Opis **zajęć (sylabus)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nazwa zajęć: | | Inżynieria oprogramowania | | | | | | | **ECTS** | | **4** |
| Nazwa zajęć w j. angielskim: | | Software engineering | | | | | | | | | |
| Zajęcia dla kierunku studiów: | | **Informatyka i Ekonometria** | | | | | | | | | |
|  | |  | | | | | | | | | |
| Język wykładowy: | | polski | | | | Poziom studiów: | | studia II stopnia | | | |
| Forma studiów: |  stacjonarne   niestacjonarne | Status zajęć: |  podstawowe   kierunkowe |  obowiązkowe   do wyboru | | Numer semestru: ……1….. | |  semestr zimowy  semestr letni | | | |
|  |  | Rok akademicki, od którego obowiązuje opis (rocznik): | | | | 2019/2020 | Numer katalogowy: | **ZIM-IE-2S-01Z-5** | | | |
|  | | | | | | | | | | | |
| Koordynator zajęć: | |  | | | | | | | | | |
| Prowadzący zajęcia: | |  | | | | | | | | | |
| Jednostka realizująca: | |  | | | | | | | | | |
| Jednostka zlecająca: | |  | | | | | | | | | |
| Założenia, cele i opis zajęć: | | Celem przedmiotu jest zapoznanie studentów z inżynierią oprogramowania i jej narzędzi.. Tematyka wykładów: Przewidzianych jest piętnaście wykładów. Obejmują one przedstawienie pojęć z inżynierii oprogramowania oraz narzędzi z tej dziedziny. W kolejności są prezentowane: metody zarządzania projektem informatycznym, w tym problematyka jakości i bezpieczeństwa, metody i narzędzia wspomagające tworzenie oprogramowanie (CASE), diagramy UML oraz zagadnienia związane z testowaniem oprogramowania, począwszy od debugowania programu, a skończywszy na testach akceptacyjnych.  Tematyka ćwiczeń laboratoryjnych:  Wykorzystanie wiedzy z wykładów do napisania i przetestowania aplikacji, włącznie z etapem negocjacji. Grupa laboratoryjna studentów pracuje wspólnie na zajęciach nad realizacją projektu. | | | | | | | | | |
| Formy dydaktyczne, liczba godzin: | | 1. wykład; liczba godzin ...15...; 2. ćwiczenia laboratoryjne; liczba godzin ...30...; 3. Konsultacje wykładowe; liczba godzin...10…; | | | | | | | | | |
| Metody dydaktyczne: | | wykład, dyskusja, ćwiczenia rachunkowe (szacowanie czasu potrzebnego na projekt IT), rysowanie diagramów UML, tworzenie kodu za pomocą narzędzi CASE, praca grupowa w nad projektem | | | | | | | | | |
| Wymagania formalne  i założenia wstępne: | | Wymagana jest wiedza z zakresu analizy matematycznej, rachunku prawdopodobieństwa, statystyki matematycznej oraz ekonometrii. | | | | | | | | | |
| Efekty uczenia się: | | Wiedza:   1. Zna problemy związane z zarządzaniem jakością, 2. Zna sposoby zarządzana czasem, w tym wykresy Gantta oraz grafy PERT, 3. Zna podstawy zarządzania wiedzą, 4. Zna podstawowe narzędzia CASE, 5. Zna najczęściej używane diagramy UML, 6. Zna metody debugowania i testowania aplikacji, | | | Umiejętności:   1. Umie zarządzać jakością projektu IT, 2. Potrafi szacować czas realizacji projektu za pomocą wykresu Gantta oraz grafów PERT, 3. Umie posługiwać się diagramami UML oraz narzędziami CASE, 4. Potrafi napisać test jednostkowy i integracyjny, 5. Potrafi prawidłowo znaleźć najczęściej popełniane błędy w kodzie, 6. Potrafi przeprowadzić testy systemowe i akceptacyjne. | | | | | Kompetencje:  ……………………..  …………………….. | |
| Sposób weryfikacji efektów uczenia się: | | Kolokwium: Wiedza(1-6), Umiejętności(2,4,5)  Ocena części projektu realizowanego na laboratorium (umiejętności od 1 do 6). | | | | | | | | | |
| Forma dokumentacji osiągniętych efektów uczenia się: | | Sprawozdania z kolejnych etapów realizacji projektu  Kolokwium z wykładu, ocena całego projektu | | | | | | | | | |
| Elementy i wagi mające wpływ  na ocenę końcową: | | **Oceny pracy nad projektem: C, ocena z kolokwium: B. Ocena końcowa K = ( (2\*B) + C)/3** | | | | | | | | | |
| Miejsce realizacji zajęć: | | Wykład -sala audytoryjna, ćwiczenia laboratoryjne – laboratorium komputerowe | | | | | | | | | |
| Literatura podstawowa i uzupełniająca:  Literatura podstawowa:   * Sacha Krzysztof, Inżynieria oprogramowania, Wydawnictwo Naukowe PWN, Warszawa 2010 * Butchter Paul, Debugowanie – Jak wyszukiwać i naprawiać błędy w kodzie oraz im zapobiegać, Helion 2010 * Wryczy Stanisława, Ćwiczenia UML 2.1, Helion 2006   Literatura uzupełniająca:   * Bereza -Jarociński Bogdan, Szomański Bolesław, Inżynieria oprogramowania, Helion 2009 * Schmuller Joseph, UML in 24 Hours, SAMS 2009, USA * 鶴保・征城、駒谷・昇一、「ずっと受けたかったソフトウェアエンジニアリングの授業①」、翔泳社、２０１０年 (Tsuruho Seishiro, Komaya Shōichi, Zdążyć na czas – kurs inżynierii oprogramowania, część 1, Shōeisha 2010, Japonia) * 鶴保・征城、駒谷・昇一、「ずっと受けたかったソフトウェアエンジニアリングの授業②」、翔泳社、２０１０年 (Tsuruho Seishiro, Komaya Shōichi, Zdążyć na czas – kurs inżynierii oprogramowania, część 2, Shōeisha 2010, Japonia) | | | | | | | | | | | |
| UWAGI  Skala ocen: 2(ndst); 3(dst); 3,5(dst+); 4(db); 4,5(db+); 5(bdb)  Przyporządkowanie wartości K:  Ocena ndst, gdy K < 2.75. Ocena dst, gdy K >= 2.75 i K <3.5. Ocena dst+, gdy K >= 3.5 i K <3.75. Ocena db, gdy K >=3.75 i K < 4.5. Ocena db+, gdy K >= 4.5 i K < 4.75. Ocena bdb gdy K >= 4.75. | | | | | | | | | | | |

Wskaźniki ilościowe charakteryzujące moduł/przedmiot:

|  |  |
| --- | --- |
| Szacunkowa sumaryczna liczba godzin pracy studenta (kontaktowych i pracy własnej) niezbędna dla osiągnięcia zakładanych dla zajęć efektów uczenia się - na tej podstawie należy wypełnić pole ECTS: | **120 h** |
| Łączna liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich lub innych osób prowadzących zajęcia: | **2 ECTS** |

Tabela zgodności kierunkowych efektów uczenia się z efektami przedmiotu:

|  |  |  |  |
| --- | --- | --- | --- |
| kategoria efektu | Efekty uczenia się dla zajęć: | Odniesienie do efektów dla programu studiów dla kierunku | Oddziaływanie zajęć na efekt kierunkowy\*) |
| Wiedza 1 | Zna problemy związane z zarządzaniem jakością | K\_W08 / P7S\_WG | 3 |
| Wiedza 2 | Zna sposoby zarządzana czasem, w tym wykresy Gantta oraz grafy PERT | K\_W12 / P7S\_WG | 2 |
| Wiedza 3 | Zna podstawy zarządzania wiedzą | K\_W12 / P7S\_WG | 2 |
| Wiedza 4 | Zna podstawowe narzędzia CASE | K\_W06 / P7S\_WG | 1 |
| Wiedza 5 | Zna najczęściej używane diagramy UML | K\_W08 / P7S\_WG | 3 |
| Wiedza 6 | Zna metody debugowania i testowania aplikacji | K\_W08 / P7S\_WG | 3 |
| Umiejętności 1 | Umie zarządzać jakością projektu IT |  |  |
| Umiejętności 2 | Potrafi szacować czas realizacji projektu za pomocą wykresu Gantta oraz grafów PERT | K\_U15 / P7S\_UW | 2 |
| Umiejętności 3 | Umie posługiwać się diagramami UML oraz narzędziami CASE | K\_U13 / P7S\_UW | 3 |
| Umiejętności 4 | Potrafi napisać test jednostkowy i integracyjny | K\_U14 / P7S\_UW | 3 |
| Umiejętności 5 | Potrafi prawidłowo znaleźć najczęściej popełniane błędy w kodzie | K\_U14 / P7S\_UW | 3 |
| Umiejętności 6 | Potrafi przeprowadzić testy systemowe i akceptacyjne | K\_U13 / P7S\_UW | 3 |
| Kompetencje - |  |  |  |
| Kompetencje - |  |  |  |

\*)

3 – zaawansowany i szczegółowy,

2 – znaczący,

1 – podstawowy,