

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Studio rozwoju oprogramowania 2</b>		Kod <b>1010512321010513908</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>1 / 2</b>
Ścieżka obieralności/specjalność <b>Software Engineering (Inżynieria)</b>	Przedmiot oferowany w języku: <b>angielski</b>	Kurs (obligatoryjny/obieralny) <b>obligatoryjny</b>
Stopień studiów: <b>II stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>stacjonarna</b>	
Godziny Wykłady: - Ćwiczenia: - Laboratoria: - Projekty/seminaria: <b>60</b>		Liczba punktów <b>5</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) <b>kierunkowy</b>		(ogólnouczelniany, z innego kierunku) <b>z danego kierunku</b>
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b>  <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>5 100%</b>  <b>5 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
mgr inż. Sylwia Kopczyńska email: Sylwia.Kopczynska@cs.put.poznan.pl tel. 61 665 2944 Wydział Informatyki ul. Piotrowo 3, 60-965 Poznań		dr inż. Mirosław Ochodek email: Miroslaw.Ochodek@cs.put.poznan.pl tel. 61 665 2944 Wydział Informatyki ul. Piotrowo 3, 60-965 Poznań
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Student powinien posiadać ogólną wiedzę z zakresu inżynierii oprogramowania oraz szczegółowa wiedza dotycząca zarządzania projektem informatycznym według metodyki Scrum.
2	<b>Umiejętności:</b>	Student powinien posiadać umiejętności związane z zarządzaniem projektem według metodyki Scrum (w szczególności zarządzanie Rejestrem Produktu, planowanie sprintów i wydań, organizowanie spotkań projektowych zdefiniowanych w metodyce Scrum)
3	<b>Kompetencje społeczne</b>	Student powinien rozumieć konieczność poszerzania swoich kompetencji / mieć gotowość do podjęcia współpracy w ramach zespołu. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
<b>Cel przedmiotu:</b>		
<p>- Przekazanie studentom oraz zilustrowanie na rzeczywistym przypadku podstawowej wiedzy z zakresu zarządzania projektami informatycznymi oraz inżynierii wymagań niezbędnej do pełnienia roli kierowniczej lub analitycznej w projekcie informatycznym.</p> <p>- Rozwijanie u studentów umiejętności związanych z tworzeniem systemów informatycznych (a w szczególności umiejętności z zakresu zarządzania projektami informatycznymi, inżynierii wymagań oraz architektury systemów) poprzez udział w realnym projekcie informatycznym rozwiązującym rzeczywisty problem na zlecenie konkretnego odbiorcy. W ramach przedmiotu główny nacisk położony jest na kształtowanie umiejętności związanych z zarządzaniem zespołem programistycznym, inżynierią wymagań (opieka nad Rejestrem Produktu) oraz projektowania i dokumentowania architektury systemu informatycznego.</p> <p>- Kształtowanie u studentów umiejętności pracy zespołowej.</p> <p>- Moduł stanowi rozszerzenie modułu Studio Rozwoju Oprogramowania 1.</p>		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
<p>1. ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną na temat zwinnych podejść do zarządzania projektami (np. Manifest Zwinności, najpopularniejsze zwinne metodyki wytwarzania oprogramowania) - [K2st_W2]</p> <p>2. ma zaawansowaną wiedzę szczegółową dotyczącą metodyki Scrum - [K2st_W3]</p> <p>3. ma zaawansowaną i szczegółową wiedzę o procesie rozwoju oprogramowania proponowanego w metodyce Scrum - [K2st_W5]</p>		
<b>Umiejętności:</b>		

<ol style="list-style-type: none"><li>1. potrafi pozyskiwać wiedzę potrzebną do realizacji projektu informatycznego (m.in. wiedzę dotyczącą budowania zespołu) - [K2st_U1]</li><li>2. potrafi używać narzędzia do zarządzania zadaniami w projekcie informatycznym (np. Jira) organizując pracę zespołu deweloperskiego - [K2st_U2]</li><li>3. potrafi pozyskać i zintegrować wiedzę techniczną oraz dziedzinową, aby prowadzić rozmowy z użytkownikami i/lub klientem, aby wyznaczyć kierunki rozwoju produktu informatycznego - [K2st_U5]</li><li>4. potrafi ocenić przydatność rozwiązań technologicznych dla rozwijanego produktu informatycznego - [K2st_U6]</li><li>5. potrafi dobrać oraz użyć metodę szacowania pracochłonności do planowania zakresów sprintów (np. Poker Planistyczny) oraz wydań - [K2st_U7]</li><li>6. potrafi wykorzystać mechanizm retrospektywy do analizy problemów i usprawnienia procesu realizacji projektu informatycznego - [K2st_U8]</li><li>7. potrafi dokonać oceny jakości kodu produktu informatycznego i zaproponować usprawnienia - [K2st_U8]</li><li>8. potrafi zaprojektować system informatyczny spełniający postawione wymagania - [K2st_U11]</li><li>9. potrafi współdziałać w zespole projektowym, przyjmując w nim role zarządcze i analityczne - [K2st_U15]</li></ol>
<b>Kompetencje społeczne:</b>
<ol style="list-style-type: none"><li>1. jest świadomy(a) kierunku i tempa rozwoju metodyk zarządzania projektami - [K2st_K1]</li><li>2. rozumie znaczenie potrzeby rozwoju metod i technologii informatycznych oraz dostosowywania ich do potrzeb realizowanego projektu - [K2st_K2]</li></ol>

<b>Sposoby sprawdzenia efektów kształcenia</b>
<p>Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:</p> <p>Ocena formująca:</p> <ul style="list-style-type: none"><li>- na podstawie odpowiedzi udzielanych odnośnie realizacji zadań związanych z projektem;</li><li>- na podstawie oceny bieżącego postępu realizacji zadań projektowych.</li></ul> <p>Ocena podsumowująca:</p> <p>Wyznaczana na podstawie oceny trzech kryteriów (średniej ważona liczby punktów procentowych):</p> <ul style="list-style-type: none"><li>- aktywne uczestnictwo w zajęciach (średnia ocena =&gt; 0-100%, waga 0,1)</li><li>- wyniki testu wiedzy (m.in. na podstawie testów certyfikacyjnych scrum.org) (średnia ocena z testów =&gt; 0-100%, waga 0,3)</li><li>- ocena realizacji projektu (średnia ocena z ocen tygodniowych =&gt; 0-100%, waga 0,6)</li></ul> <p>Ocena wynikowa określona według następującej skali:</p> <ul style="list-style-type: none"><li>- (90%, 100%) ? 5,0</li><li>- (80%, 90%) ? 4,5</li><li>- (70%, 80%) ? 4,0</li><li>- (60%, 70%) ? 3,5</li><li>- (50%, 60%) ? 3,0</li><li>- (0%, 50%) ? 2,0</li></ul>
<b>Treści programowe</b>
<p>Moduł stanowi rozwinięcie modułu Studio Rozwoju Oprogramowania 1 i polega na dalszym rozwoju systemu informatycznego na zlecenie rzeczywistego odbiorcy. Realizacja projektu odbywa się według zaleceń metodyki zarządzania projektami Scrum. W trakcie realizacji projektu student pełni rolę kierownika projektu, Scrum Mastera, analityka (Product Owner lub Proxy-Product Owner) lub architekta, natomiast rolę programistów pełnią studenci studiów pierwszego stopnia.</p> <p>Realizacja zajęć obejmuje następujące etapy projektowe: dopracowanie architektury; konstrukcja oprogramowania ? typowo obejmująca dwa wydania oraz zamknięcie projektu.</p> <p>W ramach realizacji powyższych etapów poruszane są następujące zagadnienia:</p> <ul style="list-style-type: none"><li>- struktura zespołu projektowego i odpowiedzialności</li><li>- zarządzanie projektem oraz etapem projektu (planowanie wydania; delegowanie zadań w zespole, zarządzanie zmianą, tworzenie planu oraz podsumowania wydania)</li><li>- monitorowanie postępu projektu oraz podejmowanie decyzji na podstawie faktów,</li><li>- zapewnianie jakości oprogramowania (testy akceptacyjne dotyczące wymagań funkcjonalnych oraz pozafunkcjonalnych),</li><li>- architektura oprogramowania (aktualizacja architektury oprogramowania w odpowiedzi na zmiany wymagań, technologie tworzenia systemów informatycznych)</li><li>- warsztaty retrospekcji (ang. reflection workshops)</li><li>- wdrożenie oprogramowania (formalne odbiory systemów informatycznych, przygotowanie oprogramowania do przekazania)</li><li>zarządzanie ryzykiem (identyfikacja, analiza oraz przeciwdziałanie ryzyku)</li></ul>

<p>Więcej na temat stosowanej metody dydaktycznej przedstawiono w artykule:                  Kopczyńska, Sylwia, Jerzy Nawrocki, and Miroslaw Ochodek. Software development studio: bringing industrial environment to a classroom. Proceedings of the First International Workshop on Software Engineering Education Based on Real-World Experiences. IEEE Press, 2012.</p>		
<p><b>Literatura podstawowa:</b>                  1. K. Schwaber, J. Sutherland, The Scrum Guide: Przewodnik po Scrumie: Reguły Gry, <a href="http://www.scrumguides.org">http://www.scrumguides.org</a>, (dostępny online), 2017</p>		
<p><b>Literatura uzupełniająca:</b>                  1. Nawrocki, Jerzy, et al. Agile requirements engineering: A research perspective. International Conference on Current Trends in Theory and Practice of Informatics. Springer, Cham, 2014.</p>		
<p><b>Bilans nakładu pracy przeciętnego studenta</b></p>		
<p><b>Czynność</b></p>		<p><b>Czas (godz.)</b></p>
1. udział w zajęciach		60
2. przygotowanie do zajęć		5
3. udział w konsultacjach związanych z realizacją procesu kształcenia		2
4. udział w spotkaniach projektowych		20
5. zarządzanie projektem		20
6. praca nad aktualizacją dokumentacji projektowej		10
7. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi		8
<p><b>Obciążenie pracą studenta</b></p>		
<p><b>forma aktywności</b></p>	<p><b>godzin</b></p>	<p><b>ECTS</b></p>
Łączny nakład pracy	125	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	62	3
Zajęcia o charakterze praktycznym	110	4