

# Karta przedmiotu

Nazwa i kod przedmiotu	Software engineering, PG_00045302							
Kierunek studiów	Inżynieria danych							
Data rozpoczęcia studiów	październik 2024 r.		Rok akademicki realizacji przedmiotu		2025/2026			
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							
	Adresy kursu na platformie eNauczanie: Moodle ID: 45745 Software engineering 2025/2026 <a href="https://enauczanie.pg.edu.pl/moodle/course/view.php?id=45745">https://enauczanie.pg.edu.pl/moodle/course/view.php?id=45745</a>							
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_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.			[SU4] Ocena umiejętności korzystania z metod i narzędzi [SU1] Ocena realizacji zadania		
	[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_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).			[SK2] Ocena postępów pracy [SK1] Ocena umiejętności pracy w grupie		

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

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.