

Nazwa modułu kształcenia	Testowanie Oprogramowania		
Nazwa jednostki prowadzącej moduł	Instytut Informatyki, Wydział Matematyki i Informatyki		
Kod modułu	WMI.II- P2-TO		
Język kształcenia	Polski		
Efekty kształcenia dla modułu kształcenia	Symbol	Efekty kształcenia	Odniesienie do efektów kierunkowych
	E1	Rozumie miejsce procesu testowego w ramach procesu wytwarzania oprogramowania	K_W04, K_U18
	E2	Rozumie rolę i znaczenie testowania w procesie zapewniania jakości, zwłaszcza w kontekście analizy ryzyka	K_W14, K_U18, K_K04
	E3	Zna podstawowe pojęcia z zakresu testowania oraz zapewniania jakości	K_W01, K_W03, K_U18
	E4	Zna klasyczne techniki projektowania testów	K_W01, K_W10, K_U01
	E5	Potrafi wdrożyć i przeprowadzić proces testowy w organizacji	K_U03, K_U05, K_U18, K_K02
	E6	Potrafi ustnie i pisemnie uzasadnić znaczenie testowania, raportować postępy testów oraz ma podstawową wiedzę w zakresie zarządzania procesem testowym	K_U05, K_U11, K_U20
	E7	rozumie i docenia znaczenie wartości intelektualnej w działaniach własnych oraz innych osób, rozumie potrzebę ciągłego udoskonalania własnych kwalifikacji testerskich	K_U23, K_K01, K_K05
Typ modułu kształcenia (obowiązkowy/fakultatywny)	Specjalnościowy dla specjalności "Inżynieria Oprogramowania", "Informatyka Stosowana" oraz "Informatyka Teoretyczna"		
Rok studiów	III		
Semestr	6		
Imię i nazwisko osoby/osób prowadzących moduł	dr Adam Roman, mgr Michał Mních		
Imię i nazwisko osoby/osób egzaminującej/egzaminujących	dr Adam Roman		

bądź udzielającej zaliczenia, w przypadku gdy nie jest to osoba prowadząca dany moduł	
Sposób realizacji	wykład, laboratorium
Wymagania wstępne i dodatkowe	Wymagana jest podstawowa wiedza z zakresu inżynierii oprogramowania, matematyki dyskretnej, języków formalnych, rachunku prawdopodobieństwa, programowania
Liczba godzin zajęć dydaktycznych wymagających bezpośredniego udziału nauczyciela akademickiego i studentów, gdy w danym module przewidziane są takie zajęcia	60
Liczba punktów ECTS przypisana modułowi	6
Bilans punktów ECTS	<p>Udział w wykładach - 30 godz.</p> <p>Udział w zajęciach laboratoryjnych – 30 godz.</p> <p>Samodzielna implementacja zadań programistycznych – 50 godz.</p> <p>Przygotowanie do egzaminu, analiza programów do egzaminu oraz udział w egzaminie – 30 godz.</p> <p>Łączny nakład pracy studenta: 140 godzin , co odpowiada 6 punktom ECTS</p>
Stosowane metody dydaktyczne	<ol style="list-style-type: none"> 1. Wykład ilustrowany prezentacją komputerową. 2. Ćwiczenia w laboratorium komputerowym, połączone z dyskusją przy tablicy. 3. Projektowanie i wykonywanie testów, budowa modeli jakości
Metody sprawdzania i oceny efektów kształcenia uzyskanych przez studentów	<p>Zadania domowe (E1, E2 E3,E4, E5)</p> <p>Projekty zaliczeniowe (E1, E2, E3, E4, E5)</p> <p>Egzamin (E1, E4, E6, E7)</p> <p>Prezentowanie rozwiązań zadań domowych oraz ustna obrona projektu zaliczeniowego (E6, E7, E8)</p>
Forma i warunki zaliczenia modułu, w tym zasady dopuszczenia do egzaminu,	Student otrzymuje ocenę końcową z ćwiczeń na podstawie punktów przyznawanych za systematycznie oddawane zadania domowe, oraz projektu zaliczeniowego. Maksymalna liczba punktów to 35. Maksymalna liczba punktów za egzamin to 65 punktów. Student jest dopuszczony do egzaminu niezależnie od liczby

zaliczenia, a także forma i warunki zaliczenia poszczególnych zajęć wchodzących w zakres danego modułu	zdobytych punktów na laboratoriach. Skala punktowa dla zaliczenia całości przedmiotu: 0-64 ndst 65-71 dst 72-77 +dst 78-84 db 85-90 +db 91-100 bdb
Treści modułu kształcenia	<ol style="list-style-type: none">1. Wprowadzenie do testowania<ol style="list-style-type: none">1.1. Dlaczego testowanie jest niezbędne1.2. Definicja testowania1.3. Ogólne zasady testowania1.4. Podstawowy proces testowy1.5. Psychologia testowania2. Testowanie w cyklu życia oprogramowania<ol style="list-style-type: none">2.1. Modele wytwarzania oprogramowani2.2. Poziomy testów2.3. Typy testów2.4. Testowanie pielęgnacyjne2.5. Metryki i miary3. Statyczne techniki testowania<ol style="list-style-type: none">3.1. Przeglądy nieformalne3.2. Przegląd3.3. Przegląd techniczny3.4. Inspekcja3.5. Kryteria sukcesu przeglądów3.6. Analiza statyczna3.7. Analiza dynamiczna4. Techniki projektowania testów<ol style="list-style-type: none">4.1. Techniki oparte na specyfikacji (czarnoskrzynkowe)

	<ul style="list-style-type: none">4.2. Techniki oparte na strukturze (białoskrzynkowe)4.3. Techniki oparte na doświadczeniu4.4. Techniki oparte na defektach4.5. Wybór technik testowania5. Testowanie właściwości oprogramowania<ul style="list-style-type: none">5.1. Atrybuty jakościowe do testowania dziedzinowego5.2. Atrybuty jakościowe do testowania technicznego6. Zarządzanie testowaniem<ul style="list-style-type: none">6.1. Organizacja testów6.2. Planowanie i szacowanie testów6.3. Monitorowanie postępu testów i nadzór6.4. Zarządzanie konfiguracją6.5. Ryzyko a testowanie6.6. Zarządzanie incydentami7. Testowanie wspierane narzędziami: narzędzia testowe i automatyzacja testów<ul style="list-style-type: none">7.1. Konceptcje związane z narzędziami testowymi7.2. Typy (kategorie) narzędzi testowych7.3. Wdrażanie narzędzi w organizacji8. Standardy i udoskonalanie procesu testowego<ul style="list-style-type: none">8.1. Uznane standardy8.2. Usprawnianie procesu testowania8.3. Doskonalenie procesu testowania9. Umiejętności interpersonalne - skład zespołu<ul style="list-style-type: none">9.1. Umiejętności indywidualne9.2. Dynamika zespołu testowego9.3. Dopasowanie testowania do organizacji9.4. Motywacja9.5. Komunikacja10. Jakość oprogramowania<ul style="list-style-type: none">10.1. Modele procesu produkcji oprogramowania w kontekście jakości
--	--

	<p>10.2. Podstawy teorii pomiarów 10.3. Metryki jakości oprogramowania 10.4. Siedem narzędzi Ishikawy 10.5. Efektywność usuwania defektów 10.6. Model Rayleigha 10.7. Modele przyrostu niezawodności 10.8. Pomiar i analiza zadowolenia klienta</p>
Wykaz literatury podstawowej i uzupełniającej, obowiązującej do zaliczenia danego modułu	<p>Moduł ma charakter autorski, obowiązuje przede wszystkim materiał wyłożony, literatura ma charakter pomocniczy.</p> <ul style="list-style-type: none">• Offutt, Ammann – Introduction to Software Testing• Kan – Metryki i modele w inżynierii jakości oprogramowania• Myers – Sztuka testowania oprogramowania• Hunt – JUnit. Pragmatyczne testy jednostkowe w Javie• Binder – Testowanie systemów obiektowych• Patton – Testowanie oprogramowania
Metody i kryteria oceniania	<p>Ustalone i podane na początku zajęć przez prowadzących ćwiczenia. Kryteria oceny dot. zaliczania całości przedmiotu podane są wyżej.</p>
Wymiar, zasady i forma odbywania praktyk, w przypadku, gdy program kształcenia przewiduje praktyki	<p>Nie dotyczy</p>