



Karta przedmiotu

Nazwa i kod przedmiotu	Oprogramowanie aplikacyjne - projekt zespołowy, PG_00064063						
Kierunek studiów	Fizyka Techniczna						
Data rozpoczęcia studiów	październik 2026 r.	Rok akademicki realizacji przedmiotu			2029/2030		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	4	Język wykładowy			polski		
Semestr studiów	7	Liczba punktów ECTS			4.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Fizyki Technicznej i Matematyki Stosowanej -> Instytut Fizyki i Informatyki Stosowanej -> Zakład Fizyki Teoretycznej i Informatyki Kwantowej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr hab. inż. Marta Łabuda				
	Prowadzący zajęcia z przedmiotu						
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	0.0	45.0	0.0	60
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	60		5.0		35.0	100
Cel przedmiotu	Cel przedmiotu to zapoznanie studentów z rolą oprogramowania aplikacyjnego w systemach informatycznych oraz umiejętnością jego tworzenia i zarządzania. Kurs obejmuje zagadnienia związane z zarządzaniem projektami informatycznymi, w tym planowaniem zasobów, harmonogramów i budżetów, a także z zarządzaniem konfiguracją oprogramowania i wersjonowaniem. Studenci nauczą się metod implementacji i integracji oprogramowania w systemach IT oraz zdobywają wiedzę o procesach testowania, w tym testach jednostkowych, integracyjnych i systemowych. Kurs kładzie również nacisk na automatyzację testów, zastosowanie TDD i BDD oraz na zapewnianie jakości oprogramowania w różnych fazach jego życia. Celem projektu jest przygotowanie studentów do skutecznego projektowania, implementowania i zarządzania cyklem życia oprogramowania aplikacyjnego w różnych środowiskach technologicznych.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K6_U03] posiada umiejętność programowania w wybranym języku oraz stosowania podstawowych pakietów oprogramowania	Student potrafi tworzyć, kompilować i uruchamiać programy w wybranym języku programowania, stosując podstawowe konstrukcje językowe (instrukcje sterujące, struktury danych, funkcje/metody). Student potrafi stosować wybrane pakiety oprogramowania oraz narzędzia programistyczne (np. środowisko IDE, system kontroli wersji, repozytorium, biblioteki standardowe, frameworki frontendowe, programy obliczeniowe) w procesie tworzenia oprogramowania.	[SU1] Ocena realizacji zadania
	[K6_K03] jest gotów do odpowiedzialnego pełnienia ról zawodowych, przestrzegania zasad etyki oraz dbałości o bezpieczeństwo pracy.	Student jest przygotowany do odpowiedzialnego wykonywania obowiązków zawodowych z poszanowaniem zasad etyki i bezpieczeństwa pracy.	[SK5] Ocena umiejętności rozwiązywania problemów występujących w praktyce
	[K6_W05] posiada wiedzę w zakresie metodyki i technik programowania oraz wykorzystywania wybranych narzędzi informatycznych w fizyce i technice.	Student zna metodyki wytwarzania oprogramowania (np. kaskadową, iteracyjną, zwinną) oraz podstawowe techniki programowania. Student zna wybrane narzędzia informatyczne wspomagające proces tworzenia oprogramowania (środowiska programistyczne, systemy kontroli wersji, narzędzia do modelowania (np CASE do diagramów UML) i testowania, potrafi z nich korzystać oraz rozumie ich zastosowanie w problemach fizyki i techniki.	[SW1] Ocena wiedzy faktograficznej
	[K6_U07] potrafi rozwiązywać problemy w ramach zespołu, także interdyscyplinarnego, odpowiednio planując jego pracę.	Student potrafi efektywnie współpracować w zespole, także interdyscyplinarnym. Student potrafi planować i organizować pracę zespołu w celu rozwiązania problemów. Student potrafi komunikować się i koordynować zadania w zespole projektowym.	[SU2] Ocena umiejętności analizy informacji [SU4] Ocena umiejętności korzystania z metod i narzędzi

Treści przedmiotu	<p>Treści przedmiotu - wykład</p> <ol style="list-style-type: none"> 1. Definicja, klasyfikacja i rola oprogramowania aplikacyjnego w systemach informatycznych. Zastosowanie w nauce i technice. 2. Zarządzanie projektem informatycznym: zasoby, personel, harmonogram, budżet. 3. Zarządzanie konfiguracją oprogramowania. Przegląd popularnych aplikacji i platform. Środowiska narzędziowe oraz technologie i frameworki. Wykorzystanie narzędzi CASE. 4. Implementacja i integracja oprogramowania. 5. Kodowanie, wersjonowanie i bezpieczeństwo kodu. 6. Testowanie i walidacja oprogramowania. Rodzaje testów: testy jednostkowe, integracyjne, systemowe; walidacja i testy akceptacyjne. 7. Typowe fazy i metody testowania. Testy statyczne, funkcjonalne i strukturalne. Czynniki sukcesu i rezultaty testowania. Dokumentowanie testów. Przeglądy oprogramowania. 8. Automatyzacja procesu testowania. Narzędzia do debugowania. Programowanie sterowane testami i zachowanie systemu (TDD i BDD). 9. Jakość oprogramowania. Zapewnianie jakości oprogramowania. Zarządzanie jakością. Koszty zapewniania jakości. Dobre praktyki. 10. Instalacja i wdrożenie oprogramowania. Pielęgnacja oprogramowania. Analiza potrzeby wprowadzania modyfikacji. Zarządzanie zmianą.. Koszty konserwacji oprogramowania. Kluczowe czynniki sukcesu fazy utrzymania. <hr/> <p>Treści przedmiotu - projekt</p> <p>Projekt: Konstrukcja aplikacji w oparciu o przyjętą metodykę, wzorce (analityczne, projektowe) frameworki i komponenty + dokumentacja projektu. Praca z repozytorium oraz oprogramowaniem do zarządzania projektem:</p> <ol style="list-style-type: none"> 1. Wprowadzenie. Założenia prac projektowych, ich prowadzenia i dokumentowania. Dyskusja nad sformułowaniami projektów i grup projektowych. Opis projektu, założenia projektowe. Prezentacja przykładowych rozwiązań projektowych. 2. Prezentacja pomysłów (pitch project). Identyfikacja problemu. Kontekst i udziałowcy problemu; klient projektu. Wzbożony wizerunek, definicja i wizja systemu, zakres systemu. 3. Studium wykonalności projektu; cel, produkt, zakres przedsięwzięcia; identyfikacja zagrożeń, dobór strategii realizacji projektu. Harmonogram prac projektowych. 4. Dobór strategii prowadzenia projektów. Dalsze kroki prac implementacyjnych zależnie od przyjętej metodyki realizacji. (kaskada, agile) 5. Prace implementacyjne: projekt architektury systemu, modele logiczne i statyczne, użyte algorytmy, projekt GUI, projekt bazy danych. 6. Prace implementacyjne: tworzenie oprogramowania 7. Prace implementacyjne: inspekcja kodów (code review). Raport z inspekcji. 8. Prace implementacyjne: Integracja oprogramowania. 9. Testowanie i walidacja wykonywanego oprogramowania; Raport z testów.
-------------------	---

	<p>Konferencja: Opisać techniczny projekt. Prezentacja dokumentacji i przygotowanie raportów. Umiejętność pracy zespołowej, inżynierskiej, inżynierskiej. Zaliczenie przedmiotu.</p> <p>Każdy tydzień prac kończy Raport z przebiegu prac projektowych.</p>														
Wymagania wstępne i dodatkowe	Umiejętność programowania obiektowego; Znajomość zagadnień inżynierii oprogramowania														
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	<table border="1"> <thead> <tr> <th data-bbox="448 358 794 392">Sposób oceniania (składowe)</th> <th data-bbox="794 358 1141 392">Próg zaliczeniowy</th> <th data-bbox="1141 358 1489 392">Składowa oceny końcowej</th> </tr> </thead> <tbody> <tr> <td data-bbox="448 392 794 425">Projekt grupowy</td> <td data-bbox="794 392 1141 425">50.0%</td> <td data-bbox="1141 392 1489 425">80.0%</td> </tr> <tr> <td data-bbox="448 425 794 459">Raporty z prac zespołu</td> <td data-bbox="794 425 1141 459">50.0%</td> <td data-bbox="1141 425 1489 459">10.0%</td> </tr> <tr> <td data-bbox="448 459 794 495">Prezentacja projektu</td> <td data-bbox="794 459 1141 495">50.0%</td> <td data-bbox="1141 459 1489 495">10.0%</td> </tr> </tbody> </table>			Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej	Projekt grupowy	50.0%	80.0%	Raporty z prac zespołu	50.0%	10.0%	Prezentacja projektu	50.0%	10.0%
Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej													
Projekt grupowy	50.0%	80.0%													
Raporty z prac zespołu	50.0%	10.0%													
Prezentacja projektu	50.0%	10.0%													
Zalecana lista lektur	<table border="1"> <tbody> <tr> <td data-bbox="448 499 794 1182"> Podstawowa lista lektur </td> <td colspan="2" data-bbox="794 499 1489 1182"> <ol style="list-style-type: none"> 1. Robert C. Martin, <i>Czysty kod</i>, Helion, 2020 (dodruk/aktualne wydanie). 2. Marcin Kawalec, <i>Scrum i nie tylko. Zwinne metody w praktyce</i>, Helion, 2019. 3. Paweł Jakubczyk, <i>Testowanie oprogramowania. Podstawy</i>, Helion, 2019. 4. Robert. Patton: <i>Testowanie oprogramowania</i>, Mikom, Warszawa, 2002l. 5. Mariot Tsitoara --Git i GitHub. Kontrola wersji, zarządzanie projektami i zasady pracy zespołowej (oryg: Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer) 6. Michał Bartyzel <i>DevOps. Inżynieria niezawodności systemów IT</i> Helion, 2020 </td> </tr> <tr> <td data-bbox="448 1187 794 1608"> Uzupełniająca lista lektur </td> <td colspan="2" data-bbox="794 1187 1489 1608"> <ol style="list-style-type: none"> 1. Andrzej Jaskiewicz, <i>Zarządzanie projektami informatycznymi</i>, PWN, 2019. 2. Leszek Bukowski, <i>Analiza i projektowanie systemów informatycznych</i>, PWN, 2018. 3. Adam Roman, Michał Sadowski, <i>Testowanie i jakość oprogramowania</i>, PWN, 2021. 4. Odnośniki do stron domowych wybranych przez studentów technologii, w jakich realizowany jest projekt grupowy, </td> </tr> <tr> <td data-bbox="448 1612 794 1630"> Adresy eZasobów </td> <td colspan="2" data-bbox="794 1612 1489 1630"></td> </tr> </tbody> </table>			Podstawowa lista lektur	<ol style="list-style-type: none"> 1. Robert C. Martin, <i>Czysty kod</i>, Helion, 2020 (dodruk/aktualne wydanie). 2. Marcin Kawalec, <i>Scrum i nie tylko. Zwinne metody w praktyce</i>, Helion, 2019. 3. Paweł Jakubczyk, <i>Testowanie oprogramowania. Podstawy</i>, Helion, 2019. 4. Robert. Patton: <i>Testowanie oprogramowania</i>, Mikom, Warszawa, 2002l. 5. Mariot Tsitoara --Git i GitHub. Kontrola wersji, zarządzanie projektami i zasady pracy zespołowej (oryg: Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer) 6. Michał Bartyzel <i>DevOps. Inżynieria niezawodności systemów IT</i> Helion, 2020 		Uzupełniająca lista lektur	<ol style="list-style-type: none"> 1. Andrzej Jaskiewicz, <i>Zarządzanie projektami informatycznymi</i>, PWN, 2019. 2. Leszek Bukowski, <i>Analiza i projektowanie systemów informatycznych</i>, PWN, 2018. 3. Adam Roman, Michał Sadowski, <i>Testowanie i jakość oprogramowania</i>, PWN, 2021. 4. Odnośniki do stron domowych wybranych przez studentów technologii, w jakich realizowany jest projekt grupowy, 		Adresy eZasobów					
Podstawowa lista lektur	<ol style="list-style-type: none"> 1. Robert C. Martin, <i>Czysty kod</i>, Helion, 2020 (dodruk/aktualne wydanie). 2. Marcin Kawalec, <i>Scrum i nie tylko. Zwinne metody w praktyce</i>, Helion, 2019. 3. Paweł Jakubczyk, <i>Testowanie oprogramowania. Podstawy</i>, Helion, 2019. 4. Robert. Patton: <i>Testowanie oprogramowania</i>, Mikom, Warszawa, 2002l. 5. Mariot Tsitoara --Git i GitHub. Kontrola wersji, zarządzanie projektami i zasady pracy zespołowej (oryg: Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer) 6. Michał Bartyzel <i>DevOps. Inżynieria niezawodności systemów IT</i> Helion, 2020 														
Uzupełniająca lista lektur	<ol style="list-style-type: none"> 1. Andrzej Jaskiewicz, <i>Zarządzanie projektami informatycznymi</i>, PWN, 2019. 2. Leszek Bukowski, <i>Analiza i projektowanie systemów informatycznych</i>, PWN, 2018. 3. Adam Roman, Michał Sadowski, <i>Testowanie i jakość oprogramowania</i>, PWN, 2021. 4. Odnośniki do stron domowych wybranych przez studentów technologii, w jakich realizowany jest projekt grupowy, 														
Adresy eZasobów															

<p>Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania</p>	<p>Zagadnienia:</p> <ol style="list-style-type: none"> 1. Rola i znaczenie oprogramowania aplikacyjnego we współczesnych systemach informatycznych. 2. Planowanie i zarządzanie projektem informatycznym z uwzględnieniem zasobów, harmonogramu i budżetu. 3. Zarządzanie konfiguracją i wersjonowaniem oprogramowania z wykorzystaniem nowoczesnych narzędzi i platform. 4. Proces implementacji i integracji oprogramowania w złożonych systemach informatycznych. 5. Metody i techniki testowania oprogramowania oraz ich wpływ na jakość produktu końcowego. 6. Automatyzacja testów oraz zastosowanie podejść TDD i BDD w procesie wytwarzania oprogramowania. 7. Zapewnianie jakości i utrzymanie oprogramowania dobre praktyki, koszty i zarządzanie zmianą. <p>Zadania:</p> <p>Przygotowanie harmonogramu prac</p> <p>Raporty prac zespołu projektowego</p> <p>Implementacja zadania programistycznego</p> <p>Inspekcja kodu</p> <p>Testowanie oprogramowania</p> <p>Prezentacja projektu</p>
<p>Zajęcia praktyczne w ramach przedmiotu</p>	<p>Nie dotyczy</p>

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.