GDAŃSK UNIVERSITY OF TECHNOLOGY

Subject card

Subject name and code	Applied software - team project, PG_00037523		
Field of study	Technical Physics		
Date of commencement of studies	October 2023	Academic year of realisation of subject	2026/2027
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	4	Language of instruction	Polish
Semester of study	7	ECTS credits	4.0
Learning profile	general academic profile	Assessment form	assessment
Conducting unit	Institute of Physics and Applied Computer Science -> Faculty of Applied Physics and Mathematics -> Faculties of Gdańsk University of Technology		
Name and surname of lecturer (lecturers)	Subject supervisor	dr hab. inż. Marta Łabuda	
	Teachers		

Lesson types

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	15.0	0.0	0.0	60.0	0.0	75

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	75	10.0	15.0	100

Subject objectives

The objective of the course is to familiarize students with the role of application software in information systems and to develop their skills in creating and managing it. The course covers topics related to IT project management, including resource planning, scheduling, and budgeting, as well as software configuration management and version control. Students will learn methods for software implementation and integration in IT systems and gain knowledge of testing processes, including unit, integration, and system tests. The course also emphasizes test automation, the application of TDD and BDD, and ensuring software quality at different stages of its lifecycle. The aim is to prepare students for effective design, implementation, and management of the application software lifecycle in various technological environments.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[K6_U03] Knows programming languages and can use basic software packages	The student is able to create, compile, and run programs in a chosen programming language using basic language constructs (control statements, data structures, functions/methods). The student can also utilize selected software packages and development tools (e.g., IDE, version control system, repository, standard libraries, frontend frameworks, computational programs) in the software development process.	[SU1] Assessment of task fulfilment
[K6_K05] Can present own work results, transfer information in a commonly understandable manner, communicate and self-evaluate, as well as constructively evaluate the effects of other persons' work.	The student is able to prepare and present the results of a team IT project in a way that is understandable to audiences with varying levels of technical knowledge. The student can also perform self-assessment of their own work and constructively evaluate the contributions and outcomes of other team members.	[SK4] Assessment of communication skills, including language correctness [SK1] Assessment of group work skills
[K6_K04] Can cooperate and work in a group, performing different functions.	The student is familiar with the basic team roles in IT projects (e.g., analyst, designer, programmer, tester, team leader, Scrum Master, Product Owner) and understands the principles of effective team collaboration. The student is able to work efficiently in a project team, responsibly performing the assigned role, adhering to established collaboration rules, and adapting to the evolving needs of the team. The student can communicate within the team, delegate tasks, report progress, and actively participate in team decision-making.	[SK1] Assessment of group work skills
[K6_W05] Has knowledge of programming methodology and techniques, and the use of selected IT tools in physics and technology.	The student is familiar with software development methodologies (e.g., waterfall, iterative, agile) and basic programming techniques. The student knows selected IT tools that support the software development process (programming environments, version control systems, modeling tools such as CASE for UML diagrams, and testing tools), can use them effectively, and understands their application in solving problems in physics and engineering.	[SW1] Assessment of factual knowledge
[K6_U09] Can use technical literature in English.	The student is familiar with the basic specialized literature in English in the field of software engineering and can identify the most important sources of information in this area. The student is able to independently search for and select specialized literature in English, use it in research work, and assess its reliability and relevance.	[SU4] Assessment of ability to use methods and tools
[K6_K01] Understands the need to learn and improve professional and personal competencies. Can inspire and organize other people's learning process	The student is aware of opportunities for developing professional competencies related to using and improving skills in working with modern application software and understands the importance of updating knowledge in a rapidly changing IT environment.	[SK5] Assessment of ability to solve problems that arise in practice

Subject contents	Course content – lecture 1. Definition, classification, and role of application software in information systems. Applications in science and technology. 2. IT project management: resources, personnel, schedule, and budget. 3. Software configuration management. Overview of popular applications and platforms. Tool environments as well as technologies and frameworks. Use of CASE tools. 4. Software implementation and integration. 5. Coding, version control, and code security. 6. Software testing and validation. Types of tests: unit, integration, and system tests; validation and acceptance testing. 7. Typical testing phases and methods. Static, functional, and structural testing. Success factors and outcomes of testing. Test documentation. Software reviews. 8. Test process automation. Debugging tools. Test-driven and behavior-driven development (TDD and BDD). 9. Software quality. Software quality assurance. Quality management. Costs of quality assurance. Best practices. 10. Software installation and deployment. Software maintenance. Analysis of the need for modifications. Change management. Software maintenance costs. Key success factors of the maintenance phase. Course content – project Application development based on the adopted methodology, patterns (analytical, design), frameworks, and components, along with project documentation. Work with a repository and project management software: 1. Introduction. Assumptions of the project work, its execution, and documentation. Discussion of project topics and project teams. Project description and design assumptions. Presentation of sample design solutions. 2. Presentation of ideas (project pitch). Problem identification. Context and stakeholders of the problem; project client. Enhanced profile, system definition and vision, system scope. 3. Project feasibility study: objective, product, scope of the undertaking; risk identification, selection of the project implementation strategy. Project work schedule. 4. Selection of a project management strategy. Further implementation steps depending on the adopted methodology (waterfall, agile). 5. Implementation work: system architecture design, logical and static models, applied algorithms, GUI design, database design. 6. Implementation work: software development. 7. Implementation work: code inspection (code review). Inspection report. 8. Implementation work: software integration. 9. Testing and validation of the developed software; test report. 10. Finalization of project, implementation, and documentation work. Preparation of the Final Report. Technical acceptance of the project. Presentation, discussion of results, and project summary. Each week of work concludes with a Project Progress Report.
Prerequisites and co-requisites	Ability to make an object-oriented programming; Knowledge of software engineering
Assessment methods and criteria	

Subject passing criteria	Passing threshold	Percentage of the final grade
Project	50.0%	80.0%
Presentation	50.0%	10.0%
Reports	50.0%	10.0%

Recommended reading	Basic literature	1. Robert C. Martin, *Czysty kod*, Helion, 2020 (dodruk/aktualne wydanie). 2. Marcin Kawalec, *Scrum i nie tylko. Zwinne metody w praktyce*, Helion, 2019. 3. Paweł Jakubczyk, *Testowanie oprogramowania. Podstawy*, Helion, 2019. 4. Robert. Patton: *Testowanie oprogramowania*, Mikom, Warszawa, 2002I. 5. Mariot Tsitoara --Git i GitHub. Kontrola wersji, zarządzanie projektami i zasady pracy zespołowej (oryg: Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer) 6. Michał Bartyzel *DevOps. Inżynieria niezawodności systemów IT* Helion, 2020
	Supplementary literature	1. Andrzej Jaszkiewicz, *Zarządzanie projektami informatycznymi*, PWN, 2019. 2. Leszek Bukowski, *Analiza i projektowanie systemów informatycznych* , PWN, 2018. 3. Adam Roman, Michał Sadowski, *Testowanie i jakość oprogramowania*, PWN, 2021. 4.List of the accessible homepages of the selected by students IT technologies in which the group project is prepared.
	eResources addresses	

Example issues/ example questions/ tasks being completed	Topics: 1. The role and importance of application software in modern information systems. 2. Planning and management of an IT project, including resources, scheduling, and budgeting. 3. Software configuration and version control management using modern tools and platforms. 4. The process of software implementation and integration in complex information systems. 5. Methods and techniques of software testing and their impact on the quality of the final product. 6. Test automation and the application of TDD and BDD approaches in the software development process. 7. Software quality assurance and maintenance best practices, costs, and change management. Tasks: Project schedule Reports of the work development Implementation of the project Code review Testing Project presentattion
Practical activites within the subject	Not applicable

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