



Subject card

Subject name and code	Software Engineering, PG_00044135						
Field of study	Mathematics						
Date of commencement of studies	October 2024	Academic year of realisation of subject			2024/2025		
Education level	second-cycle studies	Subject group					
Mode of study	Full-time studies	Mode of delivery			at the university		
Year of study	1	Language of instruction			Polish		
Semester of study	2	ECTS credits			4.0		
Learning profile	general academic profile	Assessment form			assessment		
Conducting unit	Department of Differential Equations and Mathematical Applications -> Faculty of Applied Physics and Mathematics						
Name and surname of lecturer (lecturers)	Subject supervisor	dr hab. Paweł Pilarczyk					
	Teachers	dr hab. Paweł Pilarczyk					
Lesson types and methods of instruction	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	30.0	0.0	15.0	15.0	0.0	60
	E-learning hours included: 0.0						
Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours		Self-study	SUM	
	Number of study hours	60	5.0		35.0	100	
Subject objectives	Becoming familiar with basic methods of software engineering and acquiring the ability to apply these methods in practice. This concerns group work on a large scale IT project at all the stages of software production: beginning with requirements engineering, through requirements analysis, software design, implementation, testing, installation, to the stage of software maintenance. Also getting acquainted with basic issues regarding IT project management, such as quality management.						
Learning outcomes	Course outcome		Subject outcome		Method of verification		
Subject contents	Lecture: Introduction to software engineering. Ethics code. Software development processes. Requirements engineering. Structural and object-oriented methods. Using the UML in modelling. Testing software. IT project management, quality management. Software maintenance. Critical systems. Agile Manifesto and agile methods, including Scrum. Lab and project: Applying the software engineering methods to designing specific software, including preparation of UML diagrams and user stories. Practical development of software prototypes.						
Prerequisites and co-requisites	Basic ability to write computer programs, e.g. in Python, C, or HTML/JavaScript.						
Assessment methods and criteria	Subject passing criteria		Passing threshold		Percentage of the final grade		
	homework, project, presentations, etc.		60.0%		50.0%		
	quizzes (in writing, 10 minutes each)		60.0%		50.0%		
Recommended reading	Basic literature		Krzysztof Sacha, Inżynieria oprogramowania, PWN 2010. Mariusz Chrapko, Scrum. O zwinnym zarządzaniu projektami. Wydanie II rozszerzone, Helion 2014.				

	Supplementary literature	<p>Ian Sommerville, Software Engineering, Pearson, 10th edition, 2015.</p> <p>Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill; 8th edition, 2014.</p> <p>Russ Miles, Kim Hamilton, Learning UML 2.0: A Pragmatic Introduction to UML, O'Reilly and Associates, 2006.</p> <p>Kenneth S. Rubin. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Pearson Education, 2013.</p>
	eResources addresses	Adresy na platformie eNauczanie:
Example issues/ example questions/ tasks being completed	<p>Requirements engineering.</p> <p>Software development processes.</p> <p>Verification and validation.</p> <p>Agile methods and the Manifesto.</p> <p>Ethics code of a software engineer.</p> <p>The INVEST features of user stories.</p>	
Work placement	Not applicable	

Document generated electronically. Does not require a seal or signature.