

WYDZIAŁ W-8 / STUDIUM.....

KARTA PRZEDMIOTUNazwa w języku polskim *Zwinne Metodyki Wytwarzania Oprogramowania*Nazwa w języku angielskim *Agile Software Development*Kierunek studiów (jeśli dotyczy): *Informatyka*

Specjalność (jeśli dotyczy):

Stopień studiów i forma: **I/ II stopień*, stacjonarna / niestacjonarna***Rodzaj przedmiotu: **obowiązkowy / wybieralny / ogólnouczelniany ***Kod przedmiotu **INZ3814**Grupa kursów **TAK / NIE***

	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium
Liczba godzin zajęć zorganizowanych w Uczelni (ZZU)	9			18	9
Liczba godzin całkowitego nakładu pracy studenta (CNPS)	30			90	30
Forma zaliczenia	Egzamin / zaliczenie na ocenę*	Egzamin / zaliczenie na ocenę*	Egzamin / zaliczenie na ocenę*	Egzamin / zaliczenie na ocenę*	Egzamin / zaliczenie na ocenę*
Dla grupy kursów zaznaczyć kurs końcowy (X)				X	
Liczba punktów ECTS	1			3	1
w tym liczba punktów odpowiadająca zajęciom o charakterze praktycznym (P)	0			3	0
w tym liczba punktów ECTS odpowiadająca zajęciom wymagającym bezpośredniego kontaktu (BK)	1			2	1

*niepotrzebne skreślić

WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1. Stosuje język programowania (np. Java, Scala) do rozwiązania problemów programistycznych i tworzenia aplikacji.

CELE PRZEDMIOTU

C1 Zapoznanie z wybranymi, najistotniejszymi, nowymi metodykami i praktykami zwinnego wytwarzania oprogramowania

PRZEDMIOTOWE EFEKTY KSZTAŁCENIA

Z zakresu wiedzy:

PEK_W01 Ma wiedzę, która pozwala wymienić i scharakteryzować wybrane, najistotniejsze zwinne metodyki i praktyki wytwarzania oprogramowania.

Z zakresu umiejętności:

PEK_U01 Rozwiązuje złożone zadania inżynierskie w zakresie instalacji i konfiguracji infrastruktury projektowej oraz realizacji projektu wykorzystując wybrane praktyki zwinnego wytwarzania oprogramowania.

PEK_U02 Potrafi przygotować prezentację multimedialną dotyczącą zagadnień z zakresu przedmiotu.

TREŚCI PROGRAMOWE

Forma zajęć – wykład		Liczba godzin
Wy1	Wprowadzenie do tematyki przedmiotu. Opis programu kursu, organizacji zajęć i zasad zaliczania.	1
Wy2	Od braku procesu, poprzez ciężki proces do zwinnych metodyk wytwarzania oprogramowania i eXtreme Programming	2
Wy3	Metodyka eXtreme Programming, Lean	2
Wy4	Metodyka SCRUM, KANBAN, kolokwium	2
Wy5	Retrospekcja, kolokwium.	2
	Suma godzin	9

Forma zajęć – projekt		Liczba godzin
Pr1	Zajęcia organizacyjne. Szkolenie bhp. Zapoznanie z warunkami zaliczenia i tematyką zajęć projektowych.	2
Pr2	Infrastruktura projektowa, opanowanie metodyki, praktyk i narzędzi realizacji projektu	2
Pr3-5	Realizacja projektu – iteracja 1	6
Pr6-8	Realizacja projektu – iteracja 2	6
Pr9	Podsumowanie uzyskanych rezultatów, retrospekcja i wystawienie ocen.	2
	Suma godzin	18

Forma zajęć – seminarium		Liczba godzin
Se1	Zajęcia organizacyjne, zapoznanie z warunkami zaliczenia, określenie harmonogramu i tematów wystąpień seminaryjnych.	1
Se2	Temat 1 wybierany przez studentów z puli tematów dotyczących zwinnych metodyk, praktyk i narzędzi wytwarzania oprogramowania np.: XP, SCRUM, KANBAN, LEAN, CRYSTAL, Test-Driven Development, Continuous Test-Driven Development, Acceptance Test-Driven Development, Behaviour-Driven Development, Pair-Programming, Continuous Delivery, narzędzia do zwinnego wytwarzania oprogramowania w Scali, narzędzia do zwinnego wytwarzania oprogramowania w Javie, narzędzia do	2

	testowania oprogramowania, programowania parami i przeglądów kodu, dobre praktyki zwinnego wytwarzania aplikacji webowych, szkielety aplikacji webowych wspierające zwinne tworzenie oprogramowania – Play framework i in.	
Se3	Temat 2 wybierany z puli tematów j.w.	2
Se4	Temat 3 wybierany z puli tematów j.w.	2
Se5	Temat 4 wybierany z puli tematów j.w.	2
	Suma godzin	9

STOSOWANE NARZĘDZIA DYDAKTYCZNE	
<p>N1. Wykład informacyjny</p> <p>N2. Strona przedmiotu używana do publikacji materiałów dydaktycznych, ogłoszeń i linków do liczących się pozycji literaturowych.</p> <p>N3. Oprogramowanie na stacje robocze wspomagające zwinne wytwarzanie oprogramowania.</p> <p>N4. Infrastruktura serwerowa wspomagająca realizację projektu (wersjonowane repozytorium kodu).</p> <p>N5. Oprogramowanie do przygotowania prezentacji i raportów (Latex, TeXnicCenter, PowerPoint/Beamer)</p>	

OCENA OSIĄGNIĘCIA PRZEDMIOTOWYCH EFEKTÓW KSZTAŁCENIA

Oceny (F – formująca (w trakcie semestru), P – podsumowująca (na koniec semestru))	Numer efektu kształcenia	Sposób oceny osiągnięcia efektu kształcenia
F1 – projekt wstęp	PEK_U01	Ocena opanowania narzędzi i infrastruktury realizacji projektu oraz postępowania zgodnie z przyjętą metodyką, praktykami, i wytycznymi [0...10 pkt] (Pr2) .
F2 – projekt	PEK_U01	Ocena realizacji projektu oraz zgodności z założoną metodyką oraz praktykami, wytycznymi, narzędziami wytwarzania oprogramowania [0...100 pkt] (Pr3-Pr8).
P1 – ocena końcowa z projektu	PEK_U01	Ocena wyznaczona na podstawie sumy punktów z ocen formujących F1...F2 zgodnie z formułą (ocena – zakres punktów): <ul style="list-style-type: none"> • 5.5 – 100...110 pkt • 5.0 – 90...99 pkt • 4.5 – 80...89 pkt • 4.0 – 70...79 pkt • 3.5 – 60...69 pkt • 3.0 – 50...59 pkt • 2.0 <50 pkt
P2 – ocena końcowa z wykładu	PEK_W01	Kolokwium - test pisemny sprawdzający wiedzę i umiejętności z zakresu wykładu. Z testu przyznawana jest ocena pozytywna, jeżeli student zdobędzie przynajmniej określony procent maksymalnej liczby punktów, zależny od skali trudności zadań i

		nie wyższy niż 50%.
P3 – ocena końcowa z seminarium	PEK_U02	Ocena z przygotowanego i zaprezentowanego w czasie zajęć seminaryjnych tematu. Pod uwagę brane będzie jak dobrze została przeprowadzona prezentacja (warto zwrócić uwagę np. na jasno zdefiniowane cele prezentacji i korzyści dla słuchaczy, poprawną komunikację, właściwy poziom szczegółowości prezentacji i dobór przykładów, właściwy czas prezentacji, kolejność poruszanych tematów, wciągnięcie słuchaczy). Ocena może być podwyższona za aktywne uczestnictwo oraz konstruktywne i twórczy wkład w zajęcia seminaryjne (dyskusja) lub (po uzgodnieniu z prowadzącym) szczegółowe opracowanie prezentowanego zagadnienia np. w postaci tutoriala.
P – ocena końcowa z grupy kursów	PEK_W01, PEK_U01, PEK_U02	Warunkiem uzyskania pozytywnej oceny z grupy kursów jest pozytywna ocena z projektu. W przypadku uzyskania pozytywnej oceny z projektu ocena z grupy kursów jest średnią ważoną ocen z wykładu, projektu i seminarium $(W+2*P+S)/4$. W przypadku uzyskania negatywnej oceny z projektu ocena z grupy kursów jest również negatywna.

LITERATURA PODSTAWOWA I UZUPEŁNIAJĄCA

LITERATURA PODSTAWOWA:

- [1] Kent Beck, Cynthia Andres, Wydajne programowanie = Extreme Programming, MIKOM, 2006.
- [2] Henrik Kniberg, Scrum and XP from the Trenches
<http://www.infoq.com/minibooks/scrum-xp-from-the-trenches>
- [3] Henrik Kniberg, Mattias Skarin, Kanban and Scrum – making the most of both
<http://www.infoq.com/minibooks/kanban-scrum-minibook>
- [4] Jeff Langr, Agile Java : crafting code with test-driven development, Prentice Hall 2005.

LITERATURA UZUPEŁNIAJĄCA:

- [1] Lech Madeyski: Test-Driven Development - An Empirical Evaluation of Agile Practice. Springer 2010 <http://dx.doi.org/10.1007/978-3-642-04288-1>
- [2] Ron Jeffries, Programowanie ekstremalne w C#, Warszawa : APN PROMISE, 2005.
- [3] Tore Dybå, Helen Sharp: What's the Evidence for Lean? IEEE Software 29(5): 19-21 (2012) <http://doi.ieeecomputersociety.org/10.1109/MS.2012.126>
- [4] Tore Dybå, Torgeir Dingsøyr: What Do We Know about Agile Software Development? IEEE Software 26(5): 6-9 (2009)
<http://doi.ieeecomputersociety.org/10.1109/MS.2009.145>

- [5] Tore Dybå, Torgeir Dingsøy: Empirical studies of agile software development: A systematic review. *Information & Software Technology* 50(9-10): 833-859 (2008)
<http://dx.doi.org/10.1016/j.infsof.2008.01.006>
- [6] Lech Madeyski, Marcin Kawalerowicz, “Continuous Test-Driven Development — A Novel Agile Software Development Practice and Supporting Tool”, *Proceedings of the 8th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE’2013)*, p.209–216.
<http://madeyski.e-informatyka.pl/download/Madeyski13ENASE.pdf>
- [7] Lech Madeyski: The impact of Test-First programming on branch coverage and mutation score indicator of unit tests: An experiment. *Information & Software Technology (INFOSOF)* 52(2):169-184 (2010)
<http://dx.doi.org/10.1016/j.infsof.2009.08.007>
<http://madeyski.e-informatyka.pl/download/Madeyski10c.pdf>
- [8] Lech Madeyski: Impact of pair programming on thoroughness and fault detection effectiveness of unit test suites. *Software Process: Improvement and Practice (SOPR)* 13(3):281-295 (2008)
<http://dx.doi.org/10.1002/spip.382>
<http://madeyski.e-informatyka.pl/download/Madeyski08.pdf>
- [9] Lech Madeyski, Wojciech Biela: Capable Leader and Skilled and Motivated Team Practices to Introduce eXtreme Programming. *CEE-SET* 2007:96-102
http://dx.doi.org/10.1007/978-3-540-85279-7_8
<http://madeyski.e-informatyka.pl/download/Madeyski08d.pdf>
- [10] Lech Madeyski, Lukasz Szala: The Impact of Test-Driven Development on Software Development Productivity - An Empirical Study. *EuroSPI* 2007:200-211
http://dx.doi.org/10.1007/978-3-540-75381-0_18
<http://madeyski.e-informatyka.pl/download/Madeyski07d.pdf>
- [11] Lech Madeyski: On the Effects of Pair Programming on Thoroughness and Fault-Finding Effectiveness of Unit Tests. *PROFES* 2007:207-221
http://dx.doi.org/10.1007/978-3-540-73460-4_20
<http://madeyski.e-informatyka.pl/download/Madeyski07.pdf>
- [12] Lech Madeyski, Wojciech Biela: Empirical Evidence Principle and Joint Engagement Practice to Introduce XP. *XP* 2007:141-144
http://dx.doi.org/10.1007/978-3-540-73101-6_19
<http://madeyski.e-informatyka.pl/download/Madeyski07b.pdf>
- [13] Lech Madeyski: The Impact of Pair Programming and Test-Driven Development on Package Dependencies in Object-Oriented Design - An Experiment. *PROFES* 2006:278-289
http://dx.doi.org/10.1007/11767718_24
<http://madeyski.e-informatyka.pl/download/Madeyski06.pdf>
- [14] Lech Madeyski: Is External Code Quality Correlated with Programming Experience or Feelgood Factor? *XP* 2006:65-74
http://dx.doi.org/10.1007/11774129_7
<http://madeyski.e-informatyka.pl/download/Madeyski06b.pdf>
- [15] Lech Madeyski: Preliminary Analysis of the Effects of Pair Programming and Test-Driven Development on the External Code Quality. *Software Engineering: Evolution and Emerging Technologies* 2005:113-123
<http://www.booksonline.iospress.nl/Content/View.aspx?piid=1150>
<http://madeyski.e-informatyka.pl/download/Madeyski05b.pdf>
- [16] Mary Poppendieck, *Leading lean software development : results are not the point*, Addison-Wesley, 2011.

OPIEKUN PRZEDMIOTU (IMIE, NAZWISKO, ADRES E-MAIL)	
Lech Madeyski	
Lech.Madeyski /at/ pwr.wroc.pl	http://madeyski.e-informatyka.pl/

MACIERZ POWIĄZANIA EFEKTÓW KSZTAŁCENIA DLA PRZEDMIOTU
Zwinne Metodyki Wytwarzania Oprogramowania
Z EFEKTAMI KSZTAŁCENIA NA KIERUNKU Informatyka
I SPECJALNOŚCI Inżynieria Oprogramowania.

Przedmiotowy efekt kształcenia	Odniesienie przedmiotowego efektu do efektów kształcenia zdefiniowanych dla kierunku studiów i specjalności (o ile dotyczy)**	Cele przedmiotu* **	Treści programowe***	Numer narzędzia dydaktycznego***
PEK_W01 (wiedza)	K2INF_W06(K2INF_W06_S2IO_W10)	C1	Wy1...	N1, N2
PEK_U01 (umiejętności)	K2INF_U07	C1, C2	Pr2...	N3, N4
PEK_U02	K2INF_U08(K2INF_U08_S2IO_U10)	C1, C2	Se2...	N5

** - wpisać symbole kierunkowych/specjalnościowych efektów kształcenia

*** - z tabeli powyżej