



Karta przedmiotu

Nazwa i kod przedmiotu	Software engineering, PG_00045302						
Kierunek studiów	Inżynieria danych						
Data rozpoczęcia studiów	październik 2020 r.	Rok akademicki realizacji przedmiotu	2021/2022				
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	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						
	Software Engineering 2021/2022 - Moodle ID: 14384 https://enauczanie.pg.edu.pl/moodle/course/view.php?id=14384						
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_W05] zna i rozumie modele programowania i ewolucję związanych z nimi języków. zna metody analizowania i projektowania systemów informatycznych i wykorzystywane w nich języki modelowania, a także podstawowe platformy programowania obiektowego		Zna metody analizowania i projektowania systemów informatycznych i wykorzystywane w nich języki modelowania.		[SW1] Ocena wiedzy faktograficznej		
	[K6_U02] projektuje, analizuje poprawność i tworzy specyfikację funkcjonalną systemów informatycznych, dobierając odpowiednie środki, tworzy modele jakości, przygotowuje i ocenia ich dokumentację projektową		Projektuje i tworzy specyfikację funkcjonalną systemów informatycznych, dobierając odpowiednie środki, przygotowuje dokumentację projektową w zakresie modeli analitycznych i projektowych.		[SU4] Ocena umiejętności korzystania z metod i narzędzi [SU1] Ocena realizacji zadania		

Treści przedmiotu	<ol style="list-style-type: none"> 1. Wprowadzenie do przedmiotu 2. Zakres i przedmiot inżynierii oprogramowania. Podstawowe motywacje i pojęcia 3. Obszary działania inżynierii oprogramowania 4. Inżynieria wymagań: pozyskiwanie, analiza i walidacja wymagań 5. Inżynieria wymagań: specyfikowanie wymagań 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: reprezentowanie stanu obiektów 13. Projektowanie: architektura systemu 14. Projektowanie: projekt ogólny (wysokiego poziomu) i projekt klas (szczegółowy) 15. Zagadnienia software reuse i wzorców projektowych 16. Modele wytwarzania oprogramowania (cykle życia oprogramowania) 17. Metodyki wytwarzania oprogramowania (zarys) 		
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	<ol style="list-style-type: none"> 1. Sacha K., Inżynieria Oprogramowania, PWN, 2010 2. Pressman R., Software Engineering: a Practitioner's Approach, 8th edition, McGraw-Hill, 2014 3. Booch G., Rumbaugh J., Jacobsen I.: UML przewodnik użytkownika, WNT, 2002 	
	Uzupełniająca lista lektur	<ol style="list-style-type: none"> 1. Sommerville I., Software Engineering, 9th edition, Addison-Wesley, 2010 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		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<ul style="list-style-type: none"> • 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		