



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	
Nazwa przedmiotu	Systemy wspomagające tworzenie oprogramowania
Nazwa przedmiotu w języku angielskim	Systems supporting software development
Obowiązuje od roku akademickiego	2020/21

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	Teleinformatyka
Poziom kształcenia	I stopień
Profil studiów	Praktyczny
Forma i tryb prowadzenia studiów	Studia stacjonarne
Zakres	
Jednostka prowadząca przedmiot	Katedra Systemów Informatycznych
Koordinator przedmiotu	Dr inż. Karol Wieczorek Mgr inż. Daniel Kaczmarek
Zatwierdził	Dziekan Wydziału Elektrotechniki Automatyki i Informatyki Dr hab. inż. Antoni Różowicz, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów*	Przedmiot kierunkowy
Status przedmiotu*	Obieralny
Język prowadzenia zajęć	Polski
Usytuowanie modułu w planie studiów - semestr	Semestr VII
Wymagania wstępne	
Egzamin (TAK/NIE)	Nie
Liczba punktów ECTS	3

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	Inne
Liczba godzin w semestrze	15	0	0	30	0



EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Wiedza z języków programowania, baz danych, inżynierii oprogramowania.	TI1_W06
	W02	Zna zagadnienia związane z tworzeniem aplikacji internetowych i mobilnych.	TI1_W09
	W03	Zna ogólne zasady i metody zarządzania projektami.	TI1_W17
Umiejętności	U01	Umiejętność tworzenia programów w językach programowania z wykorzystaniem różnych typów danych, instrukcji sterujących i funkcji oraz zdolność implementowania wybranych algorytmów.	TI1_U01 TI1_U02
	U02	Umiejętność planowania i zarządzania przykładowym projektem zgodnie z wybraną metodyką.	TI1_U01 TI1_U10 TI1_U19
	U03	Umiejętność opracowania dokumentacji i sprawozdania dotyczących rozwiązane go problemu programistycznego.	TI1_U01 TI1_U02 TI1_U20
Kompetencje społeczne	K01	Kompetencja do pracy w zespole podczas rozwiązywania problemów programistycznych i umiejętność ich podziału na mniejsze zadania.	TI1_K01 TI1_K02 TI1_K03

TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
Wykład	1. Wprowadzenie do metodologii wytwarzania oprogramowania
	2. Metody tworzenia kodu programów - TDD, BDD, Agile
	3. Narzędzia do zarządzania projektem
	4. System CVS
	5. Narzędzia do tworzenia dokumentacji technicznej projektu informatycznego
	6-8. Continuous Integration, Delivery, Development
Projekt	Studenci wykonują zadanie w zespole cztero-/pięcioosobowym. Każdy z członków zespołu ma przyporządkowaną określoną rolę (analityk, architekt, developer, tester, serwisant). Przedmiotem zadania jest wykonanie systemu zarządzania firmą lub organizacją (np. hotel, spółdzielnia mieszkaniowa). Wykonanie zadania wymaga przeprowadzenia typowych czynności realizowanych w ramach projektu tworzenia systemu informatycznego: specyfikacji wymagań, projektowania architektury oprogramowania, implementacji, opracowania testów oraz dostosowania systemu do potrzeb klienta.



METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów uczenia się					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			X	X	X	
W02			X	X	X	
W03			X	X	X	
U01			X	X	X	
U02			X	X	X	
U03			X	X	X	
K01			X	X	X	

FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia*	Warunki zaliczenia
Wykład	zaliczenie z oceną	Uzyskanie min. 50% punktów z testu końcowego
Projekt	zaliczenie z oceną	Wykonanie projektu ocenionego na ocenę pozytywną

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
1.	Udział w zajęciach zgodnie z planem studiów	W	C	L	P	S	h
		15	0	0	30	0	
3.	Inne (konsultacje, egzamin)*	2	0	0	2	0	h
4.	Razem przy bezpośrednim udziale nauczyciela akademickiego	49					h
5.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	1,96					ECTS
6.	Liczba godzin samodzielnej pracy studenta	26					h
7.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	1,04					ECTS
8.	Nakład pracy związany z zajęciami o charakterze praktycznym	30					h
9.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	1,20					ECTS
10.	Sumaryczne godzinowe obciążenie pracą studenta	75					h
11.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	3					



LITERATURA

1. Bell P., Beer B.: GitHub. Przyjazny przewodnik. Helion, Warszawa 2015.
2. Beck K.: TDD. Sztuka tworzenia dobrego kodu. Helion, Warszawa 2014.
3. Martin Robert C.: Czysty kod. Podręcznik dobrego programisty. Helion, Warszawa 2010.
4. Martin Robert C., Martin M.: Agile. Programowanie zwinne: zasady, wzroce i praktyki zwinnego wytwarzania oprogramowania w C#. Helion, Warszawa 2008.
5. Cohn M.: Agile. Metodyki zwinne w planowaniu projektów. Helion, Warszawa 2018.
6. Leszko R.: Continuous Delivery with Docker and Jenkins. Packt Publishing, 2017.
7. Woźniak D.: TDD. Techniki programowania sterowanego testami. Helion, Warszawa 2018.
8. Zmitrowicz K.: Jakość projektów informatycznych. Rozwój i testowanie oprogramowania. Helion, 2015.
9. Gajda W.: Git. Rozproszony system kontroli wersji, Helion, Warszawa 2013.
10. Ferguson S. J.: Jenkins: The Definitive Guide: Continuous Integration for the Masses, Oreilly Media, 2011.