

SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA 2024-2028

Rok akademicki 2026/2027

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	Testowanie i wdrażanie systemów informatycznych
Kod przedmiotu*	
nazwa jednostki prowadzącej kierunek	Instytut Informatyki, Wydział Nauk Ścisłych i Technicznych
Nazwa jednostki realizującej przedmiot	Instytut Informatyki, Wydział Nauk Ścisłych i Technicznych
Kierunek studiów	Informatyka i ekonometria
Poziom studiów	studia pierwszego stopnia
Profil	praktyczny
Forma studiów	studia stacjonarne
Rok i semestr/y studiów	rok III, sem. VI
Rodzaj przedmiotu	przedmiot specjalnościowy
Język wykładowy	polski
Koordynator	dr inż. Dariusz Bober
Imię i nazwisko osoby prowadzącej / osób prowadzących	dr inż. Dariusz Bober

* -opcjonalnie, zgodnie z ustaleniami w Jednostce

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt. ECTS
6	10			10					2

1.2. Sposób realizacji zajęć

- ☒ zajęcia w formie tradycyjnej
☒ zajęcia realizowane z wykorzystaniem metod i technik kształcenia na odległość

1.3 Forma zaliczenia przedmiotu (z toku) (egzamin, zaliczenie z oceną, zaliczenie bez oceny)
Zaliczenie z Oceną**2. WYMAGANIA WSTĘPNE**

Podstawowa wiedza dotycząca algorytmów i struktury danych, programowania obiektowego

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE**3.1 Cele przedmiotu**

C1	Poznanie metodyk testowania (manualnego i automatycznego)
C2	Zapoznanie z narzędziami do testowania jednostkowego, integracyjnego, systemowego, akceptacyjnego, API, GUI
C3	Poznanie narzędzi do automatyzacji testów i raportowania
C4	Poznanie podstaw testowania bezpieczeństwa i wydajności
C5	Rozwijanie umiejętności dokumentowania procesu testowego

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych
EK_01	Zna cykl powstawania oprogramowania i rozumie kluczową rolę testowania w tym procesie. Zna wybrane metodyki automatyzacji testowania. Zna podstawy testowania bezpieczeństwa.	K_Wo2,
EK_02	Zna i umie zastosować podstawowe narzędzia pozwalające testować oprogramowanie.	K_Uo5,
EK_03	Potrafi dobrać odpowiednie narzędzia do przeprowadzenia odpowiedniego typu testów oprogramowania.	K_Uo9,
EK_04	Umie dokumentować proces testowy i raportować wyniki.	K_U10,
EK_05	Potrafi pracować zespołowo, korzystając z narzędzi do integracji oprogramowania i zarządzania testami	K_U12

3.3 Treści programowe

A. Problematyka wykładu

Treści merytoryczne:
Klasyfikacja testów: jednostkowe, integracyjne, systemowe, akceptacyjne, regresyjne, wydajnościowe, bezpieczeństwa
Metodyki: TDD, BDD, ATDD
Narzędzia testowe: JUnit, TestNG, Selenium, REST Assured, Postman, Cucumber, Behave
Automatyzacja testów: frameworki, raportowanie (Allure, ExtentReports)
CI/CD: Jenkins, Docker, GIT, GitHub Actions, testy w pipeline
Testowanie API i mikroserwisów
Testowanie bezpieczeństwa (OWASP ZAP, podstawy pentestów)
Zarządzanie testami: TestRail, Zephyr, Jira
Dokumentacja testów i raportowanie
Refaktoryzacja, code review, jakość kodu

B. Problematyka ćwiczeń laboratoryjnych

Treści merytoryczne:
Tworzenie testów jednostkowych (JUnit, TestNG)
Testowanie API (REST Assured, Postman)
Automatyzacja testów GUI (Selenium, WebDriver)
Implementacja BDD (Cucumber, Behave)
Integracja testów z CI/CD (Jenkins, Docker)
Raportowanie wyników testów (Allure, ExtentReports)
Testowanie bezpieczeństwa – podstawy
Praca zespołowa z narzędziami do zarządzania testami

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną

Laboratorium: praca w grupach (rozwiązywanie zadań, projekt zespołowy, dyskusja wyników, sprawozdanie).

4. METODY I KRYTERIA OCENY

4.1 Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych (w, ćw, ...)
EK_01	Kolokwium	W, Lab
EK_02	Kolokwium, projekt	Lab
EK_03	Kolokwium, projekt	Lab
EK_04	Projekt, sprawozdanie	Lab
EK_05	Projekt, obserwacja w trakcie zajęć	Lab

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Kolokwium teoretyczne

Projekt zespołowy (np. wdrożenie testów automatycznych w pipeline)

Sprawozdania z laboratoriów

Prezentacja wyników testów i raportów

Ocena aktywności i pracy zespołowej

Efekt	Ocena	Kryteria
EK_01	dst	Umie wytłumaczyć rolę testowania w cyklu wytwórczym oprogramowania. Potrafi wskazać różnicę pomiędzy testami czarno- i białoskrzynkowymi. Rozumie potrzebę wprowadzenia testów bezpieczeństwa
	db	Zna i wymienia techniki automatyzacji testowania. Objasnia rolę testów black i white box.
	bdb	Potrafi wskazać różne kategorie testów i je porównać. Omawia wybraną technikę automatyzacji testowania.
EK_02	dst	Umie napisać test jednostkowy, wie gdzie jego tworzenie ma sens
	db	Umie napisać testy jednostkowe testujące wyjątki, czas obliczeń oraz wykorzystujące wspólne funkcje przygotowujące dane
	bdb	Umie napisać sparametryzowany test jednostkowy
EK_03	dst	Umie wymienić trzy narzędzia do wykonania testów i je scharakteryzować. Rozumie idee behavior-driven development
	db	Przyporządkowuje narzędzia do odpowiednich rodzajów testów. Podaje przykłady testów wykorzystujących język naturalny.
	bdb	Umie uzasadnić potrzebę powstania testów wykorzystujących język naturalny
EK_04, EK_05		Dokonuje wyboru i implementacji narzędzia do realizacji projektu. Opracowuje sprawozdanie z przebiegu prac projektowych..
		Umie przyporządkować odpowiednie narzędzia odpowiednim rodzajom testów - narzędzia te znajdują zastosowanie w realizacji projektu
		Umie zaplanować wielopoziomą procedurę testowania i stosuje ją w realizowanym projekcie

Zasady uzyskania oceny końcowej:

Zaliczenie laboratorium następuje na podstawie zaliczenia wszystkich efektów weryfikowanych przez planowane metody weryfikacji. Przy czym zakłada się, że każda metoda weryfikacji dostarcza osobne oceny dla każdego z weryfikowanych przez nią efektów uczenia się.

Zaliczenie wykładu może nastąpić tylko po zaliczeniu na ocenę pozytywną laboratorium
Kryteria oceny:

Student otrzymuje ocenę niedostateczny, gdy metody weryfikacji wykażą, iż co najmniej jeden z efektów nie został osiągnięty (średnia ocena dla tego efektu jest niższa niż 3.0);

Student otrzymuje ocenę dostateczny, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 3.0, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 3.75;

Student otrzymuje ocenę dobry, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 3.75, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 4.75;

Student otrzymuje ocenę bardzo dobry, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 4.75.

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	20
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	40
SUMA GODZIN	60
SUMARYCZNA LICZBA PUNKTÓW ECTS	2

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

Literatura podstawowa:

- [1] Ciągłe dostarczanie oprogramowania: automatyzacja kompilacji, testowania i wdrażania / Jez Humble, David Farley ; [tł.: Dorota Konowrocka-Sawa]. - Gliwice: Helion, cop. 2015.
- [2] Testowanie i jakość oprogramowania : modele, techniki, narzędzia / Adam Roman. - Warszawa : Wydawnictwo Naukowe PWN SA, cop. 2015.
- [3] Testowanie oprogramowania : podręcznik dla początkujących / Rafał Pawlak. - Gliwice : Wydawnictwo Helion, cop. 2014
- [4] ISTQB Foundation Level Syllabus (dostępny online)

Literatura uzupełniająca:

- [1] Certyfikowany tester ISTQB. Poziom podstawowy / Adam Roman, Lucjan Stapp - Helion 2020
- [2] Testy jednostkowe. Świat niezawodnych aplikacji/ Roy Oshero - Helion 2014
- [3] Dokumentacja narzędzi: JUnit, TestNG, Selenium, REST Assured, Cucumber, Jenkins, Docker

Akceptacja Kierownika Jednostki lub osoby upoważnionej