**Nazwa przedmiotu:**

Inżynieria oprogramowania

**Koordynator przedmiotu:**

Krzysztof SACHA

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

IOP

**Semestr nominalny:**

5 / rok ak. 2014/2015

**Liczba punktów ECTS:**

4

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

100
- 30 godzin wykładu
- 15 godzin ćwiczeń laboratoryjnych
- 5 godzin konsultacji
- 50 godzin samodzielnej pracy zwiazanej z przygotowaniem do kolokwium i przygotowaniem dokumentacji projektowej niezbędnej do zaliczenia ćwiczeń laboratoryjnych

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

2

**Język prowadzenia zajęć:**

polski

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

2

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

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

**Wymagania wstępne:**

Znajomość programowania w języku obiektowym i strukturalnym.

**Limit liczby studentów:**

120

**Cel przedmiotu:**

Celem przedmiotu jest wprowadzenie słuchaczy w podstawowe zagadnienia inżynierii oprogramowania, obejmujące organizację cyklu wytwarzania aplikacji, metodykę projektowania i weryfikacji programów oraz zarządzania projektem programistycznym. Związane z wykładem laboratorium pozwala studentom nabyć praktyczną umiejętność obiektowego modelowania i projektowania programów oraz planowania projektu.

**Treści kształcenia:**

Wprowadzenie (2). Zakres i cel inżynierii oprogramowania. Podstawowe działania i rozkład kosztów projektu. Projektowanie systemu a projektowanie programu. Kaskadowy i iteracyjny model cyklu życia oprogramowania. Metody i narzędzia.

Analiza strategiczna (2). Modelowanie procesów biznesowych. Studium wykonalności, procedura przetargu. Szacowanie kosztów projektu – metoda punktów funkcyjnych i model COCOMO.

Analiza i projektowanie obiektowe (12). Analiza i modelowanie wymagań metodą przypadków użycia (use cases): przypadki użycia, scenariusze, diagram, reguły biznesowe, schemat dokumentacji; przykład. Modelowanie dziedziny problemu: klasy, obiekty i ich relacje, diagram klas, perspektywy widzenia modelu; diagram maszyny stanowej; przykład. Modelowanie architektury w języku UML (Unified Modeling Language): diagramy pakietów, komponentów i rozmieszczenia; diagramy sekwencji i komunikacji; przeznaczenie modeli. Projektowanie architektury oprogramowania: analiza niewrażliwości (robustness analysis), warstwy danych, logiki biznesowej i prezentacji; modelowanie warstwy danych, projektowanie warstwy logiki biznesowej, wzorce projektowe; komunikacja między warstwami, technologie obiektowe, technologia EJB; przykład. Proces RUP; przykład.

Analiza i projektowanie strukturalne (2). Podstawowe pojęcia i modele strukturalne: hierarchia funkcji, diagram przepływu danych, diagram encji, diagram struktury. Wykorzystanie modeli w projekcie: ustalenie zakresu systemu, budowa modelu funkcjonalnego, specyfikowanie danych, budowa modelu implementacyjnego; przykład.

Testowanie oprogramowania (4). Weryfikacja i zatwierdzanie (verification and validation). Poziomy testowania: testowanie jednostkowe, integracyjne i systemowe; testowanie akceptacyjne. Organizacja procesu testowania: plan testów, specyfikacja testów, procedura testowania; usuwanie błędów. Metryki i metody projektowania testów i testowania; automatyzacja testów. Inne techniki weryfikacji: inspekcje kodu, przeglądy, dowody poprawności.

Zarządzanie projektem informatycznym (4). Struktura organizacyjna projektu. Planowanie projektu: struktura podziału pracy i struktura podziału produktu, tworzenie harmonogramu, narzędzia wspomagające. Tworzenie budżetu projektu. Plan projektu. Zarządzanie ryzykiem, zarządzanie biegiem projektu, metoda PRINCE2.

Kierunki rozwoju inżynierii oprogramowania (2). Metody zwinne. Architektura usługowa (SOA), korporacyjna magistrala usług.

Kolokwia zaliczeniowe (2).

**Metody oceny:**

Wykład: 2 kolokwia.
Laboratorium: ocena wyników pracy studenta.
Wymagane zaliczenie obydwu części przedmiotu.

**Egzamin:**

nie

**Literatura:**

1. Sacha K., Inżynieria oprogramowania, WNT, Warszawa 2010.
2. Fowler M., Scot K.: UML w kropelce; LTP 2002.
3. Booch G., Rumbaugh J., Jacobson I.: UML przewodnik użytkownika, WNT 2001.

**Witryna www przedmiotu:**

brak

**Uwagi:**

Przedmiot prowadzony w każdym semestrze (letnim i zimowym).
Przeznaczony dla wszystkich specjalności kierunku Informatyka (ISI oraz SID).

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt IOP\_W01:**

Ma podstawową wiedzę na temat ekonomicznych uwarunkowań projektów informatycznych

Weryfikacja:

Kolokwium

**Powiązane efekty kierunkowe:** K\_W04

**Powiązane efekty obszarowe:** T1A\_W08

**Efekt IOP\_W02:**

Ma podstawową wiedzę na z zakresu zarządzania projektami informatycznymi

Weryfikacja:

Kolokwium

**Powiązane efekty kierunkowe:** K\_W05

**Powiązane efekty obszarowe:** T1A\_W08, T1A\_W09

**Efekt IOP-W03:**

Ma uporządkowaną wiedzę ogólną z zakresu metod i narzędzi inżynierii oprogramowania oraz kierunków ich rozwoju

Weryfikacja:

Kolokwium

**Powiązane efekty kierunkowe:** K\_W13

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

**Efekt IOP\_W04:**

Ma podstawową wiedzę o cyklu życia systemów informatycznych i ich oprogramowania

Weryfikacja:

Kolokwium

**Powiązane efekty kierunkowe:** K\_W15

**Powiązane efekty obszarowe:** T1A\_W06

### Profil ogólnoakademicki - umiejętności

**Efekt IOP\_U01:**

Potrafi poznawać, analizować i modelować wymagania dla oprogramowania systemów informatycznych

Weryfikacja:

Kolokwium, ćwiczenia laboratoryjne

**Powiązane efekty kierunkowe:** K\_U14

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

**Efekt IOP\_U02:**

Potrafi projektować i modelować architekturę oprogramowania spełniającego wymagania użytkowników

Weryfikacja:

Kolokwium, ćwiczenia laboratoryjne

**Powiązane efekty kierunkowe:** K\_U15

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

**Efekt IOP\_U03:**

Potrafi przygotować specyfikację testów oprogramowania prostego systemu informatycznego

Weryfikacja:

Kolokwium, ćwiczenia laboratoryjne

**Powiązane efekty kierunkowe:** K\_U19

**Powiązane efekty obszarowe:** T1A\_U16

**Efekt IOP\_U04:**

Potrafi posłużyć się przynajmniej jedną metodę szacowania pracochłonności wytwarzania oprogramowania

Weryfikacja:

Kolokwium

**Powiązane efekty kierunkowe:** K\_U22

**Powiązane efekty obszarowe:** T1A\_U12

**Efekt IOP\_U05:**

Potrafi stworzyć plan prostego projektu informatycznego z uwzględnieniem harmonogramu i analizy ryzyka

Weryfikacja:

Kolokwium, ćwiczenia laboratoryjne

**Powiązane efekty kierunkowe:** K\_U23

**Powiązane efekty obszarowe:** T1A\_U16

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt K\_K01:**

Potrafi działać i pracować w zespole

Weryfikacja:

Ćwiczenia laboratoryjne

**Powiązane efekty kierunkowe:** K\_K03

**Powiązane efekty obszarowe:** T1A\_K03