POLITECHNIKA GDAŃSKA

Karta przedmiotu

Nazwa i kod przedmiotu	Inżynieria oprogramowania, PG_00063885		
Kierunek studiów	Informatyka		
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu	2027/2028
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	polski
Semestr studiów	4	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							
	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	2.0	13.0	75

Cel przedmiotu	
	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. 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.

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[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_W44] 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, zasady współpracy człowieka z komputerem, a także działanie i kryteria oceny metod przetwarzania, składowania i przesyłania danych, w tym algorytmów obliczeniowych, sztucznej inteligencji i eksploracji danych oraz standardy i metody administrowania systemami informatycznymi, monitorowania zachodzących w nich procesów oraz uodporniania ich na niepożądane zjawiska i działania	Student rozumie pojęcie architektury systemu informatycznego, wie jakie zagadnienia wchodzą w zakres doboru i zdefiniowania architektury	[SW1] Ocena wiedzy faktograficznej

Treści przedmiotu

Treści przedmiotu - wykład

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. Przypadki użycia
10. Obiektowe 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

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.

Sposoby i kryteria oceniania osiąganych efektów uczenia się

Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
Egzamin pisemny	50.0%	50.0%
Laboratorium (zadania i sprawdziany)	50.0%	50.0%

Zalecana lista lektur

Podstawowa lista lektur	1. Sacha K., Inżynieria Oprogramowania, PWN, 2010 2. Pressman R., Software Engineering: a Practitioners 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
Uzupełniająca lista lektur	Nie ma wymagań
Adresy eZasobów	

Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania

Zajęcia praktyczne w ramach przedmiotu

Nie dotyczy