

Karta przedmiotu

Nazwa i kod przedmiotu	Software engineering, PG_00045302						
Kierunek studiów	Inżynieria danych						
Data rozpoczęcia studiów	październik 2025 r.		Rok akademicki realizacji przedmiotu		2026/2027		
Poziom kształcenia	I stopnia - inżynierskie		Grupa zajęć		Grupa zajęć obowiązkowych z zakresu kierunku studiów Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne		Sposób realizacji		na uczelni		
Rok studiów	2		Język wykładowy		angielski		
Semestr studiów	3		Liczba punktów ECTS		3.0		
Profil kształcenia	ogólnoakademicki		Forma zaliczenia		egzamin		
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Inżynierii Oprogramowania						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr inż. Aleksander Jarzębowicz				
	Prowadzący zajęcia z przedmiotu		dr inż. Aleksander Jarzębowicz				
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	30.0	0.0	0.0	45
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	45		6.0		24.0	75
Cel przedmiotu	Celem przedmiotu jest zapoznanie studentów z obszarami analizy i projektowania na tle innych obszarów działań projektu informatycznego oraz praktyczna nauka wykorzystania języka modelowania UML do analizy i projektowania systemów IT.						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu			Sposób weryfikacji i oceny efektu	
	[K6_K03] wykazuje się zdolnością do krytycznego i analitycznego myślenia oraz integruje wiedzę z wielu dyscyplin w celu podejmowania efektywnych decyzji		Student realizuje w zespole projekt dla wybranego tematu, wykorzystując w tym celu wiedzę z wykładów oraz narzędzia w postaci języka modelowania i systemu CASE (Computer Aided Software Engineering).			[SK1] Ocena umiejętności pracy w grupie [SK2] Ocena postępów pracy	
	[K6_W01] identyfikuje uwarunkowania procesów zachodzących w analizowanych systemach i dobiera metody ich rozwiązania, wykorzystując zgromadzoną wiedzę i uwzględniając wzajemne relacje między analizowanymi zjawiskami		Student potrafi tworzyć modele analityczne obejmujące różne perspektywy spojrzenia na system (funkcjonalna, strukturalna, dynamiczna) oraz dbać o spójność pomiędzy tymi modelami.			[SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym	
	[K6_U04] formułuje logiczne rozwiązania złożonych lub nieustrukturyzowanych problemów		Student potrafi opisać organizację będącą klientem/odbiorcą projektu informatycznego, wymagania względem tworzonego systemu informatycznego oraz wyrazić te wymagania w ustrukturalizowanym języku modelowania.			[SU1] Ocena realizacji zadania [SU4] Ocena umiejętności korzystania z metod i narzędzi	

Treści przedmiotu	1. Wprowadzenie do przedmiotu 2. Zakres i przedmiot inżynierii oprogramowania. Podstawowe motywacje i pojęcia 3. Obszary i procesy inżynierii oprogramowania 4. Cykle życia oprogramowania (zarys) 5. Metodyki wytwarzania oprogramowania (zarys) 6. Pojęcie modelowania konceptualnego. Języki specyfikacji i modelowania. 7. Przypadki użycia 8. Obiektowe podejście do analizy systemu w UML 9. Modelowanie logicznej struktury systemu: diagramy klas 10. Modelowanie struktury: inne diagramy struktury 11. Modelowanie dynamiki: diagramy sekwencji i komunikacji 12. Modelowanie dynamiki: diagramy stanów i czynności 13. Inżynieria wymagań: pozyskiwanie, analiza i walidacja wymagań 14. Inżynieria wymagań: specyfikowanie wymagań 15. Projektowanie: architektura systemu 16. Projektowanie: projekt ogólny (wysokiego poziomu) i projekt klas (szczegółowy) 17. Projektowanie: wzorce projektowe 18. Projektowanie: zasady SOLID 19. Metodyka Scrum		
Wymagania wstępne i dodatkowe			
Sposoby i kryteria oceniania osiąganych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Egzamin	50.0%	50.0%
	Laboratorium	50.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	1. Sacha K., Inżynieria Oprogramowania, PWN, 2010 2. Pressman R., Maxim B., Software Engineering: a Practitioners Approach, McGraw-Hill, 9th edition, 2019 3. Booch G., Rumbaugh J., Jacobsen I.: UML przewodnik użytkownika, WNT, 2002	
	Uzupełniająca lista lektur	1. Sommerville I., Software Engineering, 10th edition, Addison-Wesley, 2015 2. Maciaszek L.: Requirements analysis and system design, Addison-Wesley, 2007 3. Fowler M., Scott K.: UML w kropelce 2.0, Lupus 2005 4. McLaughlin B., Pollice G., West D., Head First: Object-Oriented Analysis and Design, Edycja polska (Rusz głową!), Helion, 2008	
	Adresy eZasobów	Adresy na platformie eNauczanie:	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	• Narysuj diagram UML (np. klas, przypadków użycia, stanów) na podstawie podanego opisu wymagań względem systemu. • Przedstaw określony model wytwarzania oprogramowania i omów jego zalety i wady. • Wymień i opisz techniki specyfikacji wymagań względem systemu.		
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.