**Nazwa przedmiotu:**

Technologia Wytwarzania Oprogramowania

**Koordynator przedmiotu:**

Anna DEREZIŃSKA

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

TWO

**Semestr nominalny:**

6 / rok ak. 2015/2016

**Liczba punktów ECTS:**

4

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

120 godzin
Bilans nakładu pracy studenta :
- udział w wykładach: 15 x 2 godz. = 30 godz.,
- przygotowanie do kolejnych wykładów i realizacji laboratorium (przejrzenie materiałów z wykładu i dodatkowej literatury, konsultacje): 10 godzin (w tym 1 godz. konsultacji)
- realizacja zadań laboratoryjnych: (15h zajęć laboratoryjnych, oraz obejmuje także zapoznanie się z instrukcjami laboratoryjnymi, przygotowanie programów i ich testów oraz modeli przed laboratoriami, przygotowanie rozwiązań zadań, dokończenie zadań po laboratorium oraz przygotowanie kolejnych sprawozdań, recenzję pracy innego studenta, uzgodnienia zadań pomiędzy członkami zespołu), 15 + (15+25+10) = 65
- przygotowanie do 2 kolokwiów i 2 kartkówek (rozwiązanie zadań przedkolokwialnych, udział w konsultacjach, przygotowanie materiałów na kolokwia): 15 godz. (w tym 1 godz. konsultacji)

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

30 + 15 + 1 + 1 = 47 godz., co odpowiada ok. 1.5-2 punktom ECTS

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

65 godzin co odpowiada ok. 2-2.5 punktom ECTS.

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 15h |
| Projekt:  | 0h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Wiedza i umiejętności z przedmiotów Programowanie Obiektowe (np. C++, Java lub C#) oraz Inżynieria Oprogramowania - lub równoważne.
Zakłada się, że studenci posiadają podstawową wiedzę z zakresu programowania obiektowego, projektowania obiektowego i strukturalnego, UML.

**Limit liczby studentów:**

50

**Cel przedmiotu:**

- rozszerzenie i pogłębienie wiedzy i umiejętności z dziedziny inżynierii oprogramowania.
- ukształtowanie oceny różnych procesów i artefaktów inżynierii oprogramowania w kategoriach technologicznych, ekonomicznych i społecznych
- zapoznanie z wybranymi problemami i metodami ich rozwiązywania dotyczącymi: specyfikacji i zarządzania wymaganiami, specyfikacji kontraktowych oraz specyfikacji formalnych, testowania i inspekcji oprogramowania, zapewnienia wysokiej jakości i niezawodności oprogramowania, pomiarów procesu i produktu, ewolucji oprogramowania, zarządzania projektami.

**Treści kształcenia:**

Wprowadzenie (1h). Problematyka jakości w produkcji oprogramowania. Ekonomiczne i społeczne aspekty inżynierii oprogramowania. Konsekwencje przeciekania błędów.

Inżynieria wymagań(4h). Model Sommerville'a-Sayera, praktyki inżynierii wymagań, zbieranie, specyfikacja i weryfikacja wymagań, standardy, zarządzanie zmianami. Specyfikacje wymagań w SySML.

Specyfikacja kontraktowa modeli obiektowych (4h) - projektowanie z wykorzystaniem kontraktów (ang. design by contract), język OCL, niezmienniki klas i stanów, warunki początkowe i końcowe operacji, precyzyjne modele klas i stanów UML

Specyfikacje formalne (2h). Specyfikacje formalne, rodzaje, zastosowanie, specyfikacje algebraiczne.
Projektowanie z użyciem wielokrotnym (1h). Linie produkcyjne, Komponenty - wykorzystanie, tworzenie, katalogowanie.

Wzorce projektowe (2) Idea stosowania wzorców. Wzorce strukturalne (adapter, dekorator, proxy), czynnościowe (strategia, stan, obserwator), kreacyjne (wirtualny konstruktor, fabryka abstrakcyjna), przykłady, narzędzia.

Weryfikacja i walidacja oprogramowania (4h). Testowanie ukierunkowane na wyszukiwanie defektów- strukturalne i funkcjonalne (klasy równoważności, pokrycia kodu, testowanie ścieżek), testowanie integracyjne, obiektowe, inne. Porównanie kryteriów testowania. Określanie wyroczni. Narzędzia wspomagające. Inspekcje kodu, listy kontrolne. Zasady łatwo-testowalnego oprogramowania. Przykłady.

Wiarygodność oprogramowania (2h). Miary niezawodności, modele niezawodnościowe, metody podnoszenia niezawodności oprogramowania

Metryki oprogramowania (1h). Miary oprogramowania strukturalnego i obiektowego. Metryki obiektowe projektu i klas. Metryki a złożoność testowania.

Jakość oprogramowania i procesów wytwórczych (1h). Modele i standardy jakości produktów i procesów, modele ISO 9126, SQuaRE, ISO 25000, Ocena i doskonalenie procesów wytwórczych, ISO 9000, CMMI

Wybrane problemy zarządzania (3h). Zapewnienie jakości i zarządzanie jakością, miary DRE, Zarządzanie ryzykiem. Planowanie i kontrola, wykresy kontrolne. Lekkie metodyki, programowanie ekstremalne XP, SCRUM. Zespół projektowy, Zasady skutecznej pracy indywidualnej i zespołowej, People-CMM, budowanie zespołu bazujące na rolach, zarządzanie kompetencjami.

Ewolucja oprogramowania(2h). Pielęgnowanie i modyfikacja oprogramowania Koszty. Zarządzanie zmianami; narzędzia wspomagające, zarządzanie wersjami CVS. Systemy odziedziczone. Inżynieria ponowna (wsteczna) Refaktoryzacja kodu.

Zakres laboratorium:
Celem laboratorium jest utrwalenie znajomości zasad projektowania i programowania obiektowego, oraz praktyczne zapoznanie się czynnościami z różnych etapów procesu tworzenia oprogramowania. Podczas zajęć laboratoryjnych studenci poznają także możliwości różnych narzędzi CASE wspierających poszczególne zadania inżynierii oprogramowania oraz elementy pracy zespołowej.

Lab1, 2 - specyfikacja i zarządzanie wymaganiami
Określenie wymagań (cechy systemu, wymagania udziałowców, ograniczenia, wymagania funkcjonalne) Atrybuty wymagań (priorytety, wymagania nadrzędne i potomne, zależności pomiędzy wymaganiami różnych typów). Specyfikacja przypadków użycia z uwzględnieniem warunków początkowych i końcowych. Uzgodnienie wymagań w obrębie zespołu.

Wymiana specyfikacji wymagań pomiędzy studentami. Przygotowanie recenzji dla specyfikacji wymagań wg cech IEEE 830.

Lab 3-5 - projektowanie obiektowe z wykorzystaniem wzorców projektowych, precyzyjne modelowanie ze specyfikacją kontraktów (OCL), transformacje modelu do kodu i implementacja wybranych części systemu

Lab 6 - testowanie
Określenie pokrycia kodu. Tworzenie testów dla niepokrytego kodu. Minimalizacja zbioru testów dla zadanych kryteriów testowania.

Lab 7 - rezerwowy dla ww zadań

**Metody oceny:**

- rezultaty sprawdzianów audytoryjnych Spr1 i Spr2
- rozwiązywanie zadań na wykładzie dla wybranych zagadnień
- rezultaty kartkówek na laboratorium Kart1 i Kart2
- ocena pracy laboratoryjnej (prezentacji i dyskusji cząstkowych wyników, spójności danych w ramach zespołu, efektywności poprawy elementów zadań, sprawozdań z ostatecznymi wynikami, dyskusji wyników końcowych)
- ocena pracy domowej (recenzji)

**Egzamin:**

nie

**Literatura:**

[1] I. Sommerville: Inżynieria oprogramowania, WNT 2003 ,
[2] R. S. Pressman, Praktyczne podejście do inżynierii oprogramowania, WNT 2004,
[3] D. Leffingwell, D. Widrig: Zarządzanie wymaganiami, WNT 2003
[4] B. Bruegge, A.H. Dutoit: Inżynieria oprogramowania w ujęciu obiektowym, UML, wzorce projektowe i Java, Helion 2011
[5] B. Meyer Programowanie zorientowane obiektowo, Helion 2005/8
[6] S.Wrycza, B. Marcinkowski Język inżynierii systemów SysML. Architektura i zastosowanie, Profile UML 2.x w praktyce. Helion 2010
[7] J. Warmer, A. Kleppe: OCL Precyzyjne modelowanie w UML, WNT 2003
[8] A. Shaloway, J.R. Trott: Projektowanie zorientowane obiektowo – wzorce projektowe, Helion 2001 (wyd. 1), Helion 2005 (wyd. 2)
[9] E.Gamma, R.Helm, R.Johnson, J.Vlissides, Wzorce projektowe, WNT 2005, II wyd. WNT 2008, Helion 2010
[10] R. Patton: Testowanie oprogramowania, Micom 2002
[11] B.Wiszniewski, B. Bereza-Jarociński, Teoria i praktyka testowania programów, PWN 2006

**Witryna www przedmiotu:**

http://studia.elka.pw.edu.pl/

**Uwagi:**

Uwagi realizacyjne: do realizacji zajęć laboratoryjnych konieczne jest posiadanie profesjonalnych, licencjonowanych narzędzi CASE do zarządzania wymaganiami, projektowania, testowania, ...

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt IOP2\_W01:**

Student, który zaliczył przedmiot, posiada podstawową wiedzę na temat: metod specyfikacji i zarządzania wymaganiami

Weryfikacja:

Ocena wyników Spr1, Lab1-2

**Powiązane efekty kierunkowe:** K\_W04, K\_W13

**Powiązane efekty obszarowe:** T1A\_W08, T1A\_W04, T1A\_W05

**Efekt IOP2\_W02:**

Student, który zaliczył przedmiot, posiada podstawową wiedzę na temat: projektowania modeli obiektowy z wykorzystaniem wzorców projektowych

Weryfikacja:

Ocena wyników Spr1, Spr2, Lab3-5

**Powiązane efekty kierunkowe:** K\_W13

**Powiązane efekty obszarowe:** T1A\_W04, T1A\_W05

**Efekt IOP2\_W03:**

... na temat: specyfikacji kontraktów (niezmienników, warunków początkowych i końcowych) dla modeli obiektowych w języku OCL

Weryfikacja:

Ocena wyników Spr1, Spr2, Lab3-5, Kart1

**Powiązane efekty kierunkowe:** K\_W13

**Powiązane efekty obszarowe:** T1A\_W04, T1A\_W05

**Efekt IOP2\_W04:**

na temat: prostych miar niezawodności, metryk oprogramowania, jakości oprogramowania, jakości zbiorów testów i procesów wytwarzania oprogramowania

Weryfikacja:

Ocena wyników Spr2, Kart2

**Powiązane efekty kierunkowe:** K\_W13

**Powiązane efekty obszarowe:** T1A\_W04, T1A\_W05

**Efekt IOP2\_W05:**

.. na temat wybranych specyfikacji formalnych

Weryfikacja:

Ocena wyników Spr1, Spr2,

**Powiązane efekty kierunkowe:** K\_W13

**Powiązane efekty obszarowe:** T1A\_W04, T1A\_W05

**Efekt IOP2\_W06:**

...na temat: metod weryfikacji statycznej i dynamicznej oprogramowania, kryteriów testowania dla wybranych strategii testowania,

Weryfikacja:

Ocena wyników Spr1, Spr2, Kart2, Lab6

**Powiązane efekty kierunkowe:** K\_W13

**Powiązane efekty obszarowe:** T1A\_W04, T1A\_W05

**Efekt IOP2\_W07:**

... na temat: ekonomicznych i społecznych konsekwencji dla wybranych zagadnień organizacji pracy i zarządzania wytwarzaniem oprogramowania

Weryfikacja:

Ocena wyników Spr1, Spr2

**Powiązane efekty kierunkowe:** K\_W04, K\_W05, K\_W13, K\_W15

**Powiązane efekty obszarowe:** T1A\_W08, T1A\_W08, T1A\_W09, T1A\_W04, T1A\_W05, T1A\_W06

**Efekt IOP2\_W08:**

... na temat: ewolucji oprogramowania

Weryfikacja:

Ocena wyników Spr2

**Powiązane efekty kierunkowe:** K\_W13, K\_W15

**Powiązane efekty obszarowe:** T1A\_W04, T1A\_W05, T1A\_W06

### Profil ogólnoakademicki - umiejętności

**Efekt IOP2\_U01:**

Student, który zaliczył przedmiot potrafi: opracować specyfikację wymagań, ocenić jakość takiej specyfikacji,
wskazać podstawowe ekonomiczne i społeczne konsekwencje dla wybranych zagadnień organizacji pracy i zarządzania wytwarzaniem oprogramowania, oceny jakości oprogramowania, ewolucji oprogramowania

Weryfikacja:

Ocena wyników Spr1, Lab1-2, Ocena recenzji

**Powiązane efekty kierunkowe:** K\_U14, K\_U21

**Powiązane efekty obszarowe:** T1A\_U07, T1A\_U09, T1A\_U14, T1A\_U15, T1A\_U16

**Efekt IOP2\_U02:**

Student, który zaliczył przedmiot potrafi: zaprojektować model obiektowy z wykorzystaniem wzorców projektowych; zaprojektować model obiektowy z określeniem prostych kontraktów (niezmienników, warunków początkowych i końcowych) w danym języku specyfikacji; zaimplementować program na podstawie takich modeli z wykorzystaniem narzędzi CASE

Weryfikacja:

Ocena wyników Spr1, Spr2, Kart1, Lab3-5,. Rozwiązywanie zadań na wykładzie

**Powiązane efekty kierunkowe:** K\_U13, K\_U15, K\_U21

**Powiązane efekty obszarowe:** T1A\_U16, T1A\_U09, T1A\_U15, T1A\_U16, T1A\_U15, T1A\_U16

**Efekt IOP2\_U03:**

Student, który zaliczył przedmiot potrafi: obliczyć proste miary niezawodności i metryki oprogramowania

Weryfikacja:

Ocena wyników Spr2, Kart2, Rozwiązywanie zadań na wykładzie

**Powiązane efekty kierunkowe:** K\_U02

**Powiązane efekty obszarowe:** T1A\_U08, T1A\_U09

**Efekt IOP2\_U04:**

Student, który zaliczył przedmiot potrafi: zmodyfikować i zastosować proste specyfikacje formalne (specyfikacje algebraiczne)

Weryfikacja:

Ocena wyników Spr1, Spr2 Rozwiązywanie zadań na wykładzie

**Powiązane efekty kierunkowe:** K\_U01, K\_U14

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U07, T1A\_U09, T1A\_U14

**Efekt IOP2\_U06:**

Student, który zaliczył przedmiot potrafi: wykorzystać różne kryteria testowania dla wybranych strategii testowania, poprawić zbiór testów wykorzystując narzędzie do badania pokrycia kodu

Weryfikacja:

Ocena wyników Spr2, Kart2, Lab6 Rozwiązywanie zadań na wykładzie

**Powiązane efekty kierunkowe:** K\_U02, K\_U19, K\_U21

**Powiązane efekty obszarowe:** T1A\_U08, T1A\_U09, T1A\_U16, T1A\_U15, T1A\_U16

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt IOP2\_K01:**

Student, który zaliczył przedmiot potrafi: wskazać podstawowe ekonomiczne i społeczne konsekwencje dla wybranych zagadnień organizacji pracy i zarządzania wytwarzaniem oprogramowania, określenia wymagań, oceny jakości oprogramowania, ewolucji oprogramowania

Weryfikacja:

Ocena wyników Spr1, Spr2

**Powiązane efekty kierunkowe:** K\_K02

**Powiązane efekty obszarowe:** T1A\_K02

**Efekt IOP2\_K02:**

Student, który zaliczył przedmiot potrafi: określić priorytety wymagań

Weryfikacja:

Ocena wyników Spr1, Lab1-2, recenzji

**Powiązane efekty kierunkowe:** K\_K04

**Powiązane efekty obszarowe:** T1A\_K04

**Efekt IOP2\_K03:**

Student, który zaliczył przedmiot potrafi: pracować indywidualnie i w zespole przyjmujac różne role

Weryfikacja:

Ocena wyników Spr2, Lab1-2, Lab3-5

**Powiązane efekty kierunkowe:** K\_K03

**Powiązane efekty obszarowe:** T1A\_K03