GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Software engineering, PG_00051071		
Field of study	Technical Physics		
Date of commencement of studies	October 2020	Academic year of realisation of subject	2022/2023
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:
MSTeams platform

https://teams.microsoft.com/l/channel/19%3a6df80652ef104019997384e3e9e29ee8%40thread.tacv2/Og%25C3%25B3lny?groupId=12d7e934-6b22-4753-8914-ea0a50206aa9&tenantId=86760356-0022-486f-b793-a2d470bba5a5

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 ofdeveloping software Student knows rules of acquiring requirements Student knows, how to model thesystem using the UML language Student knows the fundamentals of software projects management Studentknows basic architectures of computer systems.

Learning outcomes	Course outcome	Subject outcome	Method of verification
	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
	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_U02	The student carries out project tasks related to software engineering.	[SU2] Assessment of ability to analyse information [SU1] Assessment of task fulfilment

Subject contents	LECTURE 1. Introduction to software engineering. Properties of computer systems engineering. Systems and their environments. Conceptual modeling of the system. 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. 3. Feasibility of an IT project. Objectives, assessment levels technical, economic, organizational and legal; venture risk. 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. 5. IT project strategies and processes; traditional (cascade model, V model, prototyping, incremental, spiral) and modern software development cycles (reuse and component), MDA, reengineering. 6. Agile software development methodologies. Extreme programming. SCRUM: processes, artifacts, roles. Selection of project management strategy. 7. UML language. CASE Tools 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 ofthe various fields. Architecture of distributed systems. Multiprocessing architecture, client-server, distributed objects. 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. 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. 11. Outsourcing in software engineering. 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.
Prerequisites and co-requisites	None
Assessment methods and criteria	

Subject passing criteria	Passing threshold	Percentage of the final grade
Tasks	50.0%	85.0%
Tests	50.0%	15.0%

Recommended reading	Basic literature	Sommerville I.: Inżynieria oprogramowania, WNT 2003 Inżynieriaoprogramowania w projekcie informatycznym, red. J. Górski, MIKOM2000 Booch G,. Rumbaugh J., Jacobson I.: UML przewodnik użytkownika, WNT 2002

	Supplementary literature	Subieta K.: Wprowadzenie do inżynierii oprogramowania, PJWSTK2002 Jaszkiewicz A.: Inżynieria oprogramowania, Helion 2000 BrooksF. P.: Eseje o inżynierii oprogramowania, WNT 2000
	eResources addresses	Adresy na platformie eNauczanie:
Example issues/ example questions/ tasks being completed	1. Software development processes. 2. Engineering requirements. 3. Agile software development methodology. 4. Extreme Programming. 5. Programming project management. 6. Object-oriented design. 7. UML. 8. Cost Estimating Software 9. Architectural design. 10. Design of distributed systems. 11. Design of real-time systems. 12. Critical Systems. 13. Design with reuse. 14. Design patterns	
Work placement	Not applicable	