



Karta przedmiotu

Nazwa i kod przedmiotu	Inżynieria oprogramowania, PG_00053894						
Kierunek studiów	Informatyka						
Data rozpoczęcia studiów	październik 2023 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	3		Język wykładowy		polski		
Semestr studiów	5		Liczba punktów ECTS		4.0		
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	30.0	0.0	30.0	0.0	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	<p>Przedmiot "Inżynieria Oprogramowania" jest ukierunkowany na przybliżenie zagadnień związanych z wytwarzaniem oprogramowania w warunkach przemysłowych: złożone systemy, przeznaczone dla rzeczywistego klienta, związane z określoną potrzebą biznesową i gwarancjami jakości, wytwarzane przez duże zespoły deweloperów.</p> <p>W ramach wykładu omawiane są kluczowe obszary procesu wytwarzania: planowanie i zarządzanie, inżynieria wymagań, analiza i projektowanie, implementacja, testowanie, wdrożenie i utrzymanie, wspomaganie narzędziowe, praca zespołowa.</p>						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K6_U43] potrafi analizować dane oraz formułować, stosować i oceniać właściwe modele formalne i algorytmy rozwiązywania problemów w zakresie systemów i aplikacji informacyjnych	Student posługuje się notacją modelowania Unified Modeling Language (UML) i konstruuje modele systemów informatycznych.	[SU1] Ocena realizacji zadania
	[K6_W06] zna i rozumie podstawowe procesy zachodzące w cyklu życia urządzeń, obiektów i systemów specyficznych dla danego kierunku studiów	Student rozumie znaczenie praktyk inżynierskich i organizacji pracy zespołowej w przedsięwzięciu informatycznym. Wymienia i opisuje kluczowe obszary procesu wytwarzania oprogramowania. Wyjaśnia dobór praktyk w zależności od specyfiki przedsięwzięcia.	[SW1] Ocena wiedzy faktograficznej
	[K6_U09] potrafi dokonać krytycznej analizy sposobu funkcjonowania istniejących rozwiązań technicznych związanych z kierunkiem studiów i ocenić te rozwiązania, a także wykorzystać zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską doświadczenie związane z utrzymaniem urządzeń, obiektów i systemów technicznych typowych dla kierunku studiów	Student opracowuje wizję systemu informatycznego zawierającą krytyczną analizę obecnego sposobu funkcjonowania organizacji klienckiej oraz podstawowe wymagania i ograniczenia względem systemu.	[SU1] Ocena realizacji zadania
	[K6_U03] potrafi zaprojektować, zgodnie z zadaną specyfikacją, oraz wykonać typowe dla kierunku studiów proste urządzenie, obiekt, system lub zrealizować proces, używając odpowiednio dobranych metod, technik, narzędzi i materiałów, korzystając ze standardów i norm inżynierskich, stosując właściwe dla kierunków studiów technologie i wykorzystując doświadczenie zdobyte w środowisku zajmującym się zawodowo działalnością inżynierską	Student potrafi wykonać modele analityczne i projektowe systemu informatycznego posługując się w tym celu narzędziami CASE (Computer Aided Software Engineering).	[SU1] Ocena realizacji zadania
	[K6_W42] zna i rozumie w zaawansowanym stopniu architektury, zasady projektowania oraz metody wsparcia sprzętowego i programowego dla lokalnych i rozproszonych systemów informatycznych, w tym systemów obliczeniowych, baz danych, sieci komputerowych i aplikacji informacyjnych, a także zasady współpracy człowieka z komputerem i wspomaganą komputerowo pracy zespołowej	Student rozumie zagadnienia projektowania systemów informatycznych (na poziomie architektury i poszczególnych modułów składowych), w tym aspekty software reuse oraz projektowania interfejsu użytkownika.	[SW1] Ocena wiedzy faktograficznej

Treści przedmiotu	<ol style="list-style-type: none"> 1. Wprowadzenie do przedmiotu 2. Zakres i przedmiot inżynierii oprogramowania. Podstawowe motywacje i pojęcia. 3. Faza przedprojektowa: planowanie i zakres przedsięwzięcia. Rich Picture. 4. Obszary działania inżynierii oprogramowania 5. Ryzyko i odpowiedzialność społeczna związane z systemami informatycznymi 6. Inżynieria wymagań: pozyskiwanie wymagań 7. Inżynieria wymagań: specyfikowanie wymagań 8. Pojęcie modelowania konceptualnego. Języki specyfikacji i modelowania. 9. Przyrodki użycia 10. Obiektywne podejście do analizy systemu w UML 11. Modelowanie logicznej struktury systemu: diagramy klas 12. Modelowanie struktury: inne diagramy struktury 13. Modelowanie dynamiki: diagramy sekwencji i komunikacji 14. Modelowanie dynamiki: reprezentowanie stanu obiektów 15. Projektowanie: architektura systemu 16. Projektowanie: Projekt ogólny (wysokiego poziomu) 17. Projektowanie: Projekt klas (szczegółowy) 18. Zagadnienia jakości. Metryki projektowania obiektowego. 19. Zagadnienia software reuse 20. Klasyczne wzorce projektowe 21. Inne rodzaje wzorców (wzorce aplikacji internetowych, wzorce analityczne, architektoniczne, zarządzania) 22. Projektowanie interfejsu użytkownika: motywacje, pojęcia, techniki 23. Projektowanie interfejsu użytkownika: heurystyki Nielsena i przykłady 24. Testowanie: pojęcia, umiejscowienie w procesie wytwarzania 25. Testowanie: techniki (czarna i biała skrzynka), poziomy testowania, zarządzanie testowaniem 26. Przeglądy i inspekcje oprogramowania 27. Wdrażanie oprogramowania 28. Eksploatacja i utrzymanie oprogramowania 29. Zarządzanie konfiguracją i ewolucja oprogramowania 30. Klasyczny cykl życia oprogramowania 31. Nieklasyczne cykle życia i modele wytwarzania oprogramowania 32. Dobór modelu wytwarzania do specyfiki projektu 33. Zarys problematyki zarządzania projektem informatycznym 34. Metodyki wytwórcze i zarządcze 35. Charakterystyka metodyk sterowanych planem i zwinnych 36. Narzędzia CASE 37. Inne narzędzia w inżynierii oprogramowania 											
Wymagania wstępne i dodatkowe	<p>Obowiązkowa obecność na zajęciach laboratoryjnych. Konieczne oddanie i akceptacja wszystkich zadań laboratoryjnych. Oddawanie zadań z opóźnieniem skutkuje punktami ujemnymi. Do egzaminu dopuszczeni są jedynie studenci, którzy zaliczyli laboratorium.</p>											
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Sposób oceniania (składowe)</th> <th style="width: 33%;">Próg zaliczeniowy</th> <th style="width: 33%;">Składowa oceny końcowej</th> </tr> </thead> <tbody> <tr> <td>Laboratorium (zadania i sprawdziany)</td> <td style="text-align: center;">50.0%</td> <td style="text-align: center;">50.0%</td> </tr> <tr> <td>Egzamin pisemny</td> <td style="text-align: center;">50.0%</td> <td style="text-align: center;">50.0%</td> </tr> </tbody> </table>			Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej	Laboratorium (zadania i sprawdziany)	50.0%	50.0%	Egzamin pisemny	50.0%	50.0%
Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej										
Laboratorium (zadania i sprawdziany)	50.0%	50.0%										
Egzamin pisemny	50.0%	50.0%										
Zalecana lista lektur	<p>Podstawowa lista lektur</p> <p>Uzupełniająca lista lektur</p> <p>Adresy eZasobów</p>	<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. Sommerville I., Software Engineering, 9th edition, Addison-Wesley, 2010 4. Maciaszek L.: Requirements analysis and system design, Addison-Wesley, 2007 5. Booch G., Rumbaugh J., Jacobsen I.: UML przewodnik użytkownika, WNT, 2002 6. Fowler M., Scott K.: UML w kropelce 2.0 (ang. UML distilled), Lupus 2005 7. McLaughlin B., Pollice G., West D., Head First: Object-Oriented Analysis and Design, Edycja polska (Rusz głową!), Helion, 2008 <p>Nie ma wymagań</p> <p>Adresy na platformie eNauczanie:</p>										
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania												
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy											