

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Inżynieria oprogramowania		Kod 1010334561010330109
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 inż. Adam Meissner email: Adam.Meissner@put.poznan.pl tel. 61 665 37 24 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student ma uporządkowaną i podbudowaną teoretycznie wiedzę z zakresu podstaw inżynierii oprogramowania obejmującą elementy modelowania i realizacji systemów informatycznych.
2	Umiejętności:	Student potrafi pozyskiwać informacje z literatury, baz danych i innych źródeł; potrafi integrować uzyskane informacje, dokonywać ich interpretacji, a także wyciągać wnioski oraz formułować opinie; potrafi pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zadania; potrafi opracować i zrealizować harmonogram prac; posługuje się językiem angielskim w stopniu wystarczającym do porozumiewania się oraz do czytania ze zrozumieniem wskazanej literatury przedmiotowej.
3	Kompetencje społeczne	Student rozumie potrzebę i zna możliwości ciągłego dokształcania się, podnoszenia kompetencji językowych, zawodowych, osobistych i społecznych; ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania.
Cel przedmiotu:		
w ramach drugiej części przedmiotu studentów zapoznaje się z wybranymi metodami modelowania i wytwarzania systemów informatycznych oraz z metodami walidacji, weryfikowania i testowania ich poprawności.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. Student ma uporządkowaną i podbudowaną metodologicznie wiedzę w zakresie inżynierii oprogramowania - [K_W12] 2. Student orientuje się w obecnym stanie oraz najnowszych trendach rozwojowych informatyki - [K_W19]		
Umiejętności:		
1. Student potrafi opracować dokumentację dotyczącą realizacji zadania inżynierskiego i przygotować tekst zawierający omówienie wyników realizacji tego zadania - [K_U03] 2. Student potrafi sformułować wymagania, opracować model obiektowy oraz ocenić prosty system informatyczny, uwzględniając realizowane funkcje i powiązania między elementami składowymi - [K_U16]		
Kompetencje społeczne:		
1. Student ma świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżyniera-informatyka i związaną z tym odpowiedzialność za podejmowane decyzje - [K_K02] 2. Student ma świadomość ważności dokładnego wykonania projektu, zachowania standardów notacyjnych, przestrzegania poprawności językowej i terminowego oddania prac - [K_K07]		
Sposoby sprawdzenia efektów kształcenia		

Wykład: egzamin pisemny. Projekt: ocena projektu systemu informatycznego wykonanego w standardzie UML.		
Treści programowe		
<p>Wykład. Przegląd modeli cyklu rozwojowego oprogramowania. Wytwarzanie oprogramowania w ujęciu systematycznym, standard ISO 9000, model CMM. Programowanie zwinne i programowanie ekstremalne. Metoda Scrum. Wzorce projektowe. "Zapachy" kodu. Metody walidacji, weryfikacji i testowania oprogramowania.</p> <p>Aktualizacja 2017: metoda Scrum.</p> <p>Projekt. Kontynuacja prac nad modelem wskazanego systemu informatycznego, prowadzonych w ramach pierwszej części przedmiotu. Model wykonuje się w standardzie UML, a jego nowymi elementami są diagramy czynności, diagramy kolejności oraz diagram wdrożenia.</p> <p>Zastosowane metody kształcenia:</p> <ul style="list-style-type: none"> - wykład z prezentacją slajdów uzupełniany przykładami podawanymi na tablicy - projekt - szczegółowe recenzowanie dokumentacji projektowych przez prowadzącego połączone z omówieniem typowych błędów oraz propozycjami ich usuwania. 		
Literatura podstawowa:		
<ol style="list-style-type: none"> 1. Paulish D.J., Inżynieria oprogramowania. Zarządzanie architekturocentrycznym procesem tworzenia oprogramowania. Przewodnik praktyczny, WNT, Warszawa, 2007 2. Schwaber K., Sutherland J., The Scrum Guide TM. The Definitive Guide to Scrum: The Rules of the Game, July 2016, http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Polish.pdf 3. Shore S., Warden S., Agile Development. Filozofia programowania zwinnego, Wyd. Helion, Gliwice, 2008 4. Zmitrowicz K., Jakość projektów informatycznych. Rozwój i testowanie oprogramowania, Wyd. Helion, Gliwice, 2015 		
Literatura uzupełniająca:		
<ol style="list-style-type: none"> 1. Jeffries R., Programowanie ekstremalne w C#, PWN, Warszawa, 2005 2. Rad N.K., Turley F., The Scrum Master Training Manual. A Guide to Passing the Professional Scrum Master (PSM) Exam, Management Plaza, 2013, https://mplaza.pm/downloads/Scrum%20Training%20Manual.pdf 3. Sutherland J., Jeff Sutherland's Scrum Handbook, Scrum Training Institute Press, 2010, http://www.ugrad.cs.ubc.ca/~cs310/2014W1/slides/Sutherland_Scrum_Handbook.pdf 		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. Wykłady	16	
2. Zajęcia projektowe	8	
3. Przygotowanie do zajęć projektowych	20	
4. Przygotowanie do egzaminu	15	
5. Konsultacje i egzamin	5	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	64	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	29	2
Zajęcia o charakterze praktycznym	28	2