Politechnika Krakowska im. Tadeusza Kościuszki

Karta przedmiotu

obowiązuje studentów rozpoczynających studia w roku akademickim 2012/2013

Wydział Fizyki, Matematyki i Informatyki

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma sudiów: niestacjonarne

Kod kierunku: I

Stopień studiów: I

Specjalności: Brak specjalności

1 Informacje o przedmiocie

Nazwa przedmiotu	Inżynieria oprogramowania
Nazwa przedmiotu w języku angielskim	
Kod przedmiotu	WFMiI I oIN C13 12/13
Kategoria przedmiotu	Przedmioty kierunkowe
Liczba punktów ECTS	4.00
Semestry	6

2 Rodzaj zajęć, liczba godzin w planie studiów

Semestr	Wykład	Ćwiczenia	Laboratorium	Laboratorium komputerowe	Seminarium	Projekt
6	18	0	0	0	0	18

3 Cele przedmiotu

Cel 1 Nabycie umiejętności dekompozycji rzeczywistości będącej podstawą w budowaniu kolejnych modeli inżynierii oprogramowania: od modelu opisowego, przez model wymagań i specyfikacji do numerycznego. Umiejętność identyfikacji procesów analizy i syntezy w procesach wytwórczych oprogramowania.

Cel 2 Poznanie podstawowych modeli wytwarzania oprogramowania (cykli życia): kaskadowych, cyklicznych i ewo-

lucyjnych. Poznanie poszczególnych faz procesu wytwórczego oraz tworzonych w efekcie ich realizacji artefaktów. Umiejętność adaptacji poszczególnych faz w procesach ciężkich i lekkich.

Cel 3 Poznanie metodyk identyfikacji i specyfikacji wymagań oraz procesów zarządzania nimi. Zdobycie umiejętności modelowania przy wykorzystaniu UML2 oraz BPMN2 w modelowaniu procesów biznesowych. Zapoznanie się z metodami estymacji kosztów wytwarzania oprogramowania.

Cel 4 Zaznajomienie się zasadami projektowania, implementacji, testowania i wdrożenia oprogramowania. Zdobycie umiejętności identyfikacji metodyk formalnych i agilnych w procesach wytwórczych oprogramowania. Nabycie zdolności i umiejętności dokumentowania kolejnych kroków i faz procesu wytwarzania oprogramowania.

Cel 5 Zdobycie doświadczeń w wykorzystywaniu współczesnych narzędzi automatyzacji procesów wytwarzania oprogramowania. Poznanie elementów zarządzania procesem wytwórczym, zarządzanie ryzykiem; zarządzanie wersjami; wprowadzenie do zarządzania projektami informatycznymi.

4 Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji

1 Znajomość dowolnego języka programowania wysokiego poziomu w zakresie pozwalającym na implementacje prostej aplikacji

2 Elementy algorytmów i struktur danych oraz podstawy baz danych

5 Efekty kształcenia

EK1 Wiedza Zdobywa wiedzę dotyczącą zagadnień dekompozycji rzeczywistości. Zna sposoby prowadzenia analizy zachodzących procesów oraz rozumie techniki ich modelowania. Rozumie na czym polega wykorzystywanie abstrakcji przy przejściu od rzeczywistości do jej cyfrowego modelu poprzez analizę modelu wymagań i syntezę do modelu systemu (I1_W01, I1W04)

EK2 Wiedza Zdobywa wiedzę na temat procesu wytwórczego oprogramowania i realizacji poszczególnych faz jego cyklu życia (I1_W09). Ma wiedzę w zakresie projektowania i implementacji prostych i złożonych systemów informatycznych przy wykorzystaniu współczesnych technik programistycznych i języków oprogramowania (I1_W10)

EK3 Wiedza Zdobywa wiedzę w zakresie syntezy modeli numerycznych w notacji obiektowej i strukturalnej(I1_W06); posiada wiedzę pozwalającą na właściwą ocenę jakości tworzonego oprogramowania (I1_W13)

EK4 Wiedza Zdobywa podstawową wiedzę w zakresie organizacji i prowadzenia procesu wytwórczego systemów informatycznych (I1_W14) z wykorzystaniem narzędzi interpretowania i modelowania procesów biznesowych. Zapoznaje się z narzędziami wspierającymi automatyzację procesu wytwórczego oprogramowania (I1_W08). Zapoznaje się z podstawowymi zagadnieniami prowadzenia projektu informatycznego (I1_W15). Zdobywa wiedzę pozwalającą na wybór właściwej metodyki realizacji systemu informatycznego.

EK5 Umiejętności Nabywa umiejętności dekompozycji rzeczywistości w procesie tworzenia modeli opisowych z uwzględnieniem ich aspektów społecznych i prawnych (I1_U06). Potrafi przygotować odpowiednią specyfikację wymagań (I1_U22) i potrafi estymować koszty i pracochłonność wytwarzania oprogramowania (I1_U19).

EK6 Umiejętności Zdobywa umiejętności transformacji specyfikowanego modelu wymagań na odpowiedni model obiektowy lub strukturalny (I1_U08, I1_U23) z wykorzystaniem prostych systemów bazodanowych (I1_U13)

EK7 Umiejętności Potrafi w procesie projektowania systemu określić jego czaso- i kosztochłonność (I1_U09). Nabywa umiejętności wykorzystania odpowiednich technologii programistycznych w transformacji modelu numerycznego (diagramy) na odpowiedni model cyfrowy (I1_U07)

EK8 Umiejętności Umie przeprowadzić weryfikację stworzonego systemu cyfrowego poprzez kompleksowe jego testowanie (I1_U14) Dla wdrożenia systemu potrafi ocenić stworzone narzędzie i określić jego przydatność jak również efektywność zrealizowanej funkcjonalności (I1_U20).Gwałtowny rozwój technologii informatycznych, wymuszony rozwojem sprzętu, budzi świadomość konieczności podążania za rozwijającymi się technologiami informatycznymi(I1_U24)

EK9 Kompetencje społeczne Uzyskanie zdolności komunikacji interpersonalnej - słuchania i wyrażania, pozwalające na właściwe interpretowanie rzeczywistości do modelu wymagań (I1_K02).Praca przy projekcie systemu wymaga pracy zespołowej, stąd nabywa się zdolności pracy w grupie (I1_K04). Zarówno w zespole jak i w grupie każdy musi się kierować zasadami profesjonalizmu i etyki (I1_K03)

EK10 Kompetencje społeczne Potrafi analizować procesy nie tylko techniczne ale również i społeczne. Umie je analizować i prowadzić ich modelowanie, nie tracąc kontaktu z rzeczywistością. Poznaje przyzwyczajenia i nawyki użytkowników systemów informatycznych. Zdobywa świadomość roli społecznej Inżyniera Oprogramowania (I1_K07, I1_K06, I1_K05)

6 Treści programowe

Wykład		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
W1	Wprowadzenie do przedmiotu Inżynieria Oprogramowania - tło historyczne, odniesienie do innych dyscyplin inżynierskich - ogólna charakterystyka.	1
W2	Procesy i metody produkcji oprogramowania. Modele cyklu życia oprogramowania: kaskadowe, iteracyjne, ewolucyjne. Porównanie cyklów życia oprogramowania - problemy projektów dużej i małej skali. Wprowadzenie do Nauki o Modelach, zagadnienia właściwej transformacji wymagań.	2
W3	Analiza od rzeczywistości przez model opisowy, specyfikacje wymagań do modelu numerycznego i następnie modelu cyfrowego rzeczywistości, czyli systemu. Rola wymagań i sposoby ich specyfikacji oraz dokumentacji. Szacowanie kosztów realizacji systemu.	3
W4	Dynamika i statyka systemu informatycznego. Języki i paradygmaty wspomagania modelowania systemów informatycznych (diagramy UML) i procesów biznesowych (diagramy BPMN)	3
W5	Projektowanie oprogramowania, model strukturalny a obiektowy. Transformacja od modelu numerycznego do modelu cyfrowego, architektura aplikacji, strategie narzędzia. Wzorce projektowe, model komponentowy	3
W6	Implementacja systemu informatycznego. Dobór technologii do wymagań i projektu. Standardy.	2
W7	Zagadnienia jakości, testowanie - sposoby testowania: white/black box testing, testy regresyjne, ciągła integracja, inspekcje, przeglądy artefaktów, metryki jakości	2
W8	Procesy i praktyki zwinne "agilne" na przykładzie XP i Scrum. Dobór metodyki do wielkości projektu.	1
W9	Wybrane zagadnienia inżynierii oprogramowania: zarządzanie konfiguracją, zarządzanie zmianami, zarządzanie ryzykiem, elementy pracy w zespole projektowym.	1

Projekt		
Lp	Tematyka zajęć Opis szczegółowy bloków tematycznych	Liczba godzin
P1	Studenci w grupach 3 - 5 osób realizują kolejne elementy przedsięwzięcia projektowego. Kolejne etapy jego realizacji stanowią poszczególne przedsięwzięcia projektowe: Wybór zadania projektowego - Rozmowa z klientem, stworzenie modelu opisowego i modelu wymagań	3
P2	Identyfikacja przypadków użycia systemu dla określonej grupy aktorów. Analiza problemu. Specyfikacja wymagań i ich weryfikacja. Wstępny projekt struktury systemu.	3
P3	Projekt struktury systemu. Model UML lub BMNP. Budowa ostateczne specyfikacji systemu.	3
P4	Implementacja systemu. Wybór technologi i narzędzi. Kodowanie systemu w wybranej technologii	3
P5	Opracowanie testów systemu, jednostkowych, modułowych, funkcjonalnych. Automatyzacja procesu testowania.	3
P6	Wdrożenie i uruchomienie systemu	3

7 Narzędzia dydaktyczne

N1 Wykłady

N2 Ćwiczenia projektowe

8 Obciążenie pracą studenta

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe z nauczycielem akademickim, w tym:	
Godziny wynikające z planu studiów	0
Konsultacje przedmiotowe	4
Egzaminy i zaliczenia w sesji	2
Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym:	
Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury	43
Opracowanie wyników	5
Przygotowanie raportu, projektu, prezentacji, dyskusji	30
Sumaryczna liczba godzin dla przedmiotu wynikająca z całego nakładu pracy studenta	**84**
Sumaryczna liczba punktów ECTS dla przedmiotu	4.00

9 Sposoby oceny

Ocena formująca

F1 Projekt indywidualny

Ocena podsumowująca

P1 Test

P2 Egzamin ustny

Ocena aktywności bez udziału nauczyciela

B1 Projekt indywidualny

Kryteria oceny

Efekt kształcenia 1	
Na ocenę 3.0	Wie co to jest dekompozycja i wie na czym polega dekompozycja rzeczywistości, potrafi ja zastosować do zasugerowanego przypadku.
Na ocenę 3.5	Spełnia wymaganie na ocenę 3,0 oraz potrafi przeprowadzić analizę zdekomponowanych artefaktów.

Na ocenę 4.0	Spełnia wymaganie na ocenę 3,5 oraz rozumie na czy polega wykorzystanie abstrakcji w modelowaniu.
Na ocenę 4.5	Spełnia wymaganie na ocenę 4,0 oraz potrafi przejść od rzeczywistości do modelu cyfrowego.
Na ocenę 5.0	Spełnia wymaganie na ocenę 4,5 oraz jak proces syntezy i analizy jest wykorzystywany w procesie tworzenia oprogramowania.
Efekt kształcenia 2	
Na ocenę 3.0	Zna podstawowe zagadnienia cyklu życia oprogramowania. Potrafi omówić referencyjny model kaskadowy.
Na ocenę 3.5	Spełnia wymaganie na ocenę 3,0 oraz w modelu kaskadowym potrafi określić poszczególne fazy i wyjaśnić ich znaczenie dla procesu wytwórczego oprogramowania.
Na ocenę 4.0	Spełnia wymaganie na ocenę 3,5 oraz potrafi wskazać i omówić inne cykle życia oprogramowania: iteracyjne, przyrostowe i ewolucyjne.
Na ocenę 4.5	Spełnia wymaganie na ocenę 4,0 oraz potrafi uzasadnić wybór odpowiedniej technologii informatycznej do typu realizowanego projektu w określonym cyklu życia oprogramowania.
Na ocenę 5.0	Spełnia wymaganie na ocenę 4,5 oraz zna zaawansowane procesy wytwórcze jak: RUP. Potrafi identyfikować otoczenie i środowisko programistyczne odpowiednie dla dużych i małych projektów
Efekt kształcenia 3	
Na ocenę 3.0	Zna podstawy notacji obiektowej i notacji strukturalnej.
Na ocenę 3.5	Spełnia wymagania na ocenę 3,0 oraz potrafi identyfikować z modelu wymagań poszczególne elementy modelu numerycznego w notacji obiektowej i strukturalnej.
Na ocenę 4.0	Spełnia wymagania na ocenę 3,5 oraz potrafi właściwie dobrać typ modelu programowania do analizowanego zagadnienia.
Na ocenę 4.5	Spełnia wymagania na ocenę 4,0 oraz potrafi szacować jakość tworzonego oprogramowania
Na ocenę 5.0	Spełnia wymagania na ocenę 4,5 oraz potrafi zidentyfikować elementy syntezy niezbędnej dla integralności tworzonego systemu
Efekt kształcenia 4	
Na ocenę 3.0	Zna zasady prowadzenia przedsięwzięcia wytwórczego oprogramowania i potrafi wskazać na właściwe narzędzia do wykorzystania na etapie analizy.
Na ocenę 3.5	Spełnia wymagania na ocenę 3,0 oraz potrafi wykorzystać stosowne narzędzia UML w procesie identyfikacji i analizy wymagań.
Na ocenę 4.0	Spełnia wymagania na ocenę 3,5 oraz potrafi wykorzystać narzędzia BPMN w modelowaniu procesów biznesowych.

Na ocenę 4.5	Spełnia wymagania na ocenę 4,0 oraz zna narzędzi pozwalające na automatyzację procesu wytwórczego. Biegle posługuje sie metodykami wytwarzania oprogramowania.
Na ocenę 5.0	Spełnia wymagania na ocenę 3,0 oraz potrafi identyfikować podstawowe problemy związane z prowadzeniem projektu informatycznego
Efekt kształcenia 5	
Na ocenę 3.0	Potrafi przeprowadzić dekompozycje wybranego zagadnienia.
Na ocenę 3.5	Spełnia wymagania na ocenę 3,0 oraz potrafi identyfikując aspekty prawne i społeczne zdekomponowanych wymagań.
Na ocenę 4.0	Spełnia wymagania na ocenę 3,5 oraz potrafi przygotować odpowiednią specyfikacje wymagań.
Na ocenę 4.5	Spełnia wymagania na ocenę 4,0 oraz potrafi przeprowadzić szacowanie kosztów wytwarzania oprogramowania.
Na ocenę 5.0	Spełnia wymagania na ocenę 4,5 oraz potrafi estymować pracochłonność procesu wytwórczego oprogramowania.
Efekt kształcenia 6	
Na ocenę 3.0	Potrafi zbudować odpowiednie diagramy statyki i dynamiki projektowanej aplikacji.
Na ocenę 3.5	Spełnia wymagania na ocenę 3,0 oraz potrafi zbudować odpowiednie diagramy dla modelu strukturalnego.
Na ocenę 4.0	Spełnia wymagania na ocenę 3,5 oraz potrafi podpiąć do aplikacji proste systemy bazodanowe i zarządzać nimi.
Na ocenę 4.5	Spełnia wymagania na ocenę 4,0 oraz potrafi implementować do modelu zaawansowane technologie informatyczne pozwalające na automatyczną realizację określonych funkcjonalności (frameworki)
Na ocenę 5.0	Spełnia wymagania na ocenę 4,5 oraz wykazuje ogromna swobodę w wykorzystywaniu współczesnych technologi informatycznych w modelowaniu wymagań.
Efekt kształcenia 7	
Na ocenę 3.0	Zna techniki szacowania kosztów oprogramowania.
Na ocenę 3.5	Spełnia wymagania na ocenę 3,0 oraz umie oszacować koszty i pracochłonność oprogramowania.
Na ocenę 4.0	Spełnia wymagania na ocenę 3,5 oraz zna podstawowe technologie programistyczne i możliwości ich implementacji w projekcie.
Na ocenę 4.5	Spełnia wymagania na ocenę 4,0 oraz potrafi w praktyce przeprowadzić podział i agregację poszczególnych modułów systemu.

Na ocenę 5.0	Spełnia wymagania na ocenę 4,5 oraz potrafi zastosować model komponentowy przy tworzeniu oprogramowania.
Efekt kształcenia 8	
Na ocenę 3.0	Potrafi wskazać na element zakończenia wytwarzania oprogramowania.
Na ocenę 3.5	Spełnia wymagania na ocenę 3,0 oraz potrafi przeprowadzić wstępne testowanie systemu.
Na ocenę 4.0	Spełnia wymagania na ocenę 3,5 oraz potrafi przeprowadzić zaawansowane testowanie zbudowanego systemu. Potrafi skonfigurować kolejne grupy testów.
Na ocenę 4.5	Spełnia wymagania na ocenę 4,0 oraz potrafi ocenić efektywność zrealizowanej aplikacji.
Na ocenę 5.0	Spełnia wymagania na ocenę 3,0 oraz potrafi ocenić mozliwość zastosowanie nowszej, bardziej zaawansowanej technologii
Efekt kształcenia 9	
Na ocenę 3.0	Właściwie przeprowadza dekompozycję na podstawie wywiadu z klientem
Na ocenę 3.5	Spełnia wymagania na ocenę 3,0 oraz potrafi przekazać wymagania pracownikom zespołu w formie właściwej specyfikacji.
Na ocenę 4.0	Spełnia wymagania na ocenę 3,5 oraz potrafi podzielić model rzeczywistości na podsystemy realizowane przyrostowo lub w grupach.
Na ocenę 4.5	Spełnia wymagania na ocenę 4,0 oraz potrafi właściwie zaplanować realizację poszczególnych etapów prac.
Na ocenę 5.0	Spełnia wymagania na ocenę 4,5 oraz właściwa ocena roli etyki i profesjonalizmu w realizacji przedsięwzięcia projektowego.
Efekt kształcenia 10	
Na ocenę 3.0	Potrafi rozróżnić pomiędzy procesem społecznym i procesem technicznym i pokazuję zrozumienie różnic w wymaganiach stosowanych przy ich modelowaniu.
Na ocenę 3.5	Spełnia wymagania na ocenę 3,0 oraz potrafi wprowadzić do takich procesów elementy ich aktualizacji do środowiska.
Na ocenę 4.0	Spełnia wymagania na ocenę 3,5 oraz potrafi uwzględnić w modelowaniu przyzwyczajenia użytkowników.
Na ocenę 4.5	Spełnia wymagania na ocenę 4,0 oraz potrafi decydować jako Inżynier oprogramowania o elementach istotnych z punktu widzenia wykorzystywanych narzędzi informatycznych
Na ocenę 5.0	Spełnia wymagania na ocenę 4,5 oraz wykazuje ogromne doświadczenie w pracy z klientami przy projektowaniu systemów informatycznych

10 Macierz realizacji przedmiotu

Efekt kształcenia	Odniesienie danego efektu do szczegółowych efektów zdefiniowanych dla programu	Cele przedmiotu	Treści programowe	Narzędzia dydaktyczne	Sposoby oceny
EK1	I1_W01, I1_W04	Cel 1 Cel 2 Cel 3	W1 W2 W3 P1 P2	N1 N2	F1 P2
EK2	I1_W09, I1_W10	Cel 1 Cel 2 Cel 3	W2 W3 W6 P1 P2	N1 N2	F1 P1 P2
EK3	I1_W06, I1_W13	Cel 2 Cel 3	W4 W5 W6	N1 N2	F1 P1 P2
EK4	I1_W09, I1_W14, I1_W15	Cel 3 Cel 4	W4 W5 W6 W8 W9 P2 P3	N1 N2	F1 P1 P2
EK5	I1_W06, I1_U19, I1_U22	Cel 3 Cel 4 Cel 5	W2 W3 W6 P2 P5	N1 N2	F1 P2
EK6	I1_W08, I1_U13, I1_U23	Cel 3 Cel 4	W3 W4 W5 P2 P3	N1 N2	F1 P1
EK7	I1_U07, I1_U09	Cel 2 Cel 3	W2 W3 W4 W6 P2 P3 P4	N1 N2	F1 P1
EK8	I1_U14, I1_U20, I1_U24	Cel 4 Cel 5	W5 W6 W7 W9 P4 P5 P6	N1 N2	F1 P1
EK9	I1_K02, I1_K03, I1_K04	Cel 4 Cel 5	W7 W8 W9 P6	N1 N2	F1
EK10	I1_K05, I1_K06, I1_K07	Cel 4 Cel 5	W7 W8 W9 P6	N1 N2	F1 P1 P2

11 Wykaz literatury

Literatura podstawowa

[1] Jaszkiewicz A. — *Inżynieria oprogramowania*, Gliwice, 1997, Helion

[2] Sommerville J. — *Inżynieria oprogramowania - klasyka informatyki*, Warszawa, 2003, WNT

[3] Hamlet D., Maybee J. — *Podstawy techniczne inżynierii oprogramowania*, Warszawa, 2003, WNT

[4] Maciaszek L.A., Liong B.L. — *Practical Software Engineering a Case Study Approach*, Melbourn, 2005, Pearson Education Limited

[5] Śmiałek M. — *Zrozumieć UML 2.0*, Gliwice, 2005, Helion

[6] Steve McConnell — *KOD Doskonały*, Gliwice, 2010, Helion

Literatura uzupełniająca

[1] Brooks F.P. — *Mityczny osobomiesiąc*, Warszawa, 2000, WNT

[2] Hunt A., Thomas D. — *Pragmatyczny programista - od czeladnika do mistrza*, Warszawa, 2002, WNT

[3] Joel Spolsky — *Sztuka Pisania Oprogramowania*, Gliwice, 2007, Helion

12 Informacje o nauczycielach akademickich

Osoba odpowiedzialna za kartę

dr hab. inż. prof.PK. Marek Stanuszek (kontakt: `mareks@pk.edu.pl`)

Osoby prowadzące przedmiot

1 dr inż. Piotr Poznański (kontakt: `poznan@mars.iti.pk.edu.pl`)

2 dr inż. Paweł Jarosz (kontakt: `pjarosz@pk.edu.pl`)

3 mgr inż. Marek Kosiorowski (kontakt: `mkosiorowski@pk.edu.pl`)

13 Zatwierdzenie karty przedmiotu do realizacji

(miejscowość, data) (odpowiedzialny za przedmiot) (dziekan)

Przyjmuję do realizacji (data i podpisy osób prowadzących przedmiot)

. . . . . .

. . . . . .

. . . . . .