



## Subject card

Subject name and code	Software engineering, PG_00051071						
Field of study	Technical Physics						
Date of commencement of studies	October 2021	Academic year of realisation of subject			2023/2024		
Education level	first-cycle studies	Subject group			Optional subject group Subject group related to scientific research in the field of study		
Mode of study	Full-time studies	Mode of delivery			at the university		
Year of study	3	Language of instruction			Polish		
Semester of study	6	ECTS credits			7.0		
Learning profile	general academic profile	Assessment form			assessment		
Conducting unit	Zakład Fizyki Teoretycznej i Informatyki Kwantowej -> Instytut Fizyki i Informatyki Stosowanej -> Faculty of Applied Physics and Mathematics						
Name and surname of lecturer (lecturers)	Subject supervisor	dr hab. inż. Marta Łabuda					
	Teachers	dr hab. inż. Marta Łabuda					
Lesson types and methods of instruction	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	30.0	0.0	0.0	45.0	0.0	75
	E-learning hours included: 0.0						
Additional information:							
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	75	10.0	90.0	175		
Subject objectives	Student knows, what is specific for software engineering Student knows several models of process of developing software Student knows rules of acquiring requirements Student knows, how to model the system using the UML language Student knows the fundamentals of software projects management Student knows basic architectures of computer systems.						
Learning outcomes	Course outcome	Subject outcome			Method of verification		
	K6_U02	The student carries out project tasks related to software engineering.			[SU1] Assessment of task fulfilment [SU2] Assessment of ability to analyse information		
	K6_U03	The student knows different programming languages, tools and modern technologies supporting design and architecture of the IT systems			[SU1] Assessment of task fulfilment		
	K6_W05	The student knows what is software engineering. The student knows different models of software development. The student knows the rules of collecting and documenting requirements for the IT systems. The student is able to model the system using UML language. The student knows the basics of the management of IT project. The student knows the basics of IT architecture.			[SW3] Assessment of knowledge contained in written work and projects		

Subject contents	<p>LECTURE</p> <p>1. Introduction to software engineering. Properties of computer systems engineering. Systems and their environments. Conceptual modeling of the system.</p> <p>2. Project planning. IT project: basic characteristics, concepts, project stakeholders; life cycle and scope of the project. Task scheduling. Problem identification. Rich Picture. Strategic decisions and vision of the solution.</p> <p>3. Feasibility of an IT project. Objectives, assessment levels technical, economic, organizational and legal; venture risk.</p> <p>4. Definition and analysis of requirements for the system. Requirements engineering process. Software requirements and documentation. Features of a good requirement. Methods of obtaining requirements. Division and classification of requirements. Approval of requirements and their management.</p> <p>5. IT project strategies and processes; traditional (cascade model, V model, prototyping, incremental, spiral) and modern software development cycles (reuse and component), MDA, reengineering.</p> <p>6. Agile software development methodologies. Extreme programming. SCRUM: processes, artifacts, roles. Selection of project management strategy.</p> <p>7. UML language. CASE Tools</p> <p>8. Architectural design: presentation of the concept of software architecture and architectural design. Structuring of the system, control models, the distribution of the modules, the architecture characteristic of the various fields. Architecture of distributed systems. Multiprocessing architecture, client-server, distributed objects.</p> <p>9. Designing Object: presentation of an approach to software in which the project has the structure of interacting objects. Objects and object classes, object-oriented design processes, the evolution of the project.</p> <p>10. Optimization of the design for the specificity of the software. Design distributed, mission-critical, real-time and reusable systems. System control for data collection.</p> <p>11. Outsourcing in software engineering.</p> <p>LABORATORY: As part of the laboratory, students perform exercises that make up the basic steps for identification and analysis of requirements and object modeling using CASE tools and UML. The purpose of exercises is to gain the students practical skills in conceptual models and tools supporting design. Work takes place in teams. Each group performs a set of exercises in relation to their choice. The result is a complete system documentation, design of the project and (possibly) an outline of its implementation.</p>											
Prerequisites and co-requisites	None											
Assessment methods and criteria	<table border="1"> <thead> <tr> <th data-bbox="453 1594 794 1621">Subject passing criteria</th> <th data-bbox="799 1594 1141 1621">Passing threshold</th> <th data-bbox="1145 1594 1493 1621">Percentage of the final grade</th> </tr> </thead> <tbody> <tr> <td data-bbox="453 1628 794 1655">Tests</td> <td data-bbox="799 1628 1141 1655">50.0%</td> <td data-bbox="1145 1628 1493 1655">15.0%</td> </tr> <tr> <td data-bbox="453 1662 794 1688">Tasks</td> <td data-bbox="799 1662 1141 1688">50.0%</td> <td data-bbox="1145 1662 1493 1688">85.0%</td> </tr> </tbody> </table>			Subject passing criteria	Passing threshold	Percentage of the final grade	Tests	50.0%	15.0%	Tasks	50.0%	85.0%
Subject passing criteria	Passing threshold	Percentage of the final grade										
Tests	50.0%	15.0%										
Tasks	50.0%	85.0%										
Recommended reading	Basic literature	<p>1. Sommerville I.: Inżynieria oprogramowania, WNT, 2003</p> <p>2. Inżynieria oprogramowania w projekcie informatycznym, red. J. Górski, MIKOM, 2000</p> <p>3. Booch G., Rumbaugh J., Jacobson I.: UML przewodnik użytkownika, WNT 2002</p>										

	Supplementary literature	1. Subieta K.: Wprowadzenie do inżynierii oprogramowania, PJWSTK, 2002  2. Jaskiewicz A.: Inżynieria oprogramowania, Helion, 2000  3. Brooks F. P.: Eseje o inżynierii oprogramowania, WNT, 2000
	eResources addresses	Adresy na platformie eNauczenie: Inżynieria oprogramowania 2024 - Moodle ID: 38315 <a href="https://enauczenie.pg.edu.pl/moodle/course/view.php?id=38315">https://enauczenie.pg.edu.pl/moodle/course/view.php?id=38315</a>
Example issues/ example questions/ tasks being completed	<ol style="list-style-type: none"> <li>1. Software development processes.</li> <li>2. Engineering requirements.</li> <li>3. Agile software development methodology.</li> <li>4. Extreme Programming.</li> <li>5. Programming project management.</li> <li>6. Object-oriented design.</li> <li>7. UML.</li> <li>8. Cost Estimating Software</li> <li>9. Architectural design.</li> <li>10. Design of distributed systems.</li> <li>11. Design of real-time systems.</li> <li>12. Critical Systems.</li> <li>13. Design with reuse.</li> <li>14. Design patterns</li> </ol>	
Work placement	Not applicable	