POLITECHNIKA BYDGOSKA
Wydział Telekomunikacji, Informatyki i Elektrotechniki

Karta przedmiotu

Programowanie zwinne

1. Informacje podstawowe

Kierunek studiów informatyka stosowana	**Cykl kształcenia (nabór)** 2024/25
Specjalność -	**Kod przedmiotu** 05ISTS.DI1C.0240.24
Jednostka zarządzająca kierunkiem studiów Wydział Telekomunikacji, Informatyki i Elektrotechniki	**Języki wykładowe** polski
Poziom studiów drugiego stopnia (mgr inż.)	**Obligatoryjność** Obowiązkowy
Profil studiów Profil ogólnoakademicki	**Blok zajęciowy** Przedmioty kierunkowe
Forma studiów studia stacjonarne	

Wymagania wstępne	Brak wymagań wstępnych
Przedmioty wprowadzające	Brak przedmiotów wprowadzających
Koordynator	Damian Szczegielniak

Okres	**Forma i godziny zajęć**	**Liczba punktów ECTS**
Semestr 1	• Wykład: 30, Zaliczenie na ocenę • Ćwiczenia laboratoryjne: 30, Zaliczenie na ocenę	3

2. Efekty uczenia się dla przedmiotu

Kod	**Opis efektów uczenia się**	**Odniesienie do kierunkowych efektów uczenia się**	**Odniesienie do charakterystyk PRK**
Wiedza:			

Kod	Opis efektów uczenia się	Odniesienie do kierunkowych efektów uczenia się	Odniesienie do charakterystyk PRK
W1	ma poszerzoną i pogłębioną wiedzę w zakresie inżynierii oprogramowania, w tym harmonogramowania zadań i języków modelowania	IST_O2_K_W06	P7S_WG P7S_WG_inż
Umiejętności:			
U1	potrafi posłużyć się odpowiednimi środowiskami i narzędziami programistycznymi do projektowania, realizacji i testowania aplikacji w różnych językach programowania	IST_O2_K_U02	P7S_UW P7S_UW_inż
U2	potrafi pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zleconego zadania; potrafi opracować i zrealizować harmonogram prac zapewniający dotrzymanie terminów; potrafi ocenić ryzyka związane z komunikacją i pracą w środowisku wielokulturowym	IST_O2_K_U09	P7S_UK
Kompetencje społeczne:			
K1	ma świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżyniera informatyka i związaną z tym odpowiedzialność za podejmowane decyzje; potrafi odpowiednio określić priorytety służące realizacji określonego zadania	IST_O2_K_K02	P7S_KO
K2	ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania	IST_O2_K_K04	P7S_KO

3. Treści programowe

Lp.	Treści programowe	Formy zajęć	Efekty uczenia się dla przedmiotu
1.	Przedstawienie procesu efektywnego wytwarzania oprogramowania wykorzystującego zbiór wybranych metodyk określanych jako zwinne (ang. Agile Methodologies) z naciskiem na zapewnienie wysokiej jakości oprogramowania i obniżenie kosztów jego wytwarzania, a także minimalizację ryzyka niepowodzenia przedsięwzięć informatycznych. W trakcie wykładu omawiane są również stosowane przez zespoły sprawdzone techniki i narzędzia programistyczne pozwalające usprawnić proces wytwarzania oprogramowania o wysokiej jakości. Zakres tematyczny wykładów: 1. manifest programowania zwinnego i jego założenia, 2. dobre praktyki w tworzeniu oprogramowania (zasady projektowania obiektowego, wyszukiwanie antywzorców, korzystanie z diagramów UML, stosowanie wzorców projektowych), 3. refaktoryzacja kodu, 4. wytwarzanie oprogramowania wykorzystujące elementy metodyk zwinnych m.in. XP (ang. Extreme Programming), Scrum oraz Scrumban, 5. narzędzia usprawniające proces budowania aplikacji, 6. możliwości narzędzi kontroli wersji i sposoby ich praktycznego zastosowania w projektach wymagających pracy zespołowej, 7. weryfikacja poprawności tworzonego oprogramowania za pomocą testów automatycznych, programowanie sterowane testami (ang. Test Driven Development), cechy testów wysokiej jakości, 8. metody doskonalenia procesów budowania oprogramowania stosowane przez zespoły zwinne - podejścia Lean oraz Kanban.	Wykład	W1

Lp.	Treści programowe	Formy zajęć	Efekty uczenia się dla przedmiotu
2.	Ćwiczenia laboratoryjne polegają na rozwiązywaniu przygotowanych przez prowadzącego zadań programistycznych zgodnie z zasadami i standardami postępowania powszechnie stosowanymi w metodykach zwinnych. Zbiór zadań jest tak dobrany, aby pozwalał na praktyczne wykorzystanie niezawodnych technik i narzędzi programistycznych omawianych w trakcie wykładu. Realizowane w ramach zajęć laboratoryjnych zadania służą przyrostowemu, iteracyjnemu budowaniu aplikacji webowej, a także pozwalają studentom wyciągać wnioski i spostrzeżenia na temat celowości i korzyści zastosowania zwinnego podejścia w projektowaniu i tworzeniu oprogramowania. Ćwiczenia laboratoryjne obejmują m.in.: 1. Określanie wymagań funkcjonalnych i niefunkcjonalnych za pomocą tzw. historyjek użytkownika (tworzenie tzw. rejestru produktu). 2. Uszczegóławianie wybranych wymagań, a także tworzenie na ich podstawie zadań cząstkowych i szacowanie czasu ich realizacji (tworzenie tzw. rejestru sprintu). 3. Podział i realizacja zadań składających się na zaplanowane do utworzenia w najbliższej iteracji funkcji oprogramowania. 4. Rezultatem końcowym iteracyjnego i przyrostowego budowania oprogramowania w trakcie zajęć laboratoryjnych ma być aplikacja webowa, do której utworzenie niezbędne będzie m.in.: − Korzystanie ze zdalnego repozytorium kodu źródłowego. − Wykorzystanie narzędzi Gradle lub Maven do budowania aplikacji. − Konfiguracja i podłączenie mechanizmu rejestracji np. Logback z SLF4J. − Realizacja warstwy dostępu do danych za pomocą frameworków Hibernate i Spring. W zadaniu poruszane są zagadnienia związane z mapowaniem obiektowo-relacyjnym, transakcjami bazodanowymi, adnotacjami JPA, językiem JPQL, a także stronicowaniem wyników. − Implementacja usługi typu REST (z wykorzystaniem frameworku Spring, bazy danych PostgreSQL oraz serwera Apache Tomcat) i front-endu z niej korzystającego, opartego np. na springowych rozwiązaniach - Thymeleaf oraz RestTemplate lub WebClient. Opcjonalnie do stworzenia front-endu aplikacji można użyć frameworku Angular lub biblioteki React. − Implementacja mechanizmu walidacji danych, a także przechwytywanie i obsługa błędów. − Zabezpieczenie danych i aplikacji przed niepożądanym dostępem. − Utworzenie odrębnych uprawnień dla użytkowników aplikacji. − Implementacja testów automatycznych (m.in. z wykorzystaniem JUnit5, Mockito i MockMVC), programowanie sterowane testami.	Ćwiczenia laboratoryjne	U1, U2, K1, K2

4. Metody prowadzenia zajęć, weryfikacji efektów uczenia się i warunki zaliczenia

Forma zajęć		
Wykład	**Metody prowadzenia zajęć:**	
	Wykład, Ćwiczenia laboratoryjne, Projekt	
	Metody (sposoby) weryfikacji:	**Udział:**
	Sprawdzian	100%
	Warunki zaliczenia przedmiotu:	
	Warunkiem zaliczenia jest uzyskanie minimum 51% zakładanych efektów uczenia się zgodnie z §22 Regulaminu Studiów PBŚ (ocena z kolokwium)	
Ćwiczenia laboratoryjne	**Metody prowadzenia zajęć:**	
	Ćwiczenia laboratoryjne	
	Metody (sposoby) weryfikacji:	**Udział:**
	Sprawozdanie	100%
	Warunki zaliczenia przedmiotu:	
	Warunkiem zaliczenia jest uzyskanie minimum 51% zakładanych efektów uczenia się zgodnie z §22 Regulaminu Studiów PBŚ (średnia arytmetyczna ze sprawozdań)	

Efekt uczenia się dla przedmiotu	**Metody (sposoby) weryfikacji**	
	Sprawdzian	Sprawozdanie
W1	x	
U1		x
U2		x
K1		x
K2		x

5. Literatura

Literatura podstawowa

1. Stellman A., Greene J. 2015. Agile. Przewodnik po zwinnych metodykach programowania, Helion
2. Rubin K.S. 2013. Scrum. Praktyczny przewodnik po najpopularniejszej metodyce Agile. Helion
3. Martin R.C. 2015. Zwinne wytwarzanie oprogramowania. Najlepsze zasady, wzorce i praktyki, Helion
4. Martin R.C. 2018. Czysta architektura. Struktura i design oprogramowania. Przewodnik dla profesjonalistów. Helion
5. Walls C. 2019. Spring w akcji. Wydanie V, Helion

Literatura uzupełniająca

1. Pilone D., Miles R. 2008. Head First Software Development. Edycja polska. Helion
2. Freeman E., Freeman E., Bates B., Sierra K. 2005. Head First Design Patterns. Edycja polska, Helion
3. John Ferguson Smart J.F. 2009. Java. Praktyczne narzędzia, Helion
4. Schildt H. 2023. Java. Kompendium programisty. Wydanie XII, Helion

6. Nakład pracy studenta - bilans godzin i punktów ECTS

Aktywność studenta		Obciążenie studenta Liczba godzin
Zajęcia prowadzone z bezpośrednim udziałem nauczyciela akademickiego lub innych osób prowadzących zajęcia	Wykład	30
	Ćwiczenia laboratoryjne	30
Praca własna studenta	Przygotowanie do zajęć	10
	Studiowanie literatury	10
	Konsultacje	2
	Inne (przygotowanie do egzaminu)	8
Łączny nakład pracy studenta		90
Liczba punktów ECTS		3

* Godzina (dydaktyczna) oznacza 45 minut