



Karta przedmiotu

Nazwa i kod przedmiotu	Inżynieria oprogramowania, PG_00044135						
Kierunek studiów	Matematyka						
Data rozpoczęcia studiów	październik 2023 r.	Rok akademicki realizacji przedmiotu			2023/2024		
Poziom kształcenia	II stopnia	Grupa zajęć			Grupa zajęć fakultatywnych		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			polski		
Semestr studiów	2	Liczba punktów ECTS			4.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydział Fizyki Technicznej i Matematyki Stosowanej -> Katedra Równań Różniczkowych i Zastosowań Matematyki						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr hab. Paweł Pilarczyk					
	Prowadzący zajęcia z przedmiotu	dr hab. Paweł Pilarczyk					
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	15.0	15.0	0.0	60
	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	60		5.0	35.0		100
Cel przedmiotu	Poznanie podstawowych metod inżynierii oprogramowania i zdobycie umiejętności ich stosowania. Dotyczy pracy grupowej nad projektem informatycznym dużych rozmiarów na wszystkich etapach wytwarzania oprogramowania: od inżynierii wymagań, poprzez analizę wymagań, projektowanie oprogramowania, implementację, testowanie, wdrażanie, aż do etapu konserwacji oprogramowania. Zapoznanie się z podstawowymi zagadnieniami dotyczącymi kierowania projektem informatycznym, jak np. zarządzanie jakością.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K7_W11] zna matematyczne podstawy teorii informacji, teorii algorytmów i kryptografii oraz ich praktyczne zastosowania m.in. w programowaniu i szeroko rozumianej informatyce	Zna czynności wykonywane przy projektowaniu i tworzeniu oprogramowania.	[SW1] Ocena wiedzy faktograficznej [SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym
	[K7_U13] rozumie matematyczne podstawy analizy algorytmów i procesów obliczeniowych, potrafi konstruować algorytmy o dobrych własnościach numerycznych, służące do rozwiązywania typowych i nietypowych problemów matematycznych	Potrafi projektować i tworzyć oprogramowanie spełniające wybrane założenia.	[SU1] Ocena realizacji zadania [SU4] Ocena umiejętności korzystania z metod i narzędzi
	[K7_K03] potrafi pracować zespołowo; rozumie konieczność systematycznej pracy nad wszelkimi projektami, które mają długofalowy charakter, rozumie i docenia znaczenie uczciwości intelektualnej w działaniach własnych i innych osób; postępuje etycznie	Umie opisać wybrane procesy wytwarzania oprogramowania (m.in. kaskadowy, iteracyjny, zwinny). Potrafi pracować w zespole nad tworzeniem oprogramowania. Umie przeprowadzić inżynierię wymagań oprogramowania, dokonać analizy wymagań, zaprojektować oprogramowanie. Potrafi używać języka UML do modelowania wybranych aspektów oprogramowania. Zna i rozumie zasady etyki inżyniera oprogramowania.	[SK3] Ocena umiejętności organizacji pracy [SK4] Ocena umiejętności komunikacji, w tym poprawności językowej [SK5] Ocena umiejętności rozwiązywania problemów występujących w praktyce [SK1] Ocena umiejętności pracy w grupie
[K7_W08] zna zaawansowane techniki obliczeniowe, wspomagające pracę matematyka i rozumie ich ograniczenia	Zna metody inżynierii oprogramowania służące projektowaniu, tworzeniu, testowaniu i rozwojowi oprogramowania.	[SW2] Ocena wiedzy zawartej w prezentacji [SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym	
Treści przedmiotu	<p>Wykład: Wprowadzenie do inżynierii oprogramowania. Zasady etyczne. Procesy wytwarzania oprogramowania. Inżynieria wymagań. Metody strukturalne i metody obiektowe. Modelowanie przy wykorzystaniu UML. Testowanie oprogramowania. Zarządzanie projektami informatycznymi, zarządzanie jakością. Konserwacja oprogramowania. Systemy krytyczne. Manifest zwinnego oprogramowania (Agile Manifesto) i metodyka zwinna, w tym Scrum.</p> <p>Laboratorium i projekt: Stosowanie metod inżynierii oprogramowania do projektowania konkretnego oprogramowania, w tym opracowywanie diagramów UML, przygotowywanie historyjek użytkownika. Praktyczne przygotowanie prototypów oprogramowania.</p>		
Wymagania wstępne i dodatkowe	Podstawowa umiejętność pisania programów komputerowych, np. w Pythonie, C lub HTML/JavaScript.		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	kartkówki (pisemne, 10-minutowe)	60.0%	50.0%
	zadania domowe, projekt, prezentacje itp.	60.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	<p>Krzysztof Sacha, Inżynieria oprogramowania, PWN 2010.</p> <p>Mariusz Chrapko, Scrum. O zwinnym zarządzaniu projektami. Wydanie II rozszerzone, Helion 2014.</p>	
	Uzupełniająca lista lektur	<p>Ian Sommerville, Inżynieria oprogramowania, WNT 2003.</p> <p>Roger S. Pressman, Praktyczne podejście do inżynierii oprogramowania, WNT 2004.</p> <p>Russ Miles, Kim Hamilton, UML 2.0. Wprowadzenie, Helion 2007.</p> <p>Kenneth S. Rubin. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Pearson Education, 2013.</p>	
	Adresy eZasobów	<p>Adresy na platformie eNauczanie:</p> <p>Inżynieria oprogramowania 2024 - Moodle ID: 31164  <a href="https://enauczanie.pg.edu.pl/moodle/course/view.php?id=31164">https://enauczanie.pg.edu.pl/moodle/course/view.php?id=31164</a></p>	

Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	Inżynieria wymagań.  Procesy wytwarzania oprogramowania.  Weryfikacja a zatwierdzanie.  Metodyka zwinna i jej Manifest Agile.  Zasady etyki inżyniera oprogramowania.  Cechy INVEST historyjek użytkownika.
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy