COURSE UNIT (MODULE) DESCRIPTION

Course unit (module) title	Code
Software engineering 1	

Academic staff	Core academic unit(s)
Coordinating: dr. Vaidas Giedrimas **Other:**	VU SA

Study cycle	Type of the course unit
First	Mandatory

Mode of delivery	Semester or period when it is delivered	Language of instruction
Auditorium	3rd semester	Lithuanian, English

Requisites	
Prerequisites:	**Co-requisites (if relevant):**

Number of ECTS credits allocated	Student's workload (total)	Contact hours	Individual work
5	133	56	77

Purpose of the course unit
To acquaint students with the basic concepts and principles of software systems engineering. To provide the skills needed to extract, formulate, specify, analyze, evaluate and manage requirements in the subsequent stages of software systems engineering. Cultivated competences: • BK1 Application of knowledge • BK2 Social skills • BK3 Personal abilities • DK1 Application of knowledge of program systems • DK2 Abilities to conduct program system research • DK3 PS Special Abilities

Learning outcomes of the course unit	Teaching and learning methods	Assessment methods
Will know the basic principles of software engineering and will be able to apply them in software development projects.	Lecture, case study, self-study of literature	Exam
Will know software life cycle and will be able to choose the most suitable for particular development project.	Lecture, case study, self-study of literature	Exam
Will be able to explain requirements' types and levels, requirements engineering process models.	Interactive lecture, self-study of literature	Exam
Will be able to select appropriate data sources for the requirements engineering process and aquire information from them.	Laboratory works, Group work	Defense of laboratory work, Presentation of group work results, Exam

Will be able to work in a group according to the requirements engineering process, communicate with stakeholders.	Laboratory works, Group work	Defense of laboratory work, Presentation of group work results, Exam
Will be able to prepare software specification suitable for use in subsequent stages of software engineering.	Laboratory works, Group work	Defense of laboratory work, presentation of group work results

Content	**Contact hours**							**Individual work: time and assignments**	
	Lectures	Tutorials	Seminars	Workshops	Laboratory work	Internship	**Contact hours, total**	**Individual work**	**Tasks for individual work**
Concept and principles of software systems engineering. PS life cycles	4				2		**6**	**12**	Defense of laboratory work, Presentation of group work results, Exam
PS Complexity Management. The principle of separation of contents. Grip and cohesiveness.	4				2		**6**	**12**	
Methods of subject area analysis and conceptual modeling.	4				2		**6**	**8**	
Introduction to UML. Task and activity charts.	4				6		**10**	**8**	
Types and levels of requirements for the program system. Sources of information. Requirements extraction process.	2				6		**8**	**8**	
Formulation of requirements. Structure of the requirements specification. Standards.	2				6		**8**	**12**	
Requirements analysis.	2				4		**6**	**9**	
Requirements change management.	2				4		**6**	**8**	
Total	**24**				**32**		**56**	**77**	

Assessment strategy	**Weight %**	**Deadline**	**Assessment criteria**
Defense of laboratory work	20%	Every second week	Laboratory works and their defense are evaluated.
Presentation of group work results	30%	3 times per semester	It is given in the first week of studies and is carried out in stages.
Exam	50%	During the exam session	The exam test in the Moodle environment consists of 20 open and closed type questions, each evaluated by half a point. The exam evaluation is equal to the sum of the collected points.

Author (-s)	**Publishing year**	**Title**	**Issue of a periodical or volume of a publication**	**Publishing house or web link**
Required reading				
PA Laplante	2017	Requirements Engineering for Software and Systems		Auerbach Publications
D. Farley	2021	Modern Software Engineering		Addison-Wesley
M. Richards, N. Ford	2020	Fundamentals of Software Architecture: an		O'Reilly

		Engineering Approach		
Recommended reading				
LA Maciaszek, BL Liong	2005	Practical Software Engineering: A Case Study Approach		Addison-Wesley
I. Sommerville	2015	Software Engineering (10th Ed)		Addison-Wesley