



Karta przedmiotu

Nazwa i kod przedmiotu	Inżynieria oprogramowania, PG_00047848							
Kierunek studiów	Inżynieria biomedyczna							
Data rozpoczęcia studiów	październik 2024 r.	Rok akademicki realizacji przedmiotu			2026/2027			
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć fakultatywnych 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			5.0			
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		60.0		125	
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_W06] zna i rozumie podstawowe procesy zachodzące w cyklu życia urządzeń, obiektów i systemów specyficznych dla danego kierunku studiów		Student wymienia i opisuje kluczowe obszary cyklu życia oprogramowania. Wyjaśnia dobór praktyk w poszczególnych obszarach oraz sposób organizacji całego projektu informatycznego w zależności od jego specyfiki.			[SW1] Ocena wiedzy faktograficznej		
[K6_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów		Student posługuje się notacją modelowania Unified Modeling Language (UML) i konstruuje modele systemów informatycznych. Opracowuje dokument wizji systemu informatycznego ze sformułowaniem podstawowych wymagań i ograniczeń.			[SU1] Ocena realizacji zadania			

Treści przedmiotu	<ol style="list-style-type: none"> <li>1. Wprowadzenie do przedmiotu</li> <li>2. Zakres i przedmiot inżynierii oprogramowania. Podstawowe motywacje i pojęcia.</li> <li>3. Faza przedprojektowa: planowanie i zakres przedsięwzięcia. Rich Picture.</li> <li>4. Obszary działania inżynierii oprogramowania</li> <li>5. Ryzyko i odpowiedzialność społeczna związane z systemami informatycznymi</li> <li>6. Inżynieria wymagań: pozyskiwanie wymagań</li> <li>7. Inżynieria wymagań: specyfikowanie wymagań</li> <li>8. Pojęcie modelowania konceptualnego. Języki specyfikacji i modelowania.</li> <li>9. Przypadki użycia</li> <li>10. Obiektywne podejście do analizy systemu w UML</li> <li>11. Modelowanie logicznej struktury systemu: diagramy klas</li> <li>12. Modelowanie struktury: inne diagramy struktury</li> <li>13. Modelowanie dynamiki: diagramy sekwencji i komunikacji</li> <li>14. Modelowanie dynamiki: reprezentowanie stanu obiektów</li> <li>15. Projektowanie: architektura systemu</li> <li>16. Projektowanie: Projekt ogólny (wysokiego poziomu)</li> <li>17. Projektowanie: Projekt klas (szczegółowy)</li> <li>18. Zagadnienia jakości. Metryki projektowania obiektowego.</li> <li>19. Zagadnienia software reuse</li> <li>20. Klasyczne wzorce projektowe</li> <li>21. Inne rodzaje wzorców (wzorce aplikacji internetowych, wzorce analityczne, architektoniczne, zarządzania)</li> <li>22. Projektowanie interfejsu użytkownika: motywacje, pojęcia, techniki</li> <li>23. Projektowanie interfejsu użytkownika: heurystyki Nielsena i przykłady</li> <li>24. Testowanie: pojęcia, umiejscowienie w procesie wytwarzania</li> <li>25. Testowanie: techniki (czarna i biała skrzynka), poziomy testowania, zarządzanie testowaniem</li> <li>26. Przeglądy i inspekcje oprogramowania</li> <li>27. Wdrażanie oprogramowania</li> <li>28. Eksploatacja i utrzymanie oprogramowania</li> <li>29. Zarządzanie konfiguracją i ewolucja oprogramowania</li> <li>30. Klasyczny cykl życia oprogramowania</li> <li>31. Nieklasyczne cykle życia i modele wytwarzania oprogramowania</li> <li>32. Dobór modelu wytwarzania do specyfiki projektu</li> <li>33. Zarys problematyki zarządzania projektem informatycznym</li> <li>34. Metodyki wytwórcze i zarządcze</li> <li>35. Charakterystyka metodyk sterowanych planem i zwinnych</li> <li>36. Narzędzia CASE</li> <li>37. Inne narzędzia w inżynierii oprogramowania</li> </ol>											
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"> <li>1. Sacha K., Inżynieria Oprogramowania, PWN, 2010</li> <li>2. Pressman R., Software Engineering: a Practitioner's Approach, 8th edition, McGraw-Hill, 2014</li> <li>3. Sommerville I., Software Engineering, 9th edition, Addison-Wesley, 2010</li> <li>4. Maciaszek L.: Requirements analysis and system design, Addison-Wesley, 2007</li> <li>5. Booch G., Rumbaugh J., Jacobsen I.: UML przewodnik użytkownika, WNT, 2002</li> <li>6. Fowler M., Scott K.: UML w kropelce 2.0 (ang. UML distilled), Lupus 2005</li> <li>7. McLaughlin B., Pollice G., West D., Head First: Object-Oriented Analysis and Design, Edycja polska (Rusz głową!), Helion, 2008</li> </ol> <p>Nie ma wymagań</p> <p>Adresy na platformie eNauczanie:</p>										
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania												
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy											