

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Zarządzanie wymaganiami i wytwarzaniem oprogramowania</b>		Kod <b>1010515331010510646</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>2 / 3</b>
Ścieżka obieralności/specjalność <b>Technologie wytwarzania oprogramowania</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obligatoryjny</b>
Stopień studiów: <b>II stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>niestacjonarna</b>	
Godziny Wykłady: <b>16</b> Ćwiczenia: - Laboratoria: <b>16</b> Projekty/seminaria: -		Liczba punktów <b>4</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>		Podział ECTS (liczba i %) <b>4 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
<p>dr hab. inż. Andrzej Jaszkiwicz, prof. PP            email: Andrzej.Jaszkiwicz@cs.put.poznan.pl            tel. 61 6652933            Instytut Informatyki            ul. Piotrowo 2, 60-965 Poznań</p>		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_W1-2, K_W4, K_W6-15, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl
2	<b>Umiejętności:</b>	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_U1-2, K_U4, K_U7-8, K_U14-20, K_U22-23, K_U26, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl
3	<b>Kompetencje społeczne</b>	Efekty kształcenia ze studiów I stopnia zdefiniowane w Uchwale Senatu PP, a szczególnie efekty K_K1-9, weryfikowane w procesie rekrutacji na studia 2 stopnia ? efekty te prezentowane są w serwisie internetowym wydziału www.fc.put.poznan.pl
<b>Cel przedmiotu:</b>		
<p>11. Przekazanie studentom podstawowej wiedzy, w zakresie: zarządzania i inżynierii wymagań, planowania przedsięwzięcia programistycznego, szacowania kosztów oprogramowania, harmonogramowania i monitorowania przedsięwzięć programistycznych, zarządzania ryzykiem, zarządzania zasobami ludzkimi, pomiarów i zbierania danych, zapewniania jakości i zarządzania przedsięwzięciami realizowanym zgodnie ze żwawymi metodami,</p> <p>2. Rozwijanie u studentów umiejętności w zakresie planowania, organizacji i zarządzania przedsięwzięciami dotyczącymi wytwarzania oprogramowania oraz wykorzystania do tego celu odpowiednich narzędzi</p> <p>3. Kształtowanie u studentów umiejętności pracy zespołowej szczególnie w roli kierownika projektu/zespołu albo analityka</p>		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
<p>1. ma szczegółową wiedzę w zakresie wybranych działów zarządzania - [K_W3]</p> <p>2. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie inżynierii oprogramowania - [K_W4]</p> <p>3. ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki - [K_W5]</p> <p>4. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce i w wybranych pokrewnych dyscyplinach naukowych - [K_W6]</p> <p>5. ma podstawową wiedzę o cyklu życia systemów informatycznych sprzętowych lub programowych - [K_W7]</p> <p>6. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z wybranego obszaru informatyki - [K_W8]</p> <p>7. ma podstawową wiedzę dotyczącą zarządzania jakością, w tym podstawową wiedzę nt. standardów serii ISO 9000 - [K_W12]</p>		
<b>Umiejętności:</b>		

1. potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie - [K\_U1]
2. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia - [K\_U5]
3. potrafi posługiwać się technikami informacyjno-komunikacyjnymi wykorzystywanymi przy realizacji przedsięwzięć informatycznych - [K\_U7]
4. potrafi - przy formułowaniu i rozwiązywaniu zadań inżynierskich - integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K\_U10]
5. potrafi przeprowadzić analizę ryzyka związanego z przedsięwzięciem informatycznym - [K\_U11]
6. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K\_U13]
7. potrafi poprawnie użyć przynajmniej jedną metodę szacowania pracochłonności wytwarzania oprogramowania - [K\_U15]
8. potrafi efektywnie uczestniczyć w inspekcji oprogramowania - [K\_U19]
9. potrafi zaproponować ulepszenia (usprawnienia) istniejących rozwiązań technicznych - [K\_U21]
10. potrafi sformułować specyfikację funkcjonalną w formie przypadków użycia - [K\_U22]
11. potrafi sformułować wymagania pozafunkcyjne dla wybranych charakterystyk jakościowych - [K\_U23]
12. potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie lub ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych metod i narzędzi; - [K\_U24]

#### Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe, - [K\_K1]
2. zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życia - [K\_K4]
3. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K\_K5]
4. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K\_K6]
5. prawidłowo identyfikuje i rozstrzyga dylematy związane z wykonywaniem zawodu - [K\_K7]
6. potrafi myśleć i działać w sposób przedsiębiorczy - [K\_K8]

#### Sposoby sprawdzenia efektów kształcenia

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

- a) w zakresie wykładów:
  - na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,
- b) w zakresie laboratoriów / ćwiczeń:
  - na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

- a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:
    - ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym, egzamin składa się z 5 pytań otwartych ocenianych w skali 0-5 pkt. Obejmujących cały zakres wykładu. Zaliczenie (ocena 3,0) wymaga zdobycia minimum 13 pkt.
    - omówienie wyników egzaminu,
  - b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:
    - ocenianie ciągle, na poszczególnych zajęciach ? weryfikacja dostatecznego stopnia realizacji zadań zaplanowanych na zajęcia, adnotacja niewystarczającej realizacji zadań spowodowanej brakiem zaangażowania
    - ocena wybranego zadania (lub 2 zadań) dotyczącego przerobionych zagadnień, realizowanego częściowo na laboratoriach a częściowo w ramach pracy własnej
    - ocena wiedzy i umiejętności zdobytych na laboratoriach poprzez pisemny test
- Opcjonalne uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
  - uwagi związane z udoskonaleniem materiałów dydaktycznych.

#### Treści programowe

Program wykładu obejmuje następujące zagadnienia:

Zarządzanie i inżynieria wymagań: cele i kontekst systemu, wymagania funkcjonalne i нефункционаłne, sposoby pozyskiwania wymagań, opis wymagań, analiza i negocjowanie wymagań, atestowanie wymagań, zarządzanie wymaganiami.

Planowanie przedsięwzięcia: modele cyklu życia oprogramowania, cykl życia projektu, składowe planu przedsięwzięcia, struktura organizacyjna projektu.

Szacowanie kosztów oprogramowania: metody algorytmiczne, metody eksperckie, metody dekompozycji, elementy analizy finansowej projektów.

Harmonogramowanie i monitorowanie przedsięwzięć programistycznych: definiowanie zadań, zależności kolejnościowe, definicja i przydział zasobów, szacowanie czasów trwania zadań, budowa harmonogramu, narzędzia harmonogramowania i zarządzania zadaniami.

Zarządzanie ryzykiem: ryzyko a podejmowanie decyzji, ryzyka w przedsięwzięciach programistycznych, identyfikacja i analiza ryzyka, strategię zarządzania ryzykiem, plan zarządzania ryzykiem.

Zarządzanie zasobami ludzkimi: pozyskiwanie, rozwój i motywacja pracowników, delegowanie zadań, praca grupowa, środowisko pracy, wskazówki dla liderów.

Pomiary i zbieranie danych: metryki i pomiary w przedsięwzięciu programistycznym, metoda goal-question-metric.

Zarządzanie konfiguracjami: zasady, modele i narzędzia zarządzania konfiguracjami.

Zapewniania jakości: kryteria jakości, standaryzacja, przeglądy i audyty, statystyczne zapewnianie jakości, zarządzanie przez jakość, modele CMM i ISO 900x.

Zarządzanie przedsięwzięciami realizowanymi zgodnie ze zwałymi metodykami: podstawowe praktyki i cykl życia lekkich metodyk, zalety i ograniczenia.

Program laboratorium obejmuje następujące zagadnienia:

Ogólny przegląd rodzajów narzędzi oraz technologii służących do organizacji projektu informatycznego i ogólnie wspomaganie wytwarzania oprogramowania, ze szczególnym uwzględnieniem aspektów praktyki ciągłej integracji (ang. continuous integration - CI). Nowoczesne systemy kontroli wersji (repozytoria kodu źródłowego) i ich rola w projektach informatycznych oraz organizacji pracy programistów. Podstawowe funkcje tych systemów, rodzaje architektury (scentralizowana i rozproszona) i różnice między nimi. Zadania praktyczne z dwóch przykładowych systemów kontroli wersji działających w różnych architekturach (np. technologie Git i Subversion). Automatyzacja cyklu budowania, testowania, raportów jakości, dokumentacji i publikacji oprogramowania ? ogólna koncepcja działania, cechy i zalety w kontekście organizacji przedsięwzięć programistycznych. Cechy (wady i zalety) i zastosowanie wybranych narzędzi technologicznych do automatyzacji (np. Ant, Groovy, Gradle) - zadania praktyczne polegające m.in. na użyciu i tworzeniu skryptów dla powyższych technologii. Serwery continuous integration ? ogólna idea i funkcje, przykłady technologii, zastosowanie praktyczne ? konfiguracja wybranego serwera CI (np. Jenkins) łącząca różne zagadnienia z poprzednich zadań laboratoryjnych. Zagadnienia inżynierii wymagań ? tworzenie i opis wymagań, szablon wymagania, cechy dobrych i złych wymagań, rodzaje wymagań, wymagania funkcjonalne i scenariusze, wymagania pozafunkcjonalne. Dobre praktyki inżynierii wymagań według Sommerville'a i Sawyera. Dokument specyfikacji wymagań (SRS) ? standard IEEE830: Recommended Practice for Software Requirements Specifications. Systemy i technologie służące do zarządzania zadaniami, zbierania informacji o błędach i tworzenia dokumentacji w projektach informatycznych ? funkcje, cechy. Zastosowania powyższych systemów ? zadania z wybranej technologii (np. Trac). Integracja powyższych systemów z innymi narzędziami wspomagającymi wytwarzanie oprogramowania ? integracja z repozytorium, integracja z IDE na przykładzie wybranej technologii (np. Mylyn + Eclipse). Harmonogramowanie i śledzenie postępów w projekcie informatycznym ? podstawowe pojęcia: zasoby, zadania, wykres Gantta, ścieżka krytyczna. Zdania praktyczne z harmonogramowania przy użyciu wybranej technologii (np. Microsoft Project). Cześć wymienionych wyżej treści programowych realizowana jest w ramach pracy własnej studenta.]

Metody dydaktyczne:

1. wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, dyskusja.
2. ćwiczenia laboratoryjne: słowne wprowadzenie, notatka elektroniczna, prezentacja multimedialna, zadania praktyczne typu 'krok po kroku' (ang. tutorial), otwarte zadania praktyczne.

### Literatura podstawowa:

1. Inżynieria oprogramowania, A. Jaskiewicz, Helion, Gliwice, 1997.
2. Inżynieria oprogramowania, I. Sommerville, WNT, 2003.
3. Praktyczne podejście do inżynierii oprogramowania, R. S. Pressman, WNT, 2004.
4. Metodyki zarządzania projektami informatycznymi, Z. Szyjewski, PLACET, 2004.
5. Zarządzanie procesem tworzenia systemów informacyjnych, J. Cadle, D. Yeates, WNT, 2004.
6. Zarządzanie projektami informatycznymi, M. Flasiński, PWN, 2013.
7. Inżynieria oprogramowania, K. Sacha, PWN, 2010.
8. Zarządzanie wymaganiami, D. Leffingwell, D. Widrig, WNT, 2003.
9. Agile development. filozofia programowania zwinnego, J. Shore, S. Warden, Helion, 2008.

<b>Literatura uzupełniająca:</b>		
1. Requirements Engineering: A Good Practice Guide, I. Sommerville, P. Sawyer, Wiley, 1997.		
2. APM Agile Project Management. Jak tworzyć innowacyjne produkty, J. Highsmith, PWN, 2005.		
3. 29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering.		
4. Continuous Integration, M. Fowler, 2006, <a href="http://www.martinfowler.com/articles/continuousIntegration.html">http://www.martinfowler.com/articles/continuousIntegration.html</a> .		
5. Managing Software Development with Trac and Subversion, D. J. Murphy, Packt Publishing, 2007.		
6. Building and Testing with Gradle, T. Berglund, M. McCullough, O'Reilly Media, 2011.		
<b>Bilans nakładu pracy przeciętnego studenta</b>		
<b>Czynność</b>	<b>Czas (godz.)</b>	
1. udział w zajęciach laboratoryjnych / ćwiczeniach :	16	
2. przygotowanie do ćwiczeń laboratoryjnych:	7	
3. dokończenie ocenianego zadania lub zadań z ćwiczeń laboratoryjnych:	14	
4. udział w konsultacjach związanych z realizacją procesu kształcenia	6	
5. przygotowanie do testu pisemnego na laboratorium	6	
6. udział w wykładach	16	
7. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi	16	
8. przygotowanie do egzaminu	15	
9. przeprowadzenie egzaminu	2	
10. omówienie wyników egzaminu	2	
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	100	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	42	1
Zajęcia o charakterze praktycznym	37	1