KARTA PRZEDMIOTU

I. Dane podstawowe

Nazwa przedmiotu	Inżynieria oprogramowania
Nazwa przedmiotu w języku angielskim	Software engineering
Kierunek studiów	Informatyka
Poziom studiów (I, II, jednolite magisterskie)	I
Forma studiów (stacjonarne, niestacjonarne)	Stacjonarne
Dyscyplina	Informatyka
Język wykładowy	Polski

Koordynator przedmiotu/osoba odpowiedzialna	Marcin Płonkowski

Forma zajęć *(katalog zamknięty ze słownika)*	Liczba godzin	semestr	Punkty ECTS
wykład	30	5	5
konwersatorium			
ćwiczenia	30	5	
laboratorium			
warsztaty			
seminarium			
proseminarium			
lektorat			
praktyki			
zajęcia terenowe			
pracownia dyplomowa			
translatorium			
wizyta studyjna			

Wymagania wstępne	Znajomość programowania strukturalnego i obiektowego, język angielski

II. Cele kształcenia dla przedmiotu

Podniesienie poziomu wiedzy studentów w zakresie inżynierii oprogramowania
Przedstawienie i szczegółowe omówienie wszystkich aspektów tworzenia oprogramowania od początkowej fazy specyfikacji aż do jego pielęgnacji po dacie rozpoczęcia użytkowania
Zapoznanie studentów i wyrobienie u nich umiejętności pracy zgodnie z metodykami strukturalną, obiektową i zwinną

III. Efekty uczenia się dla przedmiotu wraz z odniesieniem do efektów kierunkowych

Symbol	Opis efektu przedmiotowego	Odniesienie do efektu kierunkowego
WIEDZA		
W_01	Student wie, czym jest inżynieria oprogramowania, proces tworzenia oprogramowania, zarządzanie przedsięwzięciami	K_W04, K_W06, K_W08
W_02	Student wie, jak powinny być stawiane wymagania oprogramowaniu, jak wygląda proces inżynierii wymagań, modelowanie systemu, prototypowanie oprogramowania, weryfikacja, testowanie i odbiór zatwierdzonego oprogramowania	K_W04, K_W06
W_03	Student wie jakie są metody zarządzanie personelem, zarządzania jakością, szacowania oprogramowania, ulepszania oprogramowania	K_W06, K_W08
UMIEJĘTNOŚCI		
U_01	Student umie konstruować wymagania niefunkcjonalne i sporządzać specyfikację oprogramowania	K_U02, K_U04, K_U13, K_U14, K_U23
U_02	Student potrafi używać diagramów opisu struktury i zachowań programu	K_U02, K_U04, K_U13, K_U14, K_U23
U_03	Student potrafi w stopniu podstawowym korzystać z języka UML	K_U02, K_U04, K_U13, K_U14, K_U23
U_04	Student potrafi opracować plan przedsięwzięcia dotyczącego budowy oprogramowania	K_U02, K_U04, K_U14, K_U23
U_05	Student potrafi kontrolować i zarządzać wersjami tworzonego oprogramowania oraz stosować się do zasad obowiązujących programistów podczas pracy w zespole	K_U02, K_U04, K_U13, K_U14, K_U23, K_U29, K_U30
KOMPETENCJE SPOŁECZNE		
K_01	Student jest otwarty na złożoność problemów, z którymi może spotkać się w życiu	K_K01, K_K06
K_02	Student umiejętnie rozwiązuje problemy inżynierii oprogramowania stosując poznane metody oraz obiektywnie ocenia uzyskane wyniki	K_K01, K_K03
K_03	Student potrafi pracować zarówno indywidualnie, jak i zespołowo, właściwie planując pracę swoją i zespołu w kontekście postawionego celu	K_K02, K_K04, K_K07

IV. Opis przedmiotu/ treści programowe

1 Wprowadzenie 1
2 Procesy wytwarzania oprogramowania 2
3 Inżynieria wymagań 2
4 Metody strukturalne 3
5 Metody obiektowe i podstawy UML 5
6 Jakość kodu, inspekcje kodu 2
7 Testowanie 3

8 Dokumentacja użytkowa 1 9 Konserwacja 2 10 Systemy krytyczne 2 11 Metody formalne 1 12 Wzorce projektowe 4 13 Programowanie ekstremalne i Scrum 2

V. Metody realizacji i weryfikacji efektów uczenia się

Symbol efektu	Metody dydaktyczne *(lista wyboru)*	Metody weryfikacji *(lista wyboru)*	Sposoby dokumentacji *(lista wyboru)*
WIEDZA			
W_01	Wykład konwencjonalny/Wykład problemowy	Egzamin	Test / Sprawdzian pisemny
W_02	Wykład konwencjonalny/Wykład problemowy	Egzamin	Test / Sprawdzian pisemny
W_03	Wykład konwencjonalny/Wykład problemowy	Egzamin	Test / Sprawdzian pisemny
UMIEJĘTNOŚCI			
U_01	Ćwiczenia praktyczne	Kolokwium	Sprawdzian pisemny / Uzupełnione i ocenione kolokwium
U_02	Ćwiczenia praktyczne	Kolokwium	Sprawdzian pisemny / Uzupełnione i ocenione kolokwium
U_03	Ćwiczenia praktyczne	Kolokwium	Sprawdzian pisemny / Uzupełnione i ocenione kolokwium
U_04	Ćwiczenia praktyczne	Kolokwium	Sprawdzian pisemny / Uzupełnione i ocenione kolokwium
U_05	Ćwiczenia praktyczne	Kolokwium	Sprawdzian pisemny / Uzupełnione i ocenione kolokwium
KOMPETENCJE SPOŁECZNE			
K_01	Dyskusja	Kolokwium	Sprawdzian pisemny / Uzupełnione i ocenione kolokwium
K_02	Dyskusja	Kolokwium	Sprawdzian pisemny / Uzupełnione i ocenione kolokwium
K_03	Dyskusja	Kolokwium	Sprawdzian pisemny / Uzupełnione i ocenione kolokwium

VI. Kryteria oceny, wagi

Zaliczenie wykładu: Egzamin – 100%
Zaliczenie ćwiczeń: kolokwium – 80%, aktywność – 20%

(5.0): 90 – 100%,

(4.5): 80 – 89%,

(4.0): 70 – 79%,

(3.5): 60 – 69%,

(3.0): 50 – 59%,

(2.0): < 50%

Obciążenie pracą studenta

Forma aktywności studenta	Liczba godzin
Liczba godzin kontaktowych z nauczycielem	**75**
Liczba godzin indywidualnej pracy studenta	**65**

VII. Literatura

Literatura podstawowa
1. K. Sacha, Inżynieria oprogramowania, PWN, Warszawa 2010 2. A. Jaszkiewicz, Inżynieria oprogramowania, Helion, Gliwice 1997 3. I. Sommerville, Inżynieria oprogramowania, WNT, Warszawa 2003
Literatura uzupełniająca
1. L. Miękina, Inżynieria oprogramowania, Wydawnictwo AGH, Kraków 2009 2. P. Stevens, UML. Inżynieria oprogramowania, Helion, Gliwice 2007 3. J. Myers, C. Sandler, T. Badgett, T. M. Thomas, Sztuka testowania oprogramowania, Helion, Gliwice 2005 4. R. C. Martin. Czysty kod. Podręcznik dobrego programisty, Helion, Gliwice 2010 5. Bruegge B., Dutoit A. H., Inżynieria oprogramowania w ujęciu obiektowym. UML, wzorce projektowe i Java, Helion, Gliwice 2011 6. http://wazniak.mimuw.edu.pl