**Nazwa przedmiotu:**

Inżynieria oprogramowania 1

**Koordynator przedmiotu:**

Dr inż. Michał Okulewicz

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka i Systemy Informacyjne

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

1120-IN000-ISP-0353

**Semestr nominalny:**

5 / rok ak. 2020/2021

**Liczba punktów ECTS:**

4

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

1. godziny kontaktowe – 45h; w tym
 a) obecność na wykładach – 30h
 b) obecność na ćwiczeniach – 15h
2. praca własna studenta – 55 h; w tym
 a) zapoznanie się z literaturą – 10h
 b) przygotowanie do zajęć ćwiczeniowych – 15h
 c) przygotowanie do testów, rozwiązanie samodzielne zadań – 10h
 d) napisanie projektu, konsultacja – 20 h
Razem 100 h, co odpowiada 4 pkt. ECTS

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

1. obecność na wykładach – 30h
2. obecność na ćwiczeniach – 15h
Razem 45h, co odpowiada 2 pkt. ECTS

**Język prowadzenia zajęć:**

polski

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

1. obecność na ćwiczeniach – 15 h
2. przygotowanie do ćwiczeń – 15 h
3. przygotowanie do testów, rozwiązanie samodzielne zadań – 10h
4. napisanie projektu, konsultacja – 20 h
Razem 60h, co odpowiada 2 pkt. ECTS

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

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

**Wymagania wstępne:**

Programowanie 2 – obiektowe, Programowanie 3 – zaawansowane, Systemy operacyjne, Bazy danych, Projektowanie obiektowe

**Limit liczby studentów:**

.

**Cel przedmiotu:**

Celem przedmiotu jest przekazanie wiedzy dotyczącej podstawowych zasad i technik inżynierii oprogramowania oraz wykształcenie umiejętności tworzenia prostych modeli systemów informatycznych.

**Treści kształcenia:**

Wykład:
Pojęcia wstępne: projekt informatyczny a praca zespołowa, problemy z projektowaniem i realizacją systemów informatycznych, podstawowe potrzeby prowadzące do konieczności wykorzystania inżynierii oprogramowania.
Modele rozwoju oprogramowania i zarządzania wytwarzaniem produktu IT: kaskadowy, RUP, XP, SCRUM, DSDM.
Inżynieria wymagań: FURPS+, BPMN, User Stories.
Omówienie faz rozwoju projektu: planowanie, analiza (szacowanie złożoności oprogramowania na przykładzie Planning Poker), projektowanie (przygotowywanie pewnych dokumentów specyfikacji), implementowanie (wybór języka, sposób pracy z repozytorium kodu git, repozytoria pakietów), dokumentowanie (tworzenie dokumentacji), testowanie (testy jednostkowe i środowisko continuous integration), instalowanie (przygotowanie pakietów dla klienta), utrzymanie (przygotowywanie aplikacji do zmian i poprawek).
Projekt:
Zajęcia obejmują dyskusje związane z inżynierią wymagań oraz modelowaniem w UML w zakresie diagramów klas, diagramów przypadków użycia, diagramów stanu, aktywności i sekwencji. Konsultacje projektu obejmują weryfikację postępu prac oraz poprawność wykorzystanych modeli UML i kompletność opisu językowego systemu.

**Metody oceny:**

W czasie semestru odbywa się 7 dwugodzinnych zajęć projektowych. Podczas zajęć przeprowadzane jest 2-krotna punktowana weryfikacja postępów prac nad projektem. Na koniec semestru zajęć grupy studentów po cztery-pięć osób oddają projekt zadanego systemu informatycznego. Końcowa ocena składa się w 50% z ocen z testów teoretycznych i praktycznych przeprowadzonych na wykładzie oraz egzaminu ustnego i w 50% z oceny z wykonanej dokumentacji projektowej (w tym ocen cząstkowych wynikających z bieżących postępów w trakcie semestru). Uzyskanie przynajmniej 70% punktów z pisemnych testów teoretycznych i praktycznych uprawnia do zwolnienia z egzaminu ustnego.

**Egzamin:**

tak

**Literatura:**

1. K. Wiegers, J. Beatty, Software requirements, Pearson Education, 2013.
2. M. Belbin, Belbin team roles, Book Belbin Team Roles, 2004.
3. Business Process Model And Notation http://www.omg.org/spec/BPMN/2.0/
4. Unified Modeling Language 1.5 http://www.omg.org/spec/UML/1.5/
5. Unified Modeling Language 2.5 http://www.omg.org/spec/UML/2.5/
6. Manifesto for Agile Software Development http://agilemanifesto.org/
7. IBM Rational Unified Process https://www.ibm.com/developerworks/rational/library/1826.html
8. Scrum Guide http://www.scrumguides.org/scrum-guide.html
9. B. Chrabski, K. Zmitrowicz, Inżynieria wymagań w praktyce, PWN 2015.
10. Sz. Drejewicz, Zrozumieć BPMN modelowanie procesów biznesowych, Wydawnictwo Helion, 2012.
11. J. Żeliński, Analiza biznesowa: praktyczne modelowanie organizacji, Wydawnictwo Helion, 2017.
12. M. Belbin, Twoja rola w zespole, Gdańskie Wydawnictwo Psychologiczne, 2003.
13. GitFlow https://datasift.github.io/gitflow/IntroducingGitFlow.html
14. Jenkins CI https://www.jenkins.io/doc/

**Witryna www przedmiotu:**

e.mini.pw.edu.pl

**Uwagi:**

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W01:**

Zna język UML i sposoby stosowania go w praktyce.

Weryfikacja:

dwa testy, egzamin oraz projekt

**Powiązane charakterystyki kierunkowe:** K\_W12

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W02:**

Zna modele rozwoju oprogramowania, w tym modelu kaskadowego, spiralnego, odkrywczego wraz z poszczególnymi fazami oraz niekonwencjonalne metody wytwarzania oprogramowania: open-source, scrum, itd

Weryfikacja:

dwa testy i egzamin

**Powiązane charakterystyki kierunkowe:** K\_W15, K\_W07, K\_W09, K\_W12, K\_W14

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W03:**

Posiada podstawową wiedzę dotyczącą środowisk wytwórczych, pracy w zespole oraz narzędzi do pracy zespołowej

Weryfikacja:

dwa testy i egzamin

**Powiązane charakterystyki kierunkowe:** K\_W11, K\_W12

**Powiązane charakterystyki obszarowe:**

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U01:**

Potrafi stworzyć model systemu w języku UML obejmujący wymagania użytkownika oraz projekt rozwiązania

Weryfikacja:

dwa testy, egzamin oraz projekt

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U10, K\_U13, K\_U28

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U02:**

Potrafi zaprojektować prosty system informatyczny

Weryfikacja:

Projekt

**Powiązane charakterystyki kierunkowe:** K\_U23, K\_U26, K\_U28, K\_U30

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U03:**

Potrafi zastosować wybraną metodę oszacowania pracochłonności zadania

Weryfikacja:

obowiązkowy projekt realizowany w semestrze następnym

**Powiązane charakterystyki kierunkowe:** K\_U26

**Powiązane charakterystyki obszarowe:**

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka K01:**

Rozumie zagadnienia związane z pracą grupową

Weryfikacja:

projekt oraz obowiązkowy projekt realizowany w semestrze następnym

**Powiązane charakterystyki kierunkowe:** K\_K05

**Powiązane charakterystyki obszarowe:**

**Charakterystyka K02:**

Rozumie zasady negocjowania z klientem oraz prowadzenia wywiadu związanego z określeniem wymagań użytkownika

Weryfikacja:

projekt oraz obowiązkowy projekt realizowany w semestrze następnym

**Powiązane charakterystyki kierunkowe:** K\_K04

**Powiązane charakterystyki obszarowe:**

**Charakterystyka K03:**

Zna zagadnienia związane z jakością produktów informatycznych oraz konsekwencje szybkiego rozwoju nowych technologii w informatyce.

Weryfikacja:

projekt oraz obowiązkowy projekt realizowany w semestrze następnym

**Powiązane charakterystyki kierunkowe:** K\_K01, K\_K03, K\_K04

**Powiązane charakterystyki obszarowe:**