

KARTA KURSU (realizowanego w specjalności)**Inżynieria oprogramowania**

(nazwa specjalności)

Nazwa	Jakość i testowanie oprogramowania
Nazwa w j. ang.	Software Quality and Testing

Koordinator	mgr inż. Piotr Kukula	Zespół dydaktyczny
		mgr inż. Piotr Kukula
Punktacja ECTS*	st. stacjonarne: 4 st. niestacjonarne: 4	

Opis kursu (cele kształcenia)

Celem kursu jest zapoznanie studentów z procesem produkcji oprogramowania, w szczególności z procesami zapewnienia jakości oraz testowaniem we wszystkich jego fazach, poczynając od fazy analizy wymagań, a kończąc na testowaniu gotowego produktu. Kurs prowadzony jest w języku polskim.

Warunki wstępne

Wiedza	Podstawowe wiadomości z zakresu: algorytmiki, systemów operacyjnych oraz inżynierii oprogramowania. Znajomość języków C#, JAVA lub python.
Umiejętności	Umiejętność konstruowania prostych algorytmów. Umiejętność programowania obiektowego w języku C#, JAVA lub Python.
Kursy	Algorytmy i struktury danych. Systemy operacyjne. Programowanie obiektowe, Inżynieria oprogramowania

Efekty uczenia się

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Wiedza	Po zakończeniu kursu student ma wiedzę na temat:	
	W01: konieczności testowania oprogramowania, oraz terminologii testowania i celów.	S1_W01
	W02: procesów wytwarzania oprogramowania i ich modeli, modeli (paradygmaty) tworzenia oprogramowania (modele cyklu życia oprogramowania).	S1_W02
	W03: narzędzi i środowiska wspomagającego tworzenie oprogramowania i testowania.	S1_W03

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Umiejętności	Po zakończeniu kursu student: U01: umie specyfikować wymagania stawiane oprogramowaniu oraz przeprowadzać ich przegląd. U02: dokonuje inspekcji kodu. Potrafi stosować podstawowe metody i strategie testowania oprogramowania, a także umie sporządzić plan i ewaluację testowania. U03: potrafi używać narzędzi do testowania automatycznego	S1_U01 S1_U04 S1_U05

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Kompetencje społeczne	Po zakończeniu kursu student: K01: potrafi określić możliwości wykorzystywania swojej wiedzy dotyczącej testowania oprogramowania w pracy zawodowej. K02: rozumie konieczność kształcenia ustawicznego w szczególności w związku z dynamicznym rozwojem informatyki i nowych technologii. K03: potrafi współdziałać i pracować w grupie (zespole projektowym, programistycznym), przy jednoczesnym przestrzeganiu zasad etykiety zawodowej i netykiety	S1_K01 S1_K02 S1_K05

Studia stacjonarne

Organizacja						
Forma zajęć	Wykład (W)	Ćwiczenia w grupach				
		A	K	L	S	P
Liczba godzin				25		

Studia niestacjonarne

Organizacja						
Forma zajęć	Wykład (W)	Ćwiczenia w grupach				
		A	K	L	S	P
Liczba godzin				20		

Opis metod prowadzenia zajęć

Podczas pracy laboratoryjnej studenci będą rozwiązywać problemy zadane przez prowadzącego zajęcia oraz opracowywać własne zadania testowania kodu i oprogramowania. Na ćwiczeniach na bieżąco weryfikowana będzie wiedza przekazywana podczas ćwiczeń. Integralną częścią zajęć jest realizacja projektów informatycznych ze szczególnym uwzględnieniem testowania kodu i oprogramowania.

Formy sprawdzania efektów uczenia się

	E – learning	Gry dydaktyczne	Ćwiczenia w szkole	Zajęcia terenowe	Praca laboratoryjna	Projekt indywidualny	Projekt grupowy	Udział w dyskusji	Referat	Praca pisemna (esej)	Egzamin ustny	Egzamin pisemny	Zadania problemowe
W01					x	x		x					
W02					x								
W03					x								
W04					x								
U01					x			x					
U02					x			x					
U03					x			x					
K01								x					
K02								x					
K03								x					

Kryteria oceny	Ocena końcowa zależna od ocen częściowych oraz systematyczności realizowanych zadań. Zaliczenie na ocenę dostateczną otrzymuje student, który potrafi: - student zna i umie stosować podstawowe techniki testowania oprogramowania, - student umie posługiwać się narzędziami do testowania automatycznego, - student umie zrealizować projekt informatyczny zawierający testy oprogramowania wskazane przez prowadzącego Zaliczenie na ocenę dobrą lub bardzo dobrą otrzymuje student, który spełnia warunki oceny dostatecznej, a oprócz tego także: - systematycznie realizuje bieżące zadania i projekty,
----------------	--

Uwagi	
-------	--

Treści merytoryczne (wykaz tematów)

1. Istota testowania oprogramowania i jej główne cele.
2. Typy testów (funkcjonalne, нефункционалне, pielęgnacja testów).
3. 7 zasad testowania
4. V model (SDLC & STLC)
5. Co to jest STLC ? (Software Testing Life Cycle). Planowanie testów
6. Testy ręczne (Manual tests)
7. Testy automatyczne
8. Omówienie narzędzi do testowania
9. Testy jednostkowe (Unit testing)
10. Testy integracyjne, testy systemowe (System Testing)
11. Smoke Testing vs Sanity Testing
12. Testowanie regresyjne
13. Testowanie нефункционалне
14. Co to jest scenariusz testowy (What is Test Scenario)?
15. Jak pisać przypadki testowe (test cases)
16. Przegląd wybranych narzędzi do testowania automatycznego
17. Technika TDD (Test Driven Development)

Wykaz literatury podstawowej

1. Ian Sommerville, Inżynieria Oprogramowania, WNT- wybrane fragmenty
2. K.Beck, A.Cynthia, Wydajne programowanie – Extreme Programming, Mikom, 2005 - wybrane fragmenty
3. R. Pressman, Software Engineering, McGraw-Hill, New York 1997 - wybrane fragmenty
4. Adam Roman, Testowanie i jakość-oprogramowania.-Metody-narzędzia-techniki, PWN 2015
5. Kent Beck, TDD. Sztuka tworzenia dobrego kodu, Helion
6. Dustin, Elfriede, Effective Software Testing. Addison Wesley. p. 3. ISBN 0-201-79429-2

Wykaz literatury uzupełniającej

1. <http://testerzy.pl/baza-wiedzy>
2. <http://sjsi.org/slownik-testerski>

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) – **studia stacjonarne**

Liczba godzin w kontakcie z prowadzącymi	Wykład	
	Konwersatorium (ćwiczenia, laboratorium itd.)	25
	Pozostałe godziny kontaktu studenta z prowadzącym	15
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	15
	Realizacja zadań domowych (problemowych) po zapoznaniu się z niezbędną literaturą przedmiotu	
	Przygotowanie projektu lub prezentacji na podany temat (praca indywidualna lub w grupie)	25
	Przygotowanie do egzaminu/zaliczenia	20
Ogółem bilans czasu pracy		100
Liczba punktów ECTS w zależności od przyjętego przelicznika		4

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) – **studia niestacjonarne**

Liczba godzin w kontakcie z prowadzącymi	Wykład	
	Konwersatorium (ćwiczenia, laboratorium itd.)	20
	Pozostałe godziny kontaktu studenta z prowadzącym	10
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	25
	Realizacja zadań domowych (problemowych) po zapoznaniu się z niezbędną literaturą przedmiotu	
	Przygotowanie projektu lub prezentacji na podany temat (praca indywidualna lub w grupie)	25
	Przygotowanie do egzaminu/zaliczenia	20
Ogółem bilans czasu pracy		100
Liczba punktów ECTS w zależności od przyjętego przelicznika		4