GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Software Engineering, PG_00053894
Field of study	Informatics

Date of commencement of studies	October 2020	Academic year of realisation of subject	2022/2023
Education level	first-cycle studies	Subject group	Obligatory subject group in the field of study
			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	5	ECTS credits	4.0
Learning profile	general academic profile	Assessment form	exam

Conducting unit	Department of Software Engineering -> Faculty of Electronics, Telecommunications and Informatics

Name and surname of lecturer (lecturers)	Subject supervisor	dr inż. Aleksander Jarzębowicz
	Teachers	mgr inż. Małgorzata Pykała
		dr inż. Anna Bobkowska
		dr inż. Aleksander Jarzębowicz

Lesson types and methods of instruction	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	30.0	0.0	30.0	0.0	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	The course is focused on introducing to students the aspects of industrial software development: large systems, compliant to requirements of a specific customer, supporting business goals, providing a required level of quality characteristics, produced and maintained by large developers teams.

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[K6_W42] Knows and understands, to an advanced extent, architecture, design principles and methods of hardware and software support for local and distributed information systems, including computing systems, databases, computer networks and information applications, as well as the principles of human cooperation with computers and computer-aided teamwork	The student understands the issues of IT systems design (on both architectural and module level), including software reuse and user interface design.	[SW1] Assessment of factual knowledge
	[K6_U03] can design, according to required specifications, and make a simple device, facility, system or carry out a process, specific to the field of study, using suitable methods, techniques, tools and materials, following engineering standards and norms, applying technologies specific to the field of study and experience gained in the professional engineering environment	The student develops analytical and design models of IT system using CASE (Computer Aided Software Engineering) software supporting tools.	[SU1] Assessment of task fulfilment
	[K6_U09] can carry out a critical analysis of the functioning of existing technical solutions and assess these solutions, as well as apply experience related to the maintenance of technical systems, devices and facilities typical for the field of studies, gained in the professional engineering environment	The student develops "Vision of IT system" document that includes a critical analysis of the present state of the customer organization as well as basic requirements and restrictions of the planned IT system.	[SU1] Assessment of task fulfilment
	[K6_W06] Knows and understands the basic processes occurring in the life cycle of devices, facilities and systems specific to a given field of study.	The student understands the importance of engineering practices and organisation of teamwork in software projects. Enumerates and describes key areas of software development process. Explains the selection of practices for the specific software project.	[SW1] Assessment of factual knowledge
	[K6_U43] can analyse date and formulate, apply and assess appropriate formal models and algorithms for solving problems in the field of information systems and applications	The student uses Unified Modeling Language to prepare the models of IT systems.	[SU1] Assessment of task fulfilment

Subject contents	1. Introduction 2. Scope and subject of software engineering. Essential motivations and concepts. 3. Planning and defining scope of software project. Rich Picture. 4. Areas of software engineering - an overview 5. Conceptual modelling. Languages for modelling and specification. 6. Use cases 7. Object-oriented analysis using UML 8. Modelling of logical system structure: class diagrams 9. Modelling of system structure: other structural diagrams 10. Modelling system dynamics: sequence and communication diagrams 11. Modelling system dynamics: representing object"s state 12. System design: system architecture 13. System design: high-level design 14. System design: class design (low level) 15. Foundations of software quality. Metrics of object-oriented design. 16. Software reuse 17. Classical design patterns 18. Other patterns (Internet Applications patterns, analysis patterns, architectural patterns, management patterns) 19. Risk and social responsibility related to IT systems 20. Requirements engineering: requirements determination 21. Requirements engineering: requirements specification 22. User interface design: motivations, terms, techniques 23. User interface design: Nielsen"s heuristics and examples 24. Software testing: terms, place in software development process 25. Software testing: techniques (black/white box), levels of testing, managing tests 26. Software reviews and inspections 27. Software deployment 28. Software usage and maintenance 29. Configuration management and software evolution 30. Classical (waterfall) software lifecycle model 31. Non-classical software lifecycles and development processes 32. Adjusting development process to particular software project context 33. Outline of software project management 34. Software development and management methodologies 35. Properties of plan-driven and agile development 36. CASE tools 37. Other tools supporting software engineering
Prerequisites and co-requisites	Presence during laboratory courses is mandatory. Delivery of all laboratory exercises and positive verification by tutor is required to pass the lab. Delays in delivering exercises affects the assessments. Only students who pass the lab are entitled to write the exam.

Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	Written exam	50.0%	50.0%
	Lab (assignments & tests)	50.0%	50.0%

Recommended reading	Basic literature	1. Pressman R., Software Engineering: a Practitioner's Approach, 8th edition, McGraw-Hill, 2014 2. Sommerville I., Software Engineering, 9th edition, Addison-Wesley, 2010 3. Maciaszek L.: Requirements analysis and system design, Addison-Wesley, 2007 4. Booch G., Rumbaugh J., Jacobsen I.: The Unified Modeling Language User Guide, 2nd edition, Addison-Wesley, 2005 5. Fowler M., UML distilled, 3rd edition, Addison-Wesley, 2003
	Supplementary literature	No requirements
	eResources addresses	Adresy na platformie eNauczanie: Inżynieria Oprogramowania 2022/2023 - Moodle ID: 23788 https://enauczanie.pg.edu.pl/moodle/course/view.php?id=23788

Example issues/ example questions/ tasks being completed	
Work placement	Not applicable