

KARTA PRZEDMIOTU

Kod przedmiotu	0613-2INF-C31-IO	
Nazwa przedmiotu w języku	polskim	Inżynieria oprogramowania Software engineering
	angielskim	

1. USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

1.1. Kierunek studiów	Informatyka
1.2. Forma studiów	stacjonarne
1.3. Poziom studiów	studia I-stopnia
1.4. Profil studiów	ogólnoakademicki
1.5. Osoba przygotowująca kartę przedmiotu	Małgorzata Żabińska-Rakoczy
1.6. Kontakt	malgorzata.zabinska-rakoczy@ujk.edu.pl

2. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

2.1. Język wykładowy	polski
2.2. Wymagania wstępne	Bazy danych, Programowanie obiektowe

3. SZCZEGÓŁOWA CHARAKTERYSTYKA PRZEDMIOTU

3.1. Forma zajęć	wykłady, laboratorium, projekt	
3.2. Miejsce realizacji zajęć	zajęcia w pomieszczeniu dydaktycznym UJK, e-learning	
3.3. Forma zaliczenia zajęć	wykłady – egzamin, laboratorium – zaliczenie z oceną, projekt – zaliczenie.	
3.4. Metody dydaktyczne	wykład, ćwiczenia projektowe, e-learning	
3.5. Wykaz literatury	podstawowa	1. Cockburn A., „Jak pisać efektywne przypadki użycia” 2. Pressman R. S., „Praktyczne podejście do inżynierii oprogramowania”, 3. Sacha K., „Inżynieria oprogramowania”, 4. Sommerville I.,: „Inżynieria oprogramowania”, 5. Stevens, P., „ UML. Inżynieria oprogramowania”
	uzupełniająca	1. Booch G., Rumbaugh J., Jacobson I., „UML przewodnik użytkownika” 2. Larman C., „UML i wzorce projektowe” 3. Schmuller J., „UML dla każdego”, 4. Chrabski B., Zmitrowicz K., „Inżynieria wymagań w praktyce”, 5. Wrycza K. i in. „Język UML 2.0 w modelowaniu systemów informatycznych”.

4. CELE, TREŚCI I EFEKTY UCZENIA SIĘ

4.1 Cele przedmiotu

Wykład:

- C1. Poznanie pojęć inżynierii oprogramowania, elementów procesu wytwórczego i jego modeli
- C2. Poznanie elementów języka UML (Unified Modeling Language)
- C3. Poznanie typowych metodyk: zasad, notacji, sposobów modelowania tworzonego systemu
- C4. Praktyczne użycie elementów języka UML do opisu wymagań systemu oprogramowania oraz tworzenia specyfikacji projektowych
- C5. Opanowanie zasad dokumentowania prac w dziedzinie inżynierii oprogramowania

Laboratorium, projekt:

- C1. Praktyczne przeprowadzenie opisu dziedziny zadania inżynierskiego, jego dekompozycja,
- C2. Przeprowadzenie procesu analizy na przykładzie praktycznym, w szczególności opisu wymagań w UML,
- C3. Udokumentowanie procesu analizy, integracja prac zespołu projektowego,
- C4. Zastosowanie UML do procesu projektowania w inżynierii oprogramowania, tworzenie specyfikacji projektowych,
- C5. Integracja prac projektowych zespołu, dokumentowanie testów akceptacyjnych (test cases) i opracowanie założeń do implementacji zaprojektowanego systemu oprogramowania

4.2 Treści programowe (z uwzględnieniem formy zajęć)

Wykłady,

1. Pojęcia związane z projektowaniem systemów oprogramowania i istniejącymi problemami. Co to jest „dobry system” oprogramowania i jak do tego dążyć?
Klasyczny kaskadowy cykl życia/ procesu wytwórczego w inżynierii oprogramowania: etapy, fazy i czynności, ich wyniki, uczestnicy procesów, zalety i wady oraz zastosowania modelu kaskadowego.
2. Podstawowe modele procesu wytwórczego oprogramowania (zmodyfikowany model kaskadowy, prototypowanie wymagań, modele: przyrostowy, spiralny, iteracyjność i model iteracyjno-przyrostowy). Istota modeli, zalety, wady i obszary zastosowań.
3. Iteracyjność procesu inżynierii oprogramowania i podejście iteracyjno-przyrostowe, jego cechy oraz zastosowania. Etap planowania, studium wykonalności, metody szacowania kosztów projektu i czasu trwania projektu (Diagram Gantta). Wyniki etapu planowania. Kryteria oceny rozwiązań, wskaźniki oceny jakości tworzonego systemu oprogramowania.
4. Analiza systemowa. Opis dziedziny problemu, obszaru modelowania, zakresu odpowiedzialności systemu. Analiza potrzeb użytkowników, sposoby ich pozyskiwania, prezentacji, weryfikacja. Elementy inżynierii wymagań: czynności, tworzone artefakty. Opis biznesowy potrzeb użytkowników.
5. Wymagania funkcjonalne i niefunkcjonalne (kategorie). Opis wstępny wymagań przy pomocy „user stories”. Przykłady użycia: od nieformalnych opisów słownych, poprzez scenariusze „casual”, do pełnego opisu tabelarycznego („fully dressed”).
6. Wizualne modele wymagań – diagramy przypadków użycia Use Case (UML) i ich zastosowanie. Złożoność wymagań, metody „opanowania” złożoności. Wybrane elementy modelowania przy pomocy UML (Activity Diagrams – diagramy aktywności).
7. Konceptyjne i techniczne projektowanie systemu. Specyfikacje projektowe i ich realizacja. Projektowanie interfejsu. Rezultaty analizy i projektowania, dokumentowanie prac projektowych. Wspomaganie prac projektowych. Rola analityka systemowego i projektanta.
8. Testowanie, rodzaje testów, przypadki testowe. Elementy zarządzania ryzykiem. Jakość oprogramowania, standardy, zarządzanie jakością oprogramowania.

Laboratorium i projekty własne,

1. Narzędzia do modelowania systemu oprogramowania i ich cechy. Dyskusja tematów projektów własnych, omówienie instrukcji w formie chronologicznej listy tworzonych artefaktów (kolejnych kroków procesu wytwórczego) .
2. Opis dziedziny problemu z punktu widzenia potrzeb biznesowych, wyznaczanie zakresu odpowiedzialności systemu, dekompozycja dziedziny. Ćwiczenie użyteczności narzędzia (Visual Paradigm online)
3. Wstępny opis wymagań jako „user stories”, proces uszczegółowienia w wydzielonych obszarach dziedzinowych odpowiadających tematowi projektu. Opis szczególnych wymagań i parametrów jakościowych, weryfikacja ról; rozwinięty opis User Stories.
4. Prezentacja wizji różnych typów wymagań związanych z wydzielonymi obszarami dziedzinowymi: wymagania funkcjonalne, wymagania na dane, wymagania na interfejs. Weryfikacja zaprezentowanej wizji wymagań i dyskusja wymagań niefunkcjonalnych (wg trzech grup).
5. Tworzenie koncepcji wymagań funkcjonalnych: propozycja systemowych przypadków użycia (Use Cases) jako rozwinięcie User Stories, wg szablonu A. Cockburne’a. Tworzenie modelu wymagań przy pomocy diagramów przypadków użycia (Use Case Diagrams) wg koncepcji dekompozycji.
6. Modelowanie działania przy pomocy diagramów aktywności (Activity Diagrams) w wydzielonych obszarach dziedzinowych. Weryfikacja i integracja modeli wymagań i działania.
7. Projektowanie bazy danych systemu jako rozwinięcie opisu wymagań na dane. Projektowanie interfejsu: struktury, makiet, integracja i ustandaryzowanie w ramach dziedziny (projektu)
8. Opracowywanie przypadków testowych (Test Cases) na podstawie Use Cases.
9. Ustalanie założeń do implementacji, w tym analiza narzędzi. Analiza rozwiązań konkurencyjnych
10. Prezentacja projektów. Wzajemne recenzowanie projektów na podstawie wytworzonej (wg instrukcji) dokumentacji projektowej

4.3 Przedmiotowe efekty uczenia się

Efekt	Student, który zaliczył przedmiot	Odniesienie do kierunkowych efektów uczenia się
w zakresie WIEDZY:		
W01	zna pojęcia i modele procesu wytwórczego w inżynierii oprogramowania	INF1A_W11
W02	zna elementy języka do modelowania (UML)	INF1A_W11 INF1A_W12
W03	definiuje wymagania dotyczące tworzonego systemu i objaśnia kolejne kroki analizy i	INF1A_W11

	projektowania	INF1A_W14 INF1A_W15
w zakresie UMIEJĘTNOŚCI:		
U01	formułuje opis dziedzinowy, zakres odpowiedzialności systemu, wymagania (biznesowe) odnośnie projektowanego systemu	INF1A_U16 INF1A_U19
U02	umie zastosować elementy języka do modelowania (UML) i graficznie przedstawić wymagania funkcjonalne stosując język UML oraz opisać je scenariuszowo wg wzorców	INF1A_U14 INF1A_U16
U03	opracowuje dokumentację projektową jako wynik analizy i projektowania; projektuje część funkcjonalną i bazodanową systemu oprogramowania oraz interfejs graficzny (GUI)	INF1A_U07
U04	jest świadomy potrzeby komunikacji pomiędzy uczestnikami zespołu projektowego	INF1A_U22
U05	wykazuje aktywność w ramach współpracy w zespole oraz podziału prac projektowych	INF1A_U22
U06	jest wrażliwy na braki w dokumentacji projektowej (wzajemne recenzowanie)	INF1A_U22
w zakresie KOMPETENCJI SPOŁECZNYCH:		
K01	potrafi określić priorytety służące realizacji zadania i zaplanować pracę własną i w zespole	INF1A_K01

4.4 Sposoby weryfikacji osiągnięcia przedmiotowych efektów uczenia się																		
Efekty przedmiotowe (symbol)	Sposób weryfikacji (+/-)																	
	Egzamin pisemny			Projekt			Aktywność na zajęciach			Praca własna			Praca w grupie					
	Forma zajęć			Forma zajęć			Forma zajęć			Forma zajęć			Forma zajęć					
	W	L	P	W	L	P	W	L	P	W	L	P	W	L	P			
W01	+		+		+	+	+	+			+	+		+	+			
W02	+		+		+	+	+	+			+	+		+	+			
W03	+		+		+	+	+	+			+	+		+	+			
U01		+	+			+			+			+			+			
U02		+	+			+			+			+			+			
U03		+	+			+			+			+			+			
U04						+			+			+			+			
U05						+			+			+			+			
U06						+			+			+			+			
K01					+	+		+	+		+	+		+	+			

4.5 Kryteria oceny stopnia osiągnięcia efektów uczenia się		
Forma zajęć	Ocena	Kryterium oceny
wykład (W)	3	co najmniej 50% i nie więcej niż 60% łącznej liczby punktów możliwych do uzyskania
	3,5	ponad 60% i nie więcej niż 70% łącznej liczby punktów możliwych do uzyskania
	4	ponad 70% i nie więcej niż 80% łącznej liczby punktów możliwych do uzyskania
	4,5	ponad 80% i nie więcej niż 90% łącznej liczby punktów możliwych do uzyskania
	5	ponad 90% liczby punktów możliwych do uzyskania
laboratorium (L)	3	co najmniej 50% i nie więcej niż 60% łącznej liczby punktów możliwych do uzyskania
	3,5	ponad 60% i nie więcej niż 70% łącznej liczby punktów możliwych do uzyskania
	4	ponad 70% i nie więcej niż 80% łącznej liczby punktów możliwych do uzyskania
	4,5	ponad 80% i nie więcej niż 90% łącznej liczby punktów możliwych do uzyskania
	5	ponad 90% liczby punktów możliwych do uzyskania
Projekt (P)	3	co najmniej 50% i nie więcej niż 60% łącznej liczby punktów możliwych do uzyskania
	3,5	ponad 60% i nie więcej niż 70% łącznej liczby punktów możliwych do uzyskania
	4	ponad 70% i nie więcej niż 80% łącznej liczby punktów możliwych do uzyskania
	4,5	ponad 80% i nie więcej niż 90% łącznej liczby punktów możliwych do uzyskania
	5	ponad 90% liczby punktów możliwych do uzyskania

5 BILANS PUNKTÓW ECTS – NAKŁAD PRACY STUDENTA

Kategoria	Obciążenie studenta	
	Studia stacjonarne	Studia niestacjonarne
LICZBA GODZIN REALIZOWANYCH PRZY BEZPOŚREDNIM UDZIALE NAUCZYCIELA /GODZINY KONTAKTOWE/	90	
Udział w wykładach*	30	
Udział w laboratoriach*	30	
Realizacja projektu własnego	30	
SAMODZIELNA PRACA STUDENTA /GODZINY NIEKONTAKTOWE/	60	
Przygotowanie do laboratorium*	20	
Przygotowanie do egzaminu	25	
Zebrań materiałów do projektu, kwerenda internetowa*	10	
Opracowanie prezentacji multimedialnej*	5	
ŁĄCZNA LICZBA GODZIN	150	
PUNKTY ECTS za przedmiot	6	

*niepotrzebne usunąć

Przyjmuję do realizacji (data i czytelne podpisy osób prowadzących przedmiot w danym roku akademickim)

.....