

(pieczęć wydziału)

KARTA PRZEDMIOTU

Z1-PU7

WYDANIE N1

Strona 1 z 4

1. Nazwa przedmiotu: ANALIZA I PROJEKTOWANIE SYSTEMÓW INFORMATYCZNYCH		2. Kod przedmiotu:		
3. Karta przedmiotu ważna od roku akademickiego: 2012				
4. Forma kształcenia: studia drugiego stopnia				
5. Forma studiów: studia stacjonarne				
6. Kierunek studiów: INFORMATYKA, WYDZIAŁ AEI				
7. Profil studiów: ogólnoakademicki				
8. Specjalność:				
9. Semestr: 1				
10. Jednostka prowadząca przedmiot: Instytut Informatyki, RAU2				
11. Prowadzący przedmiot: dr inż. Piotr Bajerski				
12. Przynależność do grupy przedmiotów: przedmioty wspólne				
13. Status przedmiotu: obowiązkowy				
14. Język prowadzenia zajęć: polski				
15. Przedmioty wprowadzające oraz wymagania wstępne: Programowanie Komputerów, Inżynieria Oprogramowania, Bazy Danych. Zakłada się, że przed rozpoczęciem nauki niniejszego przedmiotu student posiada podstawowe przygotowanie w zakresie programowania, inżynierii oprogramowania oraz baz danych.				
16. Cel przedmiotu: Celem wykładu jest przekazanie studentom wiadomości w zakresie modeli procesów wytwarzania oprogramowania, analizowania cech systemów informatycznych oraz konstruowania modeli systemów informatycznych i umiejętnego posługiwania się nimi. Szczególny nacisk kładziony jest na stosowanie języka UML oraz wykorzystywanie wzorców projektowych i architektonicznych. Celem ćwiczeń laboratoryjnych jest nabycie przez studentów umiejętności w zakresie specyfikacji i zarządzania wymaganiami, tworzenia modeli w języku UML i ich transformacji w kod oraz określenia, zapisywania i zarządzania powiązaniami między artefaktami projektu.				
17. Efekty kształcenia:				
Nr	Opis efektu kształcenia	Metoda sprawdzenia efektu kształcenia	Forma prowadzenia zajęć	Odniesienie do efektów dla kierunku studiów
W1	Ma wiedzę z zakresu modelowania i analizy systemów informatycznych rozumianych jako systemy komputerowe i sieci teleinformatyczne oraz związanych z nimi artefaktów.	SP, CL	WT, WM, L	K2A_W04

W2	Ma wiedzę z zakresu projektowania zaawansowanych systemów informatycznych i sieciowych uwzględniając cykl życia projektowanego systemu.	SP, CL	WT, WM, L	K2A_W06
U1	Potrafi, do formułowania i realizacji zadań inżynierskich, posługiwać się technikami informacyjno-komunikacyjnymi w tym technologiami informatycznymi.	CL	WT, WM, L	K2A_U06
K1	Potrafi współdziałać i pracować w grupie przyjmując w niej różne role.	CL	L	K2A_K03
K2	Potrafi myśleć i działać w sposób kreatywny i przedsiębiorczy.	CL	L	K2A_K06
18. Formy zajęć dydaktycznych i ich wymiar (liczba godzin) 30 W. 0 Ćw. 30 L. 0 P. 0 Sem.				

19. Treści kształcenia:

Wykład:

Cechy systemów informatycznych i związanych z nimi artefaktów. Etyka wytwarzania oprogramowania. Klasyczne modele procesów wytwarzania oprogramowania. Rational Unified Process (RUP). Zasady zwinnego wytwarzania oprogramowania (ang. Agile Software Development), XP (ang. Extreme Programming) i SCRUM. Open Unified Process (OpenUP).

Inżynieria wymagań. Klasyfikacje wymagań. Zarządzanie wymaganiami. RequisitePro jako przykład narzędzia wspierającego dokumentację i zarządzanie wymaganiami. Dokument wizji produktu. Łączenie wymagań z innymi artefaktami projektu (ang. *traceability*). Zasady tworzenia i wykorzystywania modelu przypadków użycia.

Wybrane metody modelowania systemów informatycznych. Modelowanie biznesowe (BPMN i BPEL). Modelowanie obiektowe i strukturalne. UML. Diagramy klas, interakcji, aktywności, maszyny stanowej, pakietów i wdrożeniowe. Zasady doboru diagramów UML i zapisu powiązań pomiędzy ich elementami. Tworzenie modeli analitycznych, projektowych i implementacyjnych w UML. SysML. Generacja kodu na podstawie modeli w UML. Zastosowania narzędzi CASE.

Klasyczne wzorce projektowe: adapter, brama, dekorator, fabryki, fasada, kompozyt, interfejs oddzielony, mediator, most, metoda szablonowa, odwiedzający, obserwator, pamiętka, pełnomocnik, polecenie, pula obiektów, pyłek, rejestr, przypadek szczególny, singleton, stan, strategia, usługa zastępcza. Zasady stosowania wzorców projektowych.

Modelowanie architektury systemów informatycznych. Warstwowa budowa systemów informatycznych. Specyfikacja architektury w oparciu o „4+1 View Model” z RUP. Wytwarzanie oprogramowania sterowane modelami (ang. *Model Driven Engineering*).

Wzorce i zręby architektury aplikacji korporacyjnych. Wzorce logiki dziedziny. Wzorce architektury źródła danych. Wzorce prezentacji. Zastosowanie mapowań obiektowo-relacyjnych. Architektura serwerów aplikacyjnych.

Projektowanie w oparciu o usługi sieciowe (ang. Web Services). Architektury chmur obliczeniowych. Integracja korporacyjnych systemów informatycznych. Wzorce projektowe w systemach z przesyłem komunikatów. ESB.

Zapewnienie jakości w procesie wytwarzania oprogramowania – modele ISO 9000 i SEI CMMI. Testowanie systemów informatycznych. Automatyzacja testów. Zastosowanie ciągłej integracji (ang. *Continuous Integration*) w procesie wytwarzania oprogramowania.

Szacowanie kosztu i pracochłonności oprogramowania – model COCOMO i metoda punktów przypadków użycia.

Zajęcia laboratoryjne:

Na laboratorium są wykorzystywane narzędzia IBM-Rational: RequisitePro, Rational Software Architect i InfoSphere Data Architect.

Analiza wymagań. Tworzenie dokumentu wizji i słownika terminologii na podstawie szablonów z RUP.

Modelowanie przypadków użycia. Specyfikacja wybranego przypadku użycia w oparciu o szablon dokumentu z RUP oraz za pomocą diagramów aktywności. Dokumentowanie zależności między wymaganiami.

Opracowanie architektury systemu w oparciu o „4+1 View Model”. Zastosowanie wzorców architektonicznych.

Tworzenie diagramów klas dla warstw: prezentacji, aplikacji, dziedziny i zasobów. Specyfikacja realizacji przypadków użycia. Użycie wzorców projektowych. Wykorzystanie diagramu zmian stanów do opisu zachowania klas.

Generacja schematu bazy danych oraz fragmentów aplikacji.

20. Egzamin: nie

21. Literatura podstawowa:

1. Sommerville I.: Inżynieria oprogramowania. WNT, Warszawa, 2003 (polecane wydanie 9, Person Education, Inc. 2011).
2. Wrycza S., Marcinkowski B., Wyrzykowski K.: Język UML 2.0 w modelowaniu systemów informatycznych. Helion, Gliwice, 2005.
3. Larman C.: UML i wzorce projektowe. Wydanie 3. Helion 2011
4. Gamma E., Helm R., Johnson R., Vlissides J.: Wzorce projektowe. WNT, Warszawa, 2005 lub Helion Gliwice 2010.
5. Fowler M, Rice D., Foemmel M., Hieatt E., Mee R., Stafford R.: Architektura systemów zarządzania przedsiębiorstwem. Wzorce projektowe. Helion, Gliwice, 2005.

22. Literatura uzupełniająca:

1. Erl T.: SOA Design Patterns. Prentice Hall, 2009.
2. Hohpe G., Woolf B.: Enterprise Integration Patterns. Designing, Building and Deploying Messaging Solutions. Addison-Wesley 2004.
3. Kroll P., Kruchten P.: Rational Unified Process od strony praktycznej. WNT, Warszawa, 2007.
4. Beck K., Andres C.: Wydajne programowanie. eXtreme Programming. MIKOM, Warszawa, 2006.
5. Leffingwell D., Widrig D.: Zarządzanie wymaganiami. WNT, Warszawa, 2003.
6. Schneider G., Winters J.: Stosowanie przypadków użycia. WNT, Warszawa, 2004.

23. Nakład pracy studenta potrzebny do osiągnięcia efektów kształcenia

Lp.	Forma zajęć	Liczba godzin kontaktowych / pracy studenta
1	Wykład	30/30
2	Ćwiczenia	0/0
3	Laboratorium	30/30
4	Projekt	0/0
5	Seminarium	0/0
6	Inne	10/0
	Suma godzin	70/60

24. Suma wszystkich godzin: 130**25. Liczba punktów ECTS: 3**

26. Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego (30 + 30 + 10) / 30 = 2,3

27. Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty) (30 + 30) / 30 = 2

26. Uwagi:

Zatwierdzono:

.....
(data i podpis prowadzącego)

.....
(data i podpis dyrektora instytutu/kierownika katedry/
Dyrektora Kolegium Języków Obcych/kierownika lub
dyrektora jednostki międzywydziałowej)