

KARTA PRZEDMIOTU DLA NABORU 2020/2021

INFORMACJE OGÓLNE

1. Nazwa przedmiotu kształcenia

TESTOWANIE OPROGRAMOWANIA

2. Nazwa jednostki prowadzącej moduł Zakład Informatyki

3. Grupa treści kształcenia -

4. Typ przedmiotu obowiązkowy

5. Poziom studiów pierwszego stopnia

6. Liczba punktów ECTS 2

7. Poziom przedmiotu średniozaawansowany

8. Rok studiów, semestr IV rok, semestr VII

9. Liczba godzin w semestrze laboratoria 15h, wykłady 15 h

10. Język wykładowy polski

11. Wykładowca (wykładowcy)

Marcin Klimek, dr inż.

INFORMACJE SZCZEGÓŁOWE

12. Wymagania wstępne

1. Posiadanie podstawowej wiedzy na temat metod i narzędzi inżynierii oprogramowania.
2. Posiadanie podstawowej wiedzy programistycznej
3. Wiedza z zakresu baz danych, logiki

13. Cele przedmiotu

C1. nabycie umiejętności tworzenia i zarządzania testami.

C2. zapoznanie studentów z podstawami testowania oprogramowania, a także z metodami, technikami i narzędziami inżynierii jakości

14. Efekty uczenia się w zakresie wiedzy, umiejętności i kompetencji społecznych

Student, który zaliczył przedmiot potrafi:	odniesienie do kierunkowych efektów uczenia się
--	---

WIEDZA

EU01 zna standardy i dobre praktyki wykonywania testów	I1P_W13 I1P_W14
EU02 zna zagadnienia związane z technikami testowania oprogramowania, podziałem testów i dostępnymi standardami testowania oprogramowania.	
EU03 zna zasady zarządzania rozległymi projektami informatycznymi.	

UMIEJĘTNOŚCI

EU04 umie testować oprogramowanie oraz stosuje właściwe testy na odpowiednim etapie procesu wytwórczego oprogramowania	I1P_U03 I1P_U09 I1P_U14 I1P_U18 I1P_U20
EU05 umie sporządzać dokumentację specyfikującą testy i raporty o ich przebiegu.	
EU06 posiada umiejętność weryfikacji projektów informatycznych pod kątem jakościowym	
EU07 potrafi stosować narzędzia testujące w procesach wytwarzania oprogramowania.	

KOMPETENCJE SPOŁECZNE

EU08 potrafi zespołowo przeprowadzić testy	I1P_K03 I1P_K04
EU09 zna kodeks etyczny i zagadnienia etyczne związane z pracą testera	

15. Treści programowe

Forma zajęć – wykłady

Wprowadzenie do testowania. Testowanie w cyklu życia oprogramowania.
Charakterystyka i podział testów oprogramowania: Błędy i awarie. Testy funkcjonalne i testy właściwości. Testy białoskrzynkowe (strukturalne) i czarnoskrzynkowe (funkcjonalne), testy poprawek błędów, testy regresywne.
Testowanie statyczne i dynamiczne.
Zastosowania testów w procesie rozwoju oprogramowania: Poziomy testowania oprogramowania. Testy jednostkowe, testy integracyjne wewnętrzne, testy systemowe, testy integracyjne zewnętrzne, testy akceptacyjne.
Standardy w testowaniu oprogramowania.

Testowanie w fazie utrzymania: Testowanie po rozbudowie i udoskonalaniu systemu. Modyfikacje naprawcze i awaryjne. Systemy raportowania i śledzenia błędów. Proces usuwania błędów Techniki scalania rozwiązań informatycznych: Integracja aplikacji, integracja danych, integracja sieciowa, integracja systemowa. Testowanie niefunkcjonalne.	
Forma zajęć – laboratoria	
Statyczna inspekcja i przeglądy kodu: Sprawdzanie kodu zgodnie z listą potencjalnych błędów. Projektowanie przypadków testowych w testach czarno skrzynkowych. Projektowanie przypadków testowych w testach biało skrzynkowych. Testy jednostkowe. Przygotowanie planu testów i raportu testów. Wersje oprogramowania i testy akceptacyjne. Testowanie aplikacji internetowych i systemów rozproszonych. Wykorzystanie narzędzi do automatyzacji: xUnit, Selenium, JIRA (zarządzanie incydentami) itp. Testowanie eksploracyjne Analiza statyczna i dynamiczna Testowanie niefunkcjonalne (testy użyteczności, testy wydajności)	
16. Narzędzia/metody dydaktyczne	
1. Podczas laboratoriów praca indywidualna studentów: pisanie testów np. jednostkowych. Studenci samodzielnie rozwiązują zadany problem praktyczny, dobierając odpowiednie narzędzia . a) dyskusje wokół tematyki poruszanej na wykładzie, b) praktyczne projektowanie przypadków testowych, c) manualne wykonywanie testów, d) automatyzacja wykonania testów. 2. Laboratoria w części odbywają się przy tablicy: studenci np. analizują różne strategie dotyczące testowania oprogramowanie 3. Wykład prowadzony z wykorzystaniem projektora multimedialnego. Wykłady przygotowane w formie prezentacji PowerPoint. 4. Treści prezentowane na wykładzie są wzbogacone o pokazy praktycznych rozwiązań odnoszących się do prezentowanych zagadnień.	
17. Sposoby oceny (F – formująca; P – podsumowująca)	
F1. Kolokwium przy komputerze: implementacja testów jednostkowych	
F2. Kartkówki na laboratoriach	
F3. Ocena realizacji projektu zaliczeniowego.	
F4. Ocena ciągła pracy indywidualnej podczas laboratoriów.	
P1. Egzamin w formie pisemnej sprawdzający wiedzę z wykładów i laboratoriów	
18. Obciążenia pracą studenta	
forma aktywności	średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe z nauczycielem*	30+15
Nakład pracy studenta	
Przygotowanie się do zajęć	10
Przygotowanie projektu zaliczeniowego	10
Przygotowanie do egzaminu	10
SUMA	75
SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU	2
19. Literatura podstawowa i uzupełniająca	
Literatura podstawowa:	
1) Janusz Górski, Cezary Orłowski, Inżynieria Oprogramowania w procesach integracji systemów informatycznych , PWNT, 2010.	
2) Praca zbiorowa pod redakcją Jana Werewki, Zarządzanie projektami w przedsiębiorstwie informatycznym, AGH UWND, 2012.	
3) Adam Roman, Testowanie i jakość oprogramowania. Modele, techniki, narzędzia, PWN 2015.	
4) Stephen H. Kan, Metryki i modele w inżynierii jakości oprogramowania, PWN 2006	
Literatura uzupełniająca:	
1) Roy Osheroe, Testy jednostkowe - świat niezawodnych aplikacji , Helion 2014	

- 2) Glenford Myers i in., Sztuka testowania oprogramowania, 2005
- 3) Black Mitchell, Advanced Software Testing, Helion 2010

20. Formy oceny - szczegóły

Warunki uzyskania zaliczenia przedmiotu: zajęcia kończą się zaliczeniem z oceną.

Sposób weryfikacji efektów uczenia się:

Ocena stopnia osiągniętych przez studenta efektów uczenia się następuje wg poniższych kryteriów:

- 5.0 – zakładany efekt uczenia się został osiągnięty bez zastrzeżeń
- 4.5 – zakładany efekt uczenia się został osiągnięty z pojedynczymi brakami/błędami
- 4.0 – zakładany efekt uczenia się został osiągnięty z nielicznymi brakami/błędami
- 3.5 – zakładany efekt uczenia się został osiągnięty z wieloma brakami/błędami
- 3.0 – zakładany efekt kształcenia został osiągnięty z licznymi i istotnymi brakami/błędami (minimalnie wymagany poziom osiągnięcia efektu)
- 2.0 – zakładany efekt uczenia się nie został osiągnięty

21. Inne przydatne informacje o przedmiocie

1. Bezpośrednich informacji o problematyce zajęć i treściach programowych udziela prowadzący w trakcie zajęć i podczas konsultacji
2. Zajęcia odbywać się będą w PSW w Białej Podlaskiej
3. Zajęcia odbywać się będą zgodnie z aktualnym planem zajęć
4. Konsultacje odbywać się będą zgodnie z obowiązującym terminarzem
5. Materiały z wykładów i laboratoriów udostępnione są dla studentów na stronie psw.na11.pl
6. Przedmiot przygotowuje do pracy na stanowiskach tj. Junior Tester, Software Quality Engineer czy Test Automation Engineer.

* Zajęcia prowadzone z bezpośrednim udziałem nauczyciela akademickiego lub innej osoby prowadzącej zajęcia oraz konsultacje