Opis **zajęć (sylabus)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nazwa zajęć: | | Inżynieria oprogramowania | | | | | | | | **ECTS** | **5** |
| Nazwa zajęć w j. angielskim: | | Software engineering | | | | | | | | | |
| Zajęcia dla kierunku studiów: | | **Informatyka** | | | | | | | | | |
|  | |  | | | | | | | | | |
| Język wykładowy: | | polski | | | | Poziom studiów: | | | studia I stopnia | | |
| Forma studiów: |  stacjonarne   niestacjonarne | Status zajęć: | podstawowe   kierunkowe |  obowiązkowe   do wyboru | | Numer semestru: ……4….. | | |  semestr zimowy  semestr letni | | |
|  |  | Rok akademicki, od którego obowiązuje opis (rocznik): | | | | 2020/2021 | Numer katalogowy: | | **ZIM-IN-1Z-04L-20** | | |
|  | | | | | | | | | | | |
| Koordynator zajęć: | |  | | | | | | | | | |
| Prowadzący zajęcia: | |  | | | | | | | | | |
| Jednostka realizująca: | |  | | | | | | | | | |
| Jednostka zlecająca: | |  | | | | | | | | | |
| Założenia, cele i opis zajęć: | | Celem przedmiotu jest przedstawienie i wykorzystanie praktyczne inżynierii 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 ...18...; 2. ćwiczenia laboratoryjne; liczba godzin ...18...; 3. konsultacje z wykładowcą; liczba godzin ...8…; | | | | | | | | | |
| 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: | | Programowanie obiektowe, podstawy rachunku prawdopodobieństwa i statystyki | | | | | | | | | |
| 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, 7. Zna podstawy wzorców projektowych. | | | 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:   1. Student jest gotowy do pracy w projekcie zespołowym. | | | |
| Sposób weryfikacji efektów uczenia się: | | Egzamin pisemny oraz ocena za wkład własny studenta w projekt grupowy realizowany podczas zajęć. | | | | | | | | | |
| Forma dokumentacji osiągniętych efektów uczenia się: | | Sprawozdania z pracy, dokumentacja projektu oraz egzamin. | | | | | | | | | |
| Elementy i wagi mające wpływ  na ocenę końcową: | | **Egzamin – 70%, laboratorium – 30%** | | | | | | | | | |
| 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 * Ezust Alan, Ezust Paul, C++ i Qt, Wprowadzenie do wzorców projektowych, Wydanie II, Helion 2014 * Antani Ved, Stefanov Stoyan, Programowanie zorientowane obiektowo w języku JavaScript, Wydanie III, Helion 2018 * Zandstra Matt, PHP Obiekty, wzorce, narzędzia, Wydanie IV, Helion 2014   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  Minimalna liczba punktów konieczna do zaliczenia: 50. Punkty liczone są według następującej zasady:  Punkty z laboratorium wynikają z pracy nad projektem, która pochodzi z dwóch czynników. Pierwszym z nich jest budżet punktów zdobyty przez grupę za projekt. Budżet jest iloczynem liczby studentów w grupie oraz punktów za projekt i jest on w całości rozdysponowany pomiędzy studentów, według ich zaangażowania podczas tworzenia projektu. Istnieje możliwość zdobycia premii punktowej (w ocenie projektu), gdy jest on wykonany lepiej, niż zostało to określone w specyfikacji. Jednakże we wspomnianym budżecie punktów nie ma możliwości udzielenia kredytu, czyli dania komuś więcej punktów, niż to wynika z bilansu.  Na egzaminie można zdobyć maksymalnie 70 punktów. Liczba punktów zdobytych na laboratorium może być większa niż 30 (w przypadku premii). W przypadku, gdy student zdobędzie więcej niż 100 punktów, również otrzymuje ocenę 5,0 z przedmiotu. | | | | | | | | | | | |

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: | **150 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\_W10 / P6S\_WG | 2 |
| Wiedza 2 | Zna sposoby zarządzana czasem, w tym wykresy Gantta oraz grafy PERT. | K\_W10 / P6S\_WG | 2 |
| Wiedza 3 | Zna podstawy zarządzania wiedzą. | K\_W10 / P6S\_WG | 2 |
| Wiedza 4 | Zna podstawowe narzędzia CASE. | K\_W10 / P6S\_WG | 2 |
| Wiedza 5 | Zna najczęściej używane diagramy UML. | K\_W07 / P6S\_WG | 3 |
| Wiedza 6 | Zna metody debugowania i testowania aplikacji. | K\_W07/ P6S\_WG | 3 |
| Wiedza 7 | Zna podstawy wzorców projektowych. | K\_W07 / P6S\_WG | 3 |
| Umiejętności 1 | Umie zarządzać jakością projektu IT. | K\_U26 / P6S\_UW | 2 |
| Umiejętności 2 | Potrafi szacować czas realizacji projektu za pomocą wykresu Gantta oraz grafów PERT. | K\_U25 / P6S\_UW | 3 |
| Umiejętności 3 | Umie posługiwać się diagramami UML oraz narzędziami CASE. | K\_U12 / P6S\_UW | 3 |
| Umiejętności 4 | Potrafi napisać test jednostkowy i integracyjny. | K\_U20 / P6S\_UW | 3 |
| Umiejętności 5 | Potrafi prawidłowo znaleźć najczęściej popełniane błędy w kodzie. | K\_U20 / P6S\_UW | 3 |
| Umiejętności 6 | Potrafi przeprowadzić testy systemowe i akceptacyjne. | K\_U20 / P6S\_UW | 3 |
| Kompetencja 1 | Student jest gotowy do pracy w projekcie zespołowym. | K\_K06 / P6S\_KO | 3 |

\*)

3 – zaawansowany i szczegółowy,

2 – znaczący,

1 – podstawowy,