



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Studio rozwoju oprogramowania 2 [S2Inf1E-IO>SDS2]

Przedmiot

Kierunek studiów

Informatyka/Computing

Rok/Semestr

1/2

Studia w zakresie (specjalność)

Inżynieria oprogramowania

Profil studiów

ogólnoakademicki

Poziom studiów

drugiego stopnia

Język oferowanego przedmiotu

angielski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

0

Laboratorium

0

Inne

0

Ćwiczenia

0

Projekty/seminaria

75

Liczba punktów ECTS

6,00

Koordynatorzy

dr hab. inż. Mirosław Ochodek prof. PP
miroslaw.ochodek@put.poznan.pl

dr inż. Sylwia Kopczyńska
sylwia.kopczynska@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać wiedzę z zakresu zarządzania projektami (w tym szczegółową wiedzę na temat metodyki Scrum), inżynierii wymagań oraz programowania. Powinien posiadać także umiejętności związane z zarządzaniem projektem według metodyki Scrum (w szczególności zarządzanie Rejestrem Produktu, planowanie sprintów i wydań, organizowanie spotkań projektowych zdefiniowanych w metodyce Scrum). Ponadto powinien również posiadać umiejętność pozyskiwania informacji ze wskazanych źródeł oraz mieć gotowość do podjęcia współpracy w ramach zespołu.

Cel przedmiotu

1) 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. 2) 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 pracą zespołu programistycznego, inżynierią wymagań (opieka nad Rejestrem Produktu) oraz projektowaniem i dokumentowaniem architektury systemu informatycznego. 3) Kształtowanie u studentów umiejętności pracy zespołowej. Przedmiot stanowi rozszerzenie przedmiotu Studio rozwoju oprogramowania 1.

Przedmiotowe efekty uczenia się

Wiedza:

1. ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną na temat zwinnych podejść do zarządzania projektami (np. manifest zwinności, najpopularniejsze zwinne metodyki wytwarzania oprogramowania).
2. ma zaawansowaną wiedzę szczegółową dotyczącą metodyki scrum.
3. ma zaawansowaną i szczegółową wiedzę o procesie rozwoju oprogramowania proponowanego w frameworku scrum.

Umiejętności:

1. potrafi organizować pracę zespołu programistycznego, a także używać narzędzia do zarządzania zadaniami w projekcie informatycznym (np. jira).
2. potrafi pozyskać i zintegrować wiedzę techniczną oraz dziedzinową, aby prowadzić rozmowy z użytkownikami i/lub klientem, aby wyznaczyć kierunki rozwoju produktu informatycznego.
3. potrafi wykorzystać mechanizm retrospektywy do analizy problemów i usprawnienia procesu realizacji projektu informatycznego.
4. potrafi zaprojektować system informatyczny spełniający postawione wymagania.
5. potrafi zarządzać rozwojem produktu informatycznego.

Kompetencje społeczne:

1. jest świadomy(a) kierunku i tempa rozwoju metodyk zarządzania projektami.
2. rozumie znaczenie potrzeby rozwoju metod i technologii informatycznych oraz dostosowywania ich do potrzeb realizowanego projektu.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

- na podstawie odpowiedzi udzielanych odnośnie do realizacji zadań związanych z projektem;
- na podstawie oceny bieżącego postępu realizacji zadań projektowych.

Ocena podsumowująca:

Wyznaczana na podstawie oceny czterech kryteriów (średniej ważonej liczby punktów procentowych):

- wyniki testu wiedzy (m.in. na podstawie testów certyfikacyjnych scrum.org) (średnia ocena z testów => 0-100%) – waga 0,15;
- ocena warsztatu przygotowanego przez grupę projektową (0-100%) – waga 0,15.
- cotygodniowa ocena postępów realizacji projektu uwzględniająca – planowanie, praktyki i produktu – waga 0,4 oraz wsparcie zespołu programistycznego (artefakty przygotowanie na potrzeby projektu) – waga 0,2.
- aktywne uczestnictwo w zajęciach (0-100%) – waga 0,10;

Ocena wynikowa jest określona według następującej skali:

- [90%, 100%] - 5,0
- [80%, 90%) - 4,5
- [70%, 80%) - 4,0
- [60%, 70%) - 3,5
- [50%, 60%) - 3,0

- [0%, 50%) - 2,0

Treści programowe

W ramach modułu Studio rozwoju oprogramowania (I i II) studenci nadzorują realizację projektu informatycznego, którego celem jest rozwiązanie rzeczywistego problemu zamawiającego poprzez dostarczenie oprogramowania.

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.

Realizacja zajęć obejmuje następujące etapy projektowe: dopracowanie architektury; konstrukcja oprogramowania.

W ramach realizacji powyższych etapów poruszane są następujące zagadnienia:

- struktura zespołu projektowego i odpowiedzialności;
- zarządzanie projektem oraz etapem projektu (planowanie wydania; delegowanie zadań w zespole, zarządzanie zmianą, tworzenie planu oraz podsumowania wydania);
- monitorowanie postępu projektu oraz podejmowanie decyzji na podstawie faktów;
- zapewnianie jakości oprogramowania (testy akceptacyjne dotyczące wymagań funkcjonalnych oraz pozafunkcjonalnych);
- architektura oprogramowania (aktualizacja architektury oprogramowania w odpowiedzi na zmiany wymagań, technologie tworzenia systemów informatycznych);
- warsztaty retrospekcji (ang. reflection workshops);
- wdrożenie oprogramowania (formalne odbiory systemów informatycznych, przygotowanie oprogramowania do przekazania);
- zarządzanie ryzykiem (identyfikacja, analiza oraz przeciwdziałanie ryzyku).

Tematyka zajęć

1. Spotkanie organizacyjne - przedstawienie zasad realizacji przedmiotu
2. Cykliczne spotkania z mentorem (cotygodniowe) - prezentacja postępów w realizacji projektu, omówienie napotkanych problemów i środków zaradczych, analiza zmierzonych ryzyk, planowanie kolejnych kroków
3. Warsztaty tematyczne prowadzone w ramach całej grupy zajęciowej
4. Nadzór nad realizacją projektu obejmujący:
 - Dopracowanie i aktualizację architektury oprogramowania
 - Planowanie i prowadzenie sprintów
 - Zapewnianie jakości organizacji spotkań Scrum (Daily Standup, Sprint Planning, Sprint Review, Sprint Retrospective)
 - Monitorowanie postępu prac i jakości kodu
 - Zarządzanie rejestrem produktu
 - Spotkania z klientem i zespołem programistów
5. Finalizacja projektu - przekazanie oprogramowania klientowi, przygotowanie dokumentacji końcowej, podsumowanie realizacji projektu.

Metody dydaktyczne

Zajęcia realizowane są w modelu projektowym (ang. capstone projects). W części szkoleniowej wykorzystuje się prezentacje multimedialne oraz metodę analizy przypadków.

Więcej informacji na temat stosowanych metod dydaktycznych przedstawiono w artykule:

Kopczyńska, Sylwia, Jerzy Nawrocki, and Mirosław 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.

Literatura

Podstawowa

1. K. Schwaber, J. Sutherland, The Scrum Guide: Przewodnik po Scrumie: Reguły Gry, <http://www.scrumguides.org>, (dostępny online), 2017

Uzupełniająca

1. Nawrocki, Jerzy, et al. Agile requirements engineering: A research perspective. International

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	150	6,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	75	3,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	75	3,00