the architecture and the prototype according to the requirements. 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.
3rd laboratory assignment	15%	End of the semester	Students should define business needs and perform an analysis of the business domain, then update the requirements, the architecture and the prototype according to the results of the business analysis. 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.
Object Management Group	2015	OMG Unified Modeling Language, Version 2.5		http://www.omg.org/spec/UML/2.5
Recommended reading				
A. Čaplinskas	1996	Programų sistemų inžinerijos pagrindai, I dalis		MII, Vilnius.
A. Čaplinskas	1998	Programų sistemų inžinerijos pagrindai, II dalis		MII, Vilnius.