

## ANALIZA I TESTOWANIE SYSTEMÓW INFORMATYCZNYCH

Wydział Informatyki			
Nazwa programu kształcenia (kierunku)	<b>Informatyka</b>		Poziom i forma studiów <b>drugiego stopnia niestacjonarne</b>
Specjalność	<b>Systemy Inteligentne</b>		Ścieżka dyplomowania <b>2025/2026L - 2026/2027L</b>
Nazwa przedmiotu	<b>Analiza i testowanie systemów informatycznych</b>		Kod przedmiotu <b>INZ2ATS</b>
Rodzaj przedmiotu	<b>obowiązkowy</b>	<b>Semestr 1</b>	Punkty ECTS <b>4</b>
Liczba godzin w semestrze	<b>W - 10 Ćw - 0 PS - 20 P - 0 L - 0 S - 0</b>		
Przedmioty wprowadzające			
Założenia i cele przedmiotu	<p>Celem przedmiotu jest przekazanie studentom podstawowej wiedzy dotyczącej inżynierskich pryncypiów projektowania i testowania systemów programowych. Studenci stosują poznane pryncypia w grupowym projekcie w którym implementują system programowy (powyżej 10K rozkazów C++) stosując "agile" iteracyjny proces rozwoju programów. Studenci uczą się fundamentalnych zasad projektowania programów, projektowania architektur i interfejsów modułów programowych; rozpatrywania alternatywnych rozwiązań projektowych i uzasadniania wybranych decyzji projektowych; testów czarnej i białej skrzynki, testowania specyfikacji wymagań, narzędzi do automatycznego testowania, pisanie planu testów, raportowania wyników z testowania. Student wykształci umiejętności: projektowania architektury systemu i interfejsów modułów; pracy w grupie; planowania i realizacji iteracyjnego procesu rozwoju programów; planowania, testowania, pisanie przypadków testowych, dokumentowania wyników z przeprowadzonych testów, używania narzędzi wspierających proces testowy. Celem przedmiotu jest również wykształcenie umiejętności skutecznego komunikowania się - na piśmie i w dyskusji - w zakresie odnajdywanych problemów z innymi członkami zespołu projektowego.</p>		
Forma zaliczenia	<p>Wykład - egzamin pisemny. Pracownia specjalistyczna - sprawozdania, projekt.</p>		
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> <li>Zasady projektowania programów, projektowania architektur i interfejsów modułów programowych.</li> <li>Rozpatrywanie i uzasadnianie alternatywnych rozwiązań projektowych; "agile" iteracyjny proces rozwoju programów.</li> <li>Testy czarnej i białej skrzynki.</li> <li>Testy integracyjne i systemowe testowanie.</li> <li>Narzędzia do automatycznego testowania, pisanie planu testów, raportowania wyników z testowania.</li> </ol> <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> <li>Praca projektowa z zastosowaniem metod wyjaśnianych w czasie wykładu.</li> <li>Praca w grupie i komunikowanie się - na piśmie i w dyskusji - w zakresie odnajdywanych problemów z innymi członkami zespołu projektowego.</li> <li>Projekt architektury, implementacja, testowanie.</li> </ol>		
Metody dydaktyczne	metoda projektów, ćwiczenia przedmiotowe, programowanie z użyciem komputera, burza mózgów, metoda sytuacyjna, klasyczna metoda problemowa, wykład problemowy,		
Efekty kształcenia	Zapisać minimum 4, maksimum 8 efektów kształcenia zachowując kolejność: wiedza-umiejętności-kompetencje. Każdy efekt kształcenia musi być weryfikowalny.		Odniesienie do kierunkowych efektów kształcenia
EK1	zna i rozumie miejsce testowania w cyklu życia oprogramowania i potrafi stosować w praktyce		INF2_W02 INF2_W08 INF2_U10
EK2	zna i rozumie podstawowe metody i techniki testowania oprogramowania i potrafi testować oprogramowanie w praktyce		INF2_W03 INF2_W05 INF2_U04 INF2_U08
EK3	potrafi pracować w grupie, przyjmując w niej różne role		INF2_U13
EK4	zna i potrafi stosować w praktyce metody projektowania i weryfikacji architektury i implementacji dużych systemów w obliczu wysokich wymagań niezawodności		INF2_W05 INF2_W06 INF2_W08 INF2_U04 INF2_U08
EK5	ma umiejętności z zakresu soft skills		INF2_U13 INF2_K04
EK6	potrafi projektować i używać interfejsy modułowe		INF2_W03 INF2_U03 INF2_U08
EK7	potrafi planować i realizować iteracyjny proces rozwoju programów		INF2_U04 INF2_U07
Nr efektu kształcenia	Metoda weryfikacji efektu kształcenia		Forma zajęć (jeśli jest więcej niż jedna), na której zachodzi weryfikacja
EK1	egzamin pisemny, sprawozdania		W, Ps
EK2	egzamin pisemny, sprawozdania		W, Ps
EK3	ocena projektu wykonywanego na zajęciach		Ps
EK4	egzamin pisemny, ocena sprawozdań		W, Ps

EK5	ocena sprawozdań		Ps
EK6	ocena sprawozdań		Ps
EK7	ocena sprawozdań		Ps
Bilans nakładu pracy studenta (w godzinach)	1 - Udział w wykładach	10x1h	10
	2 - Udział w pracowni specjalistycznej	10x2h	20
	3 - Praca projektowa (wykonywanie zadań domowych)		50
	4 - Pisanie raportu projektowego i przygotowanie prezentacji		10
	5 - Udział w konsultacjach		2
	6 - Przygotowanie do egzaminu		6
	7 - Udział w egzaminie		2
		<b>RAZEM:</b>	<b>100</b>
Wskaźniki ilościowe	Nakład pracy studenta związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela: (5)+(1)+(2)+(7)	34	<b>ECTS</b> 1.4
	Nakład pracy związany z zajęciami o charakterze praktycznym: (2)+(4)+(3)	80	3.2
Literatura podstawowa	1. R. Patton: Testowanie oprogramowania. Wydawnictwo MIKOM, 2002. 2. B. Wiszniewski, B. Bereza-Jarociński: Teoria i praktyka testowania programów. Wydawnictwo PWN, 2006. 3. S. Jarzabek: Podręcznik projektu inżynierskiego (PDF udostępniany przez prowadzącego), 2020. 4. R. Pressman, B. Maxim: Praktyczne podejście do inżynierii oprogramowania, WNT, 2004.		
Literatura uzupełniająca	1. J. Nielsen, H. Loranger: Optymalizacja funkcjonalności serwisów internetowych. Wydawnictwo Helion, 2007. 2. M.L. Hutcheson: Software Testing Fundamentals: Methods and Metrics. Wiley Publishing, Inc., 2003. 3. D. Graham, E. Van Veenendaal, I. Evans, R. Black: Foundations of Software Testing: ISTQB Certification. Cengage Learning, Inc., 2008.		
Jednostka realizująca	Katedra Oprogramowania	Program opracował(a)	Wpisać osobę, która opracowała program
Data opracowania programu	2020.05.22		dr hab. Stanisław Jarzabek