

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Inżynieria wymagań		Kod 1010515331010500345
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 2 / 3
Ścieżka obieralności/specjalność Zaawansowane technologie internetowe	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obieralny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 16 Ćwiczenia: - Laboratoria: - Projekty/seminaria: 16		Liczba punktów 4
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (ogólnouczelniany, z innego kierunku) kierunkowy z danego kierunku		
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne		Podział ECTS (liczba i %) 4 100% 4 100%
Odpowiedzialny za przedmiot / wykładowca: dr inż. Adam Wojciechowski email: Adam.Wojciechowski@put.poznan.pl tel. 61-6652983 Wydział Informatyki ul. Piotrowo 3 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Znajomość dowolnego języka programowania i praktyczne doświadczenia w konstruowaniu aplikacji.
2	Umiejętności:	Podstawowa umiejętność opisywania architektury systemu informatycznego przy pomocy diagramów UML.
3	Kompetencje społeczne	Umiejętność podziału zadań i odpowiedzialności na poziomie wymaganym podczas realizacji pracy inżynierskiej w zespole kilkuosobowym.
Cel przedmiotu: Przekazanie studentom podstawowej wiedzy w zakresie znajomości problematyki definiowania, dokumentowania i aktualizacji specyfikacji wymagań dla systemów informatycznych. Omówione są tzw. dobre praktyki inżynierii wymagań wg Sommerville'a i Sawyera, standard specyfikowania wymagań IEEE 830, problematyka zarządzania wymaganiami i ryzykiem w projektach informatycznych, metodyki inspekcji wymagań oraz automatyzacja procesu testowania systemów komputerowych. Zarządzanie wymaganiami w tzw. lekkich metodykach rozwoju systemów informatycznych. W czasie semestralnej pracy studenci zaangażowani są w pierwszej kolejności w analizę i specyfikację wymagań dla wybranych systemów informatycznych, którymi mogą być np. systemy i aplikacje budowane w ramach realizacji prac magisterskich. Kolejnym ćwiczeniem jest ocena kompletności specyfikacji wymagań i stopnia zrozumienia zakresu implementacji na podstawie specyfikacji sporządzonej przez inną osobę. Podczas ćwiczeń uczestnicy zajęć biorą udział w grach dydaktycznych ?Team building? oraz ?eXtreme89?, których celem jest zaakcentowanie znaczenia dobrej współpracy i komunikacji pomiędzy klientem i zespołem programistów.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. Student ma uporządkowaną, wiedzę ogólną w zakresie: metodyk gromadzenia i dokumentowania wymagań dla budowy systemów informatycznych oraz zarządzania projektem informatycznym. - [K2st_W2]		
2. Student ma wiedzę szczegółową wiedzę związaną zagadnieniami takimi jak: metody pozyskiwania, gromadzenia i aktualizacji wymagań dla systemów informatycznych oraz zarządzanie ryzykiem. - [K2st_W3]		
3. Student ma wiedzę o trendach i tendencjach w zakresie gromadzenia i dokumentowania wymagań w różnych metodykach realizacji projektów informatycznych. - [K2st_W4]		
4. Student zna tzw. tradycyjne i lekkie metodyki budowy systemów informatycznych. - [K2st_W6]		
Umiejętności:		

1. Student potrafi planować użycie i efektywnie wykorzystać systemy informatyczne w procesie zarządzania wymaganiami. - [K2st_U3]
2. Student potrafi skorzystać z metod statystycznych dla planowania złożoności systemów informatycznych - [K2st_U4]
3. Student potrafi integrować wiedzę z informatyki: programowanie, wspomaganie podejmowania decyzji, jak i innych dyscyplin naukowych takich jak ekonomia w procesie gromadzenia i analizy wymagań. - [K2st_U5]
4. Student potrafi przeprowadzić krytyczną ocenę narzędzi wspomagających proces zarządzania wymaganiami. - [K2st_U6]
5. Student potrafi stworzyć model obiektowy prostego systemu (np. w języku UML). - [K2st_U10]
6. Student potrafi sformułować specyfikację funkcjonalną w formie przypadków użycia. - [K2st_U11]
Kompetencje społeczne:
1. Student docenia możliwość korzystania z nowej wiedzy i doświadczeń korzystając m.in. z referatów konferencyjnych dostępnych w Internecie. - [K2st_K1]
2. Student potrafi stosować tzw. dobre praktyki w procesie zarządzania wymaganiami i zarządzania ryzykiem. - [K2st_K2]

Sposoby sprawdzenia efektów kształcenia
Ocena formująca: a) w zakresie wykładów: na podstawie udziału w dyskusji i odpowiedzi na pytania dotyczące materiału omawianego na wykładach, b) w zakresie laboratoriów: na podstawie oceny bieżącego postępu realizacji zadań koncepcyjnych i implementacyjnych. Ocena podsumowująca: a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez: ocenę wiedzy i umiejętności oraz innowacyjności zawartych w pracach realizowanych indywidualnie w ramach laboratoriów i w formie pracy domowej, możliwy jest test z zakresu znajomości notacji UML, zasad pozyskiwania i specyfikacji wymagań oraz zarządzania projektem informatycznym. b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez: ocenę wiedzy i umiejętności związanych z tematyką w formie zadań projektowych polegających na: udziale w grach dydaktycznych, oraz realizacji ćwiczeń zleczanych podczas zajęć i w ramach pracy domowej.. ocenę i obronę przez studentów sprawozdania-specyfikacji wymagań dla systemu informatycznego. Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za uwagi związane z udoskonaleniem materiałów dydaktycznych. Studenci mogą również dodatkowo podnieść swoją ocenę proponując ciekawe zagadnienia warte omówienia podczas wykładów lub przygotowując krótką prezentację stanowiącą rozwinięcie lub uzupełnienie zaproponowanego tematu.
Treści programowe
Program wykładu obejmuje m.in. następujące zagadnienia: Problematyka pracy zespołowej (klient, analityk i programista) i zarządzania projektem informatycznym. Metodyki pozyskiwania i gromadzenia wymagań dla systemów informatycznych. Notacja UML. Narzędzia wspomagające proces zarządzania wymaganiami (IBM Rational Requisite Pro) oraz model Sommerville'a-Sawyer'a jako zbiór dobrych praktyk w zakresie zarządzania dokumentem specyfikacji wymagań oraz metoda oceny dojrzałości firm software'owych w zakresie zarządzania wymaganiami. Standard redagowania specyfikacji wymagań IEEE830. Problematyka zarządzania wymaganiami w lekkich metodykach budowy oprogramowania. Inspekcja wymagań oraz automatyzacja testowania. Model CMMI. Prototypowanie na papierze. Zajęcia laboratoryjne prowadzone są w formie szesnastu godzin ćwiczeń, odbywających się w laboratorium, poprzedzonych 2-godzinną sesją instruktazową na początku semestru. Studenci realizują zadania indywidualnie oraz w małych zespołach (2 osoby) podczas zajęć oraz w ramach pracy domowej polegające na opracowaniu specyfikacji wymagań dla wybranego systemu informatycznego. Systemem tym może być np. aplikacja budowana przez studenta w ramach pracy dyplomowej. Podczas zajęć, ciekawym ćwiczeniem jest sesja prototypowania na papierze, która pozwala na emocjonalne zaangażowanie w zespołowe projektowanie interfejsu dla np. gra komputerowej. Opracowane interfejsy są prezentowane na forum uczestników zajęć i poddawane krytycznej ocenie. Cześć wymienionych wyżej treści programowych realizowana jest w ramach pracy własnej studenta.

Literatura podstawowa:

1. Sawyer, P. ; Sommerville, I. ; Viller, S., Capturing the benefits of requirements engineering, IEEE Software, March-April 1999, Vol.16(2), pp.78-85
2. Capability Maturity Model Integration, Version 1.1, Staged Representation, CMU/SEI-2002-TR-004, Carnegie Mellon University, 2002, <http://www.sei.cmu.edu/cmmi/products/v1.1ipppd-staged.pdf>, January 11,2002.
3. K. Beck, Extreme Programming Explained: Embrace Change. Addison-Wesley, Boston, 2000.
4. A.Jaszkievicz, Inżynieria oprogramowania, Helion 1997
5. M. Fowler, K. Scott, UML w kropelce, Oficyna Wydawnicza LTP, Warszawa 2002.
6. J.Nawrocki, M.Jasiński, B.Walter, A.Wojciechowski, Extreme Programming Modified: Embrace Requirements Engineering Practices, 10th IEEE Joint International Requirements Engineering Conference, RE'02, Essen (Germany), 2002, w: IEEE Press, Los Alamitos, 303-310.
7. Adam Wojciechowski, J.Nawrocki, G.Jachimko, L.Gawron, M.Jagielski, B.Walter, Experimental Evaluation of Two Approaches to Software Artifacts Review, Foundations of Computing and Decision Sciences, vol. 27 (2002), 249 - 259

Literatura uzupełniająca:

1. I. Sommerville, P. Sawyer, Requirements Engineering: Good Practice Guide, John Wiley & Sons, Chichester, 1997.
2. R. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, New York, 4th edition, 1997.
3. L. Maciaszek, Requirements Analysis and System Design, Addison-Wesley, 2005
4. J. Górski (red.), Inżynieria oprogramowania w projekcie informatycznym, Wydawnictwo MIKOM, Warszawa 1999.

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. udział w zajęciach laboratoryjnych / projekcie : 16 x 1 godz.	16
2. udział w konsultacjach, które mogą mieć charakter zespołowy, związanych z realizacją projektu, np. pozyskanie wymagań w przypadku opracowywania specyfikacji wymagań	2 10
3. prowadzenie sesji pozyskiwania wymagań oraz inspekcji specyfikacji wymagań.	20
4. Opracowanie specyfikacji wymagań (dokończenie w domu zadań rozpoczętych podczas zajęć laboratoryjnych) oraz przypadków testowych Praca częściowo zdalnie konsultowana przez nauczyciela.	16 25
5. udział w wykładach: 16 x 1 godz	6
6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 250 stron	
7. przygotowanie materiałów i prezentacja z sesji prototypowania na papierze	

Obciążenie pracą studenta

forma aktywności	godzin	ECTS
Łączny nakład pracy	95	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	34	2
Zajęcia o charakterze praktycznym	54	2