


KAPITAŁ LUDZKI
 NARODOWA STRATEGIA SPÓJNOŚCI

 Projekt współfinansowany przez
 Unię Europejską w ramach
 Europejskiego Funduszu
 Społecznego

UNIA EUROPEJSKA
 EUROPEJSKI
 FUNDUSZ SPOŁECZNY


Nazwa przedmiotu		Kod ECTS	
Podstawy inżynierii oprogramowania (Z)		11.3.2127	
Nazwa jednostki prowadzącej przedmiot			
Katedra Informatyki Ekonomicznej			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	niestacjonarne (zaoczne)
		moduł	wszystkie
		specjalnościowy	wszystkie
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
prof. UG, dr hab. Bartosz Marcinkowski; prof. UG, dr hab. Viktoriia Onyshchenko; dr Adam Kostulak			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		3 Przedmiot w wymiarze 10h wykładu i 10h laboratorium + praca własna studenta 55h	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia on-line, zajęcia w sali dydaktycznej			
Liczba godzin			
Wykład: 10 godz., Ćw. laboratoryjne: 10 godz.			
Termin realizacji przedmiotu			
2024/2025 letni			
Status przedmiotu		Język wykładowy	
obowiązkowy		polski	
Metody dydaktyczne		Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne	
<ul style="list-style-type: none"> - Dyskusja - Praca w grupach - Rozwiązywanie zadań - Wykład z prezentacją multimedialną 		Sposób zaliczenia	
		Zaliczenie na ocenę	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru - egzamin pisemny (dłuższa wypowiedź pisemna / rozwiązanie problemu) 	
		Podstawowe kryteria oceny	
		Próg zaliczeniowy	Procent ostatecznej oceny
	kolokwium	50%	20%
	projekt	50%	30%
	pisemne zadanie zaliczeniowe	50%	50%
Sposób weryfikacji założonych efektów uczenia się			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	referat	raport	aktywność w dyskusji	obserwacja postawy studenta
Wiedza							
K_W04		X	X				
P_W01			X				
P_W02			X				
P_W03			X				
Umiejętności							
K_U05			X				X
K_U07			X				X
P_U01			X				X
P_U02			X				X
P_U03			X				X
Kompetencje							
K_K02			X				X
							P_K01
		X					X

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi**A. Wymagania formalne**

brak

B. Wymagania wstępne

Języki programowania, algorytmy i struktury danych, bazy danych.

Cele kształcenia

Celem przedmiotu jest zapoznanie studentów z podstawowymi problemami, metodami, technikami i narzędziami produkcji oprogramowania wysokiej jakości. Nacisk jest położony na obiektowe podejście do projektowania i modelowanie systemu w UML.

Treści programowe

1. Proces tworzenia Systemów Informatycznych
2. Pojęcie procesu i projektu (budowy) oprogramowania. Modele cyklu życia oprogramowania (model kaskadowy, realizacja kierowana dokumentami, V-model, prototypowanie, model przyrostowy, spiralny)
3. Określanie wymagań i analiza. Rodzaje i role notacji wykorzystywanych w fazie analizy, obiektowe i strukturalne metody analizy.
4. UML 2. – standard modelowania obiektowego: poziomy model i typy diagramów. Przykłady użycia (use case)- elementy diagramów, organizowanie. Model logiczny - diagramy klas, diagramy behawioralne (sekwencji, stanów, aktywności), Model implementacyjny – diagram komponentów (komponenty, podsystemy, porty) i model fizyczny (diagram montażowy).
5. Testowanie i inspekcja. Testowanie ukierunkowane na wyszukiwanie defektów – funkcjonalne, strukturalne (złożoność cyklomatyczna McCabe, pokrycie kodu - ścieżki niezależne, pokrycie danych, testowanie pętli). Testowanie obiektów. Estymacja kosztów. Metoda punktów funkcyjnych, model COCOMO 81 i COCOMO II. Model CMM. Miary niezawodności oprogramowania i techniki programowania dla systemów o dużej niezawodności.

Wykaz literatury

1. I. Sommerville: Inżynieria oprogramowania, WNT 2003
2. Szejko St. (red). Metody wytwarzania oprogramowania. MIKOM, 2002.
3. Jaskiewicz A.: Inżynieria oprogramowania. Helion, 1997.
4. Materiały umieszczone na platformie edukacyjnej.

Uzupełniająca:

5. Booch G., Rumbaugh J., Jacobson I.: UML podręcznik użytkownika. WNT, 2001
6. Dumnicki R., Kasprzyk A., Kozłowski M.: Analiza i projektowanie obiektowe. Helion, 1998
7. Eriksson H-E, Penker M.: UML Toolkit. Wiley Computer Publishing, John Wiley & Sons, Inc. 1998.
8. Pressman R.S.: Software Engineering. A Practitioner's Approach. McGraw-Hill, Inc. 1992.

Kierunkowe efekty uczenia się

K_W04: ma uporządkowaną wiedzę w zakresie inżynierii oprogramowania, specyfikacji, walidacji i weryfikacji oprogramowania oraz narzędzi wspomagających proces

Wiedza

P_W01: Student zna różne modele cyklu życia oprogramowania; (K_W04)
 P_W02: Student zna metody wytwarzania oprogramowania; (K_W04)
 P_W03: Student posiada wiedzę odnośnie podstaw wytworzenia dokumentacji,

<p>wytwarzania oprogramowania</p> <p>K_U04: potrafi pracować w zespole informatyków, zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminy, porozumiewać się przy użyciu różnych technik w tym z wykorzystaniem dedykowanych narzędzi</p> <p>K_U06: potrafi projektować, tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz adekwatnych wzorców</p> <p>K_K02: rozumie potrzebę i docenia zalety pracy zespołowej, rozumie konieczność systematycznej pracy nad zespołowymi projektami informatycznymi, jest gotów do aktywnego działania w zespole</p>	<p>analizy wymagań, projektowania, testowania oprogramowania. (K_W04)</p>
	<p>Umiejętności</p> <p>P_U01: Student potrafi zarządzać analizą wymagań; (K_U04, K_U06)</p> <p>P_U02: Student potrafi tworzyć specyfikację ze zdefiniowanymi metrykami; (K_U04)</p> <p>P_U03: Student potrafi utworzyć projekt oprogramowania za pomocą języka UML. (K_U04, K_U06)</p>
	<p>Kompetencje społeczne (postawy)</p> <p>P_K01: Student posiada podstawowe umiejętności komunikacji w zespole informatycznym (K_K02)</p>
<p>Kontakt</p> <p>Bartosz.Marcinkowski@ug.edu.pl</p>	