

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2021/2022

Wydział Informatyki i Telekomunikacji

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma studiów: niestacjonarne

Kod kierunku: I

Stopień studiów: I

Specjalności: Brak specjalności

1 INFORMACJE O PRZEDMIOCIE

NAZWA PRZEDMIOTU	Inżynieria oprogramowania
NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM	Software engineering
KOD PRZEDMIOTU	WiIT I oIN C22 21/22
KATEGORIA PRZEDMIOTU	Przedmioty kierunkowe
LICZBA PUNKTÓW ECTS	4.00
SEMESTRY	6

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

SEMESTR	WYKŁAD	ĆWICZENIA	LABORATORIUM	LABORATORIUM KOMPUTERO- WE	SEMINARIUM	PROJEKT
6	18	0	0	0	0	18

3 CELE PRZEDMIOTU

Cel 1 Nabycie umiejętności dekompozycji rzeczywistości będącej podstawą w budowaniu kolejnych modeli inżynierii oprogramowania: od modelu opisowego, przez model wymagań i specyfikacji do numerycznego. Umiejętność identyfikacji procesów analizy i syntezy w procesach wytwórczych oprogramowania.

Cel 2 Poznanie podstawowych modeli wytwarzania oprogramowania (cykli życia): kaskadowych, cyklicznych i ewo-

lucyjnych. Poznanie poszczególnych faz procesu wytwórczego oraz tworzonych w efekcie ich realizacji artefaktów. Umiejętność adaptacji poszczególnych faz w procesach ciężkich i lekkich.

Cel 3 Poznanie metodyk identyfikacji i specyfikacji wymagań oraz procesów zarządzania nimi. Zdobywanie umiejętności modelowania przy wykorzystaniu UML. Zapoznanie się z metodami estymacji kosztów wytwarzania oprogramowania.

Cel 4 Zaznajomienie się zasadami projektowania, implementacji, testowania i wdrożenia oprogramowania. Zdobywanie umiejętności identyfikacji metodyk formalnych i agilnych w procesach wytwórczych oprogramowania. Nabycie zdolności i umiejętności dokumentowania kolejnych kroków i faz procesu wytwarzania oprogramowania.

4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1 Znajomość dowolnego języka programowania wysokiego poziomu w zakresie pozwalającym na implementację prostej aplikacji

5 EFEKTY KSZTAŁCENIA

EK1 Wiedza Zdobywa wiedzę dotyczącą zagadnień dekompozycji rzeczywistości. Zna sposoby prowadzenia analizy zachodzących procesów oraz rozumie techniki ich modelowania. Rozumie na czym polega wykorzystywanie abstrakcji przy przejściu od rzeczywistości do jej cyfrowego modelu poprzez analizę modelu wymagań i syntezę do modelu systemu (I1_W01, I1W04)

EK2 Wiedza Zdobywa wiedzę na temat procesu wytwórczego oprogramowania i realizacji poszczególnych faz jego cyklu życia (I1_W09). Ma wiedzę w zakresie projektowania i implementacji prostych i złożonych systemów informatycznych przy wykorzystaniu współczesnych technik programistycznych i języków oprogramowania (I1_W10)

EK6 Umiejętności Zdobywa umiejętności transformacji specyfikowanego modelu wymagań na odpowiedni model obiektowy lub strukturalny (I1_U08, I1_U23) z wykorzystaniem prostych systemów bazodanowych (I1_U13)

EK7 Umiejętności Potrafi w procesie projektowania systemu określić jego czaso- i kosztochłonność (I1_U09). Nabycie umiejętności wykorzystania odpowiednich technologii programistycznych w transformacji modelu numerycznego (diagramy) na odpowiedni model cyfrowy (I1_U07)

6 TREŚCI PROGRAMOWE

PROJEKT		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
P1	Realizacja projektu składającego się z czterech podprojektów: - specyfikacja - architektura (UML) - implementacja - unit testy	18

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W1	Wprowadzenie do przedmiotu Inżynieria Oprogramowania - tło historyczne, odniesienie do innych dyscyplin inżynierskich - ogólna charakterystyka.	1

WYKŁAD		
LP	TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH	LICZBA GODZIN
W2	Procesy i metody produkcji oprogramowania. Modele cyklu życia oprogramowania: kaskadowe, iteracyjne, ewolucyjne. Porównanie cykli życia oprogramowania - problemy projektów dużej i małej skali. Wprowadzenie do Nauki o Modelach, zagadnienia właściwej transformacji wymagań.	2
W3	Analiza od rzeczywistości przez model opisowy, specyfikacje wymagań do modelu numerycznego i następnie modelu cyfrowego rzeczywistości, czyli systemu. Rola wymagań i sposoby ich specyfikacji oraz dokumentacji. Szacowanie kosztów realizacji systemu.	2
W4	Dynamika i statyka systemu informatycznego. Języki i paradygmaty wspomagania modelowania systemów informatycznych (diagramy UML)	2
W5	Projektowanie oprogramowania, model strukturalny a obiektowy. Transformacja od modelu numerycznego do modelu cyfrowego, architektura aplikacji, strategie narzędzia. Wzorce projektowe, model komponentowy	2
W6	Implementacja systemu informatycznego. Dobór technologii do wymagań i projektu. Standardy.	2
W7	Zagadnienia jakości, testowanie - sposoby testowania: white/black box testing, testy regresyjne, ciągła integracja, inspekcje, przeglądy artefaktów, metryki jakości	2
W8	Procesy i praktyki zwinne "agilne" na przykładzie XP i Scrum. Dobór metodyki do wielkości projektu.	2
W9	Wybrane zagadnienia inżynierii oprogramowania: zarządzanie konfiguracją, zarządzanie zmianami, zarządzanie ryzykiem, elementy pracy w zespole projektowym.	3

7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia projektowe

8 OBCIĄŻENIE PRACĄ STUDENTA

FORMA AKTYWNOŚCI	ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	36
Konsultacje przedmiotowe	10
Egzaminy i zaliczenia w sesji	0
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	30
Opracowanie wyników	14
Przygotowanie raportu, projektu, prezentacji, dyskusji	30
SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA	120
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	4.00

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Projekt indywidualny

OCENA PODSUMOWUJĄCA

P1 Test

P2 Egzamin ustny

OCENA AKTYWNOŚCI BEZ UDZIAŁU NAUCZYCIELA

B1 Projekt indywidualny

KRYTERIA OCENY

EFEKT KSZTAŁCENIA 1	
NA OCENĘ 3.0	opanowanie materiału w 50%
NA OCENĘ 3.5	opanowanie materiału w 60%
NA OCENĘ 4.0	opanowanie materiału w 70%
NA OCENĘ 4.5	opanowanie materiału w 80%

NA OCENĘ 5.0	opanowanie materiału w 90%
EFEKT KSZTAŁCENIA 2	
NA OCENĘ 3.0	opanowanie materiału w 50%
NA OCENĘ 3.5	opanowanie materiału w 60%
NA OCENĘ 4.0	opanowanie materiału w 70%
NA OCENĘ 4.5	opanowanie materiału w 80%
NA OCENĘ 5.0	opanowanie materiału w 90%
EFEKT KSZTAŁCENIA 6	
NA OCENĘ 3.0	opanowanie materiału w 50%
NA OCENĘ 3.5	opanowanie materiału w 60%
NA OCENĘ 4.0	opanowanie materiału w 70%
NA OCENĘ 4.5	opanowanie materiału w 80%
NA OCENĘ 5.0	opanowanie materiału w 90%
EFEKT KSZTAŁCENIA 7	
NA OCENĘ 3.0	opanowanie materiału w 50%
NA OCENĘ 3.5	opanowanie materiału w 60%
NA OCENĘ 4.0	opanowanie materiału w 70%
NA OCENĘ 4.5	opanowanie materiału w 80%
NA OCENĘ 5.0	opanowanie materiału w 90%

10 MACIERZ REALIZACJI PRZEDMIOTU

EFEKT KSZTAŁCENIA	ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	NARZĘDZIA DYDAKTYCZNE	SPOSOBY OCENY
EK1	I1_W04 I1_W10	Cel 1 Cel 2 Cel 3	P1 W1 W2 W3	N1 N2	F1 P2

EFEKT KSZTAŁCENIA	ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU	CELE PRZEDMIOTU	TREŚCI PROGRAMOWE	NARZĘDZIA DYDAKTYCZNE	SPOSOBY OCENY
EK2	I1_W04 I1_W06	Cel 1 Cel 2 Cel 3	P1 W2 W3 W6	N1 N2	F1 P1 P2
EK6	I1_W02 I1_W04 I1_W06 I1_W10 I1_W11	Cel 3 Cel 4	W3 W4 W5	N1 N2	F1 P1
EK7	I1_U14 I1_K03	Cel 2 Cel 3	W2 W3 W4 W6	N1 N2	F1 P1

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

- [1] | Jaskiewicz A. — *Inżynieria oprogramowania*, Gliwice, 1997, Helion
- [2] | Sommerville J. — *Inżynieria oprogramowania - klasyka informatyki*, Warszawa, 2003, WNT
- [3] | Hamlet D., Maybee J. — *Podstawy techniczne inżynierii oprogramowania*, Warszawa, 2003, WNT
- [4] | Maciaszek L.A., Liong B.L. — *Practical Software Engineering a Case Study Approach*, Melbourne, 2005, Pearson Education Limited
- [5] | Śmiałek M. — *Zrozumieć UML 2.0*, Gliwice, 2005, Helion

LITERATURA UZUPEŁNIAJĄCA

- [1] | Brooks F.P. — *Mityczny osobomiesiąc*, Warszawa, 2000, WNT
- [2] | Hunt A., Thomas D. — *Pragmatyczny programista - od czeladnika do mistrza*, Warszawa, 2002, WNT

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr inż. Piotr Poznański (kontakt: ppozanski@pk.edu.pl)

OSOBY PROWADZĄCE PRZEDMIOT

1 dr inż. Paweł Jarosz (kontakt: pjarosz@pk.edu.pl)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)

(odpowiedzialny za przedmiot)

(dziekan)



PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....