

Subject code	Credits
INF3021	6

Title

PROGRAM INŽINERIJOS PAGRINDAI

Title in English

SOFTWARE ENGINEERING BASICS

Subject goal and annotation

The aim of this course . gain knowledge in software engineering and develop skills necessary for efficient design of software systems. In this course main software system life-cycle phases are presented: requirement analysis and specification, design and construction, testing, delivery to the user, use and maintenance. Also in this course are presented the modern practices and tools used in requirements analysis and system design. The students will learn how to model business domain with UML, apply use case method, document requirements, review requirement specifications, design the system and identify test cases. The theory and practices will be discussed in lectures, and the tools will be explored in labs.

Prerequisites

Undergraduate courses: logics, set theory, algebra, statistics

Relationship between the learning outcomes of the Programme and learning outcomes of the subject

Learning outcomes of the Programme	Learning outcomes of the subject	Criteria for measuring the achievement of learning outcomes
3.Knowledge of basic and advanced computer science and its application.	Knowledge on the discipline of software engineering	Student demonstrates the ability to analyse and design provided processes/systems
8. Perform interdisciplinary research and development in Internet systems area, apply results in practical applications.	An ability to design a system, component, or process to meet desired needs within realistic constraints Ability to understand the appropriateness of different software engineering methodologies for different circumstances and different type of software systems	Student demonstrates skills in systems/system components design. Student is able to choose appropriate software engineering method for different type of systems working in different circumstances.
7. Formalization and specification of real-world problems, and ability to describe them at an abstract level 10. Analysis, design and development of advanced Internet systems. 12. Analysis, design and development of diverse software systems.	Ability to understand the common structures of software development team and develop practical skills in solving small problems in teams Ability to understand the awareness of software engineer responsibilities Ability to plan a software engineering process to account for quality issues and functional / non-functional requirements; Ability to use basic modelling techniques for definition and description of the software systems behavior Ability to select a concepts and techniques for completion of a small-scale analysis and design project.	Students demonstrate the ability to develop in team, the creativity in choosing type of system for the analysis and design; show the ability to plan software engineering process. Student is able to use modelling techniques for software systems behavior definition and description and choose appropriate methods for system analysis and design.

15. Clear and convincing presentation of problems and solutions to experts and non-experts using ground knowledge, reasoning, relevant presentation tools and methods. 16. Project managing and collaborating in teams.	Improve team working skills by including general organization of the project, planning and time management, inter-group negotiation. Presenting results	Students demonstrate the ability to develop in team, creativity in choosing type of system for analysis and design. Students presentations of their work in software development team.
--	--	---

Subject content

	Lecture topics and contents	Hours
1.	Introduction to software engineering	3
2.	Software process	6
3.	Rational Unified Process	1
4.	Agile software development	3
5.	Requirements engineering	6
6.	System modelling	8
7.	Architectural design	6
8.	Design and implementation	4
9.	Software testing	4
10.	Software evolution	4
	Total	45

Practical work contents

Practical work assignments are focused on team working and on practice in using UML for software development process with MagicDraw.

Evaluation of study results

Final written exam (50%), mid-term written exam (17%), and assessments of laboratory (practical) work (33%).

Distribution of subject study hours

Lectures	45
Laboratory work	30
Individual studies (including studies in groups, preparation for the mid-term and final exams)	81
Total	156

Recommended literature

No	Authors of publication and title	Number of copies available		
		<i>in the Library of VMU</i>	<i>in specialized publication collections at VMU</i>	<i>in other libraries</i>
Basic materials				
1.	D. Budgen. Software Design. Harlow: Person Education; New York N.Y. : Addison-Wesley, 2003.	1		
2.	L. A. Maciaszek, B. L. Liang. Practical software engineering. Harlow: Pearson Addison Wesley, 2005.	1		
3.	R. S. Pressman. Software engineering: a practitioner's approach. Boston Mass. etc.: McGraw-Hill/Higher Education, 2010.	1		
	Ian Sommerville. An Introduction to Software Engineering. Open textbook.	http://www.merlot.org/merlot/index.htm		
Supplementary materials				
1.	D. Leffingwell, D. Widrig. Managing software requirements: a use case approach. Boston Mass. etc.: Addison-Wesley, 2003.	1		

Subject prepared and coordinated by

Dr.Auzra Mackut -Varoneckien