

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Inżynieria oprogramowania		Kod 1010334461010330109
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) (brak)	Rok / Semestr 3 / 6
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 16 Ćwiczenia: - Laboratoria: - Projekty/seminaria: 8		Liczba punktów 4
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (brak)		(ogólnouczelniany, z innego kierunku) (brak)
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne		Podział ECTS (liczba i %) 4 100%
Odpowiedzialny za przedmiot / wykładowca:		
dr hab. inż. Barbara Begier email: Barbara.Begier@put.poznan.pl tel. 665-3724 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	K_W05: ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie podstawowych konstrukcji programistycznych, implementacji algorytmów, paradygmatów i stylów programowania, metod weryfikacji poprawności programów, języków formalnych, kompilatorów, platform. Ma uporządkowaną i podbudowaną metodologicznie wiedzę w zakresie inżynierii oprogramowania (pierwszej części tego przedmiotu).
2	Umiejętności:	K_U01: potrafi pozyskiwać informacje z literatury, baz danych i innych źródeł; potrafi integrować uzyskane informacje, dokonywać ich interpretacji, a także wyciągać wnioski oraz formułować i uzasadniać opinie.
3	Kompetencje społeczne	K_K02: ma świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżyniera-informatyka i związaną z tym odpowiedzialność za podejmowane decyzje. K_K01: rozumie potrzebę i zna możliwości ciągłego dokształcania się (studia drugiego i trzeciego stopnia, studia podyplomowe, kursy), podnoszenia kompetencji językowych, zawodowych, osobistych i społecznych.
Cel przedmiotu:		
Celem drugiej części przedmiotu jest zapoznanie studentów z metodami zapewniania i oceny jakości oprogramowania, poznanie zwrotnych metodyk wytwarzania oprogramowania, a podczas zajęć projektowych pogłębienie znajomości standardu UML 2.0.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. Ma uporządkowaną i podbudowaną metodologicznie wiedzę w zakresie inżynierii oprogramowania. - [K_W12] 2. Orientuje się w obecnym stanie oraz najnowszych trendach rozwojowych informatyki. - [K_W19]		
Umiejętności:		
1. Potrafi sformułować wymagania, opracować model obiektowy oraz ocenić prosty system informatyczny, uwzględniając realizowane funkcje i powiązania między elementami składowymi. - [K_U16] 2. Potrafi opracować dokumentację dotyczącą realizacji zadania inżynierskiego i przygotować tekst zawierający omówienie wyników realizacji tego zadania. Umie opracować plan testów oraz drzewo jakości oprogramowania. - [K_U03]		
Kompetencje społeczne:		
1. Ma świadomość ważności dokładnego wykonania oprogramowania, zachowania standardów notacyjnych, przestrzegania poprawności językowej i terminowego oddania prac. - [K_K07] 2. Ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania. - [K_K04]		

Sposoby sprawdzenia efektów kształcenia		
<p>Treści prezentowane podczas wykładu są przedmiotem egzaminu pisemnego. Student wykazuje umiejętność tworzenia podstawowych konstrukcji w UML, znajomość zasad i praktyk postępowania w metodach twardych i zwinnych (umie porównać wybrane metodyki) oraz zna kryteria i miary oceny jakości oprogramowania.</p> <p>Zaliczenie zajęć projektowych odbywa się na podstawie ocen cząstkowych, wystawianych oddzielnie za każdy opracowany dokument oraz diagram UML.</p>		
Treści programowe		
<p>Wykład. Jakość wyrobu programowego i jej charakterystyki według standardów ISO 9126 oraz ISO 25010. Polityki jakości w procesie wytwarzania oprogramowania. Planowanie testowania. Wspólna charakterystyka metodyk zwinnych, zasady wyrażone w Agile Manifesto. Przegląd metodyk zwinnych: XP (eXtreme Programming), TDD (Test Driven Development), AMDD (Agile Model Driven Development), FDD (Feature Driven Development), BDD (Behavior Driven Development), Scrum. Rola czynnika ludzkiego w produkcji oprogramowania. Satysfakcja użytkownika z wyrobu programowego, model EUCS (End User Computing Satisfaction). Niektóre aspekty etyczne w inżynierii oprogramowania.</p> <p>Projekt. Opracowanie modelu zachowań przyszłego systemu. Opracowanie planu testowania oraz drzewa jakości tworzonego oprogramowania.</p>		
Literatura podstawowa:		
<ol style="list-style-type: none"> 1. Martin R., Martin M., Agile. Programowanie zwinne. Zasady, wzorce i praktyki zwinnego wytwarzania oprogramowania w C?, Helion, Gliwice 2008. 2. Wrycza St., Marcinkowski B., Wyrzykowski K., Język UML 2.0 w modelowaniu systemów informatycznych, Helion, Gliwice 2005. 3. Dokumenty z Internetu, aktualne w zakresie omawianej tematyki. 		
Literatura uzupełniająca:		
<ol style="list-style-type: none"> 1. Begier B., Inżynieria oprogramowania - problematyka jakości, Wydawnictwo Politechniki Pozn., Poznań 1999. 2. Hnatkowska B., Huzar Z., Inżynieria oprogramowania. Metody wytwarzania i wybrane zagadnienia, PWN, Warszawa 2008. 3. Pitone D., Pitman N., UML 2.0. Almanach, Helion, Gliwice 2007. 4. Subieta K., Wprowadzenie do inżynierii oprogramowania, Wydawnictwo PJWSTK, Warszawa 2002. 		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. uczestnictwo w wykładach	16	
2. uczestnictwo w zajęciach projektowych	8	
3. przygotowanie i opracowanie projektu	26	
4. egzamin, konsultacje	10	
5. przygotowanie do egzaminu	20	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	80	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	34	1
Zajęcia o charakterze praktycznym	34	1