



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Zarządzanie jakością i eksperymentalna inżynieria oprogramowania [S2Inf1E-IO>QMAN]

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

60

Liczba punktów ECTS

5,00

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 powinien posiadać podstawową wiedzę z zakresu inżynierii oprogramowania oraz podstawową wiedzę z zakresu matematyki i statystyki. Ponadto powinien posiadać podstawowe umiejętności w zakresie formułowania i testowania prostych hipotez statystycznych, umiejętności niezbędne do przygotowania krótkiego opracowania naukowego oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.

Cel przedmiotu

1) Przekazanie studentom podstawowej wiedzy związanej z eksperymentalną inżynierią oprogramowania, w szczególności w zakresie empirycznych metod badawczych oraz ich podbudowy teoretycznej, 2) Przekazanie studentom podstawowej wiedzy z zakresu zarządzania jakością w organizacjach wytwarzających systemy informatyczne w zakresie systemów zarządzania jakością, metod oceny poziomu dojrzałości oraz ciągłego doskonalenia procesów w organizacji, 3) Rozwijanie u studentów umiejętności rozwiązywania problemów związanych z oceną metod, narzędzi oraz zjawisk w obszarze inżynierii oprogramowania z wykorzystaniem metod empirycznych, 4) Rozwijanie u studentów umiejętności związanych z oceną i poprawą jakości procesu wytwarzania oprogramowania w organizacji.

Przedmiotowe efekty uczenia się

Wiedza:

1. ma zaawansowaną i pogłębioną wiedzę na temat metod empirycznych stosowanych w inżynierii oprogramowania.
2. ma zaawansowaną i pogłębioną wiedzę na temat pomiaru w procesie wytwarzania oprogramowania (gqm+strategies, skale pomiarowe, rodzaje metryk).
3. ma zaawansowaną wiedzę szczegółową dotyczącą inspekcji i przeglądów kodu.
4. ma wiedzę ogólną na temat metod zarządzania jakością na poziomie organizacji i oceny dojrzałości procesów wytwarzania oprogramowania (np. iso 9001, tqm, itil, cmmi).

Umiejętności:

1. potrafi pozyskiwać informacje z literatury także przeprowadzając systematyczny przegląd literatury.
2. potrafi dobrać metodę empiryczną do sformułowanego problemu badawczego / pytania badawczego (np. eksperyment, studium przypadku, badanie ankietowe, meta-analiza).
3. potrafi planować i przeprowadzać badania o charakterze empirycznym oraz interpretować uzyskane wyniki i wyciągać wnioski.
4. potrafi przeprowadzać przeglądy kodu.

Kompetencje społeczne:

1. rozumie znaczenie i potrzebę prowadzenia badań empirycznych w celu rozwoju i oceny nowych metod i narzędzi służących do rozwoju oprogramowania.
2. rozumie potrzebę rzetelnego opisywania wyników badań naukowych.
3. ma świadomość znaczenia potrzeby przestrzegania zasad etyki w trakcie prowadzenia badań naukowych (np. postępowania z danymi wrażliwymi).

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena wyznaczana jest na podstawie trzech kryteriów (średnia ważona): (1) podejścia do realizacja projektu badawczego przygotowania raportu z przeprowadzenia badania (waga 0.6), (2) prezentacji w trakcie zajęć (waga 0.20) oraz (3) recenzji pracy innej grupy studentów (waga 0.2).

Ocena wynikowa 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

Zarządzanie jakością oprogramowania:

- zarządzanie jakością w procesie wytwarzania oprogramowania (pojęcie jakości; znaczenie jakości; koszt jakości),
- ciągłe doskonalenie (podejścia: Plan-Do-Check-Act, TQM),
- systemy zarządzania jakością i normy z serii ISO9000 (pojęcie systemu zarządzania jakością; struktura normy ISO9001; zasady jakości),
- ocena dojrzałości procesów w organizacji w oparciu o model CMMI oraz SCAMPI,
- dobre praktyki zarządzania usługami ITIL ,
- przeglądy i inspekcje (pojęcie inspekcji i przeglądu artefaktu; proces przeglądu i inspekcji oraz ich rezultaty).

Eksperymentalna inżynieria oprogramowania:

- badania empiryczne w inżynierii oprogramowania (rola badań empirycznych w ocenie metod i narzędzi inżynierii oprogramowania; pojęcia obserwacji, prawa oraz teorii i zależności między nimi; formułowanie hipotez i pytań badawczych; metody ilościowe i jakościowe),
- skale pomiarowe (definicja oraz własności skali nominalnej, porządkowej, interwałowej oraz ilorazowej, rodzaje błędów pomiarowych),
- kontrolowany eksperyment (cele przeprowadzania kontrolowanych eksperymentów; definicja eksperymentu; wybór kontekstu; formułowanie hipotez eksperymentu; wybór zmiennych: zmienne

zależne i niezależne; wybór próby; przygotowanie eksperymentu; zbieranie danych; walidacja danych z wykorzystaniem metod statystycznych; wizualizacja i analiza rozkładów cechy; testowanie hipotez testami statystycznymi; analiza mocy testu statystycznego, narzędzia informatyczne wspierające proces analizy danych z eksperymentów; interpretacja wyników eksperymentu; typy zagrożeń do poprawności eksperymentu),

- studium przypadku (cele przeprowadzania studium przypadku; planowanie studium przypadku; zbieranie danych i obserwacji; analiza danych w studium przypadku; raportowanie wyników studium przypadku),

- meta-analiza (cele przeprowadzania meta-analzy; systematyczne przeglądy literatury: planowanie oraz przygotowanie protokołu przeglądu, przeprowadzenie przeglądu, dokumentowanie przeglądu),

- badania ankietowe (cele prowadzenia badań ankietowych; formy ankietowania; tworzenie ankiety; ewaluacja ankiety; analiza danych pochodzących z badania ankietowego),

W ramach zajęć o charakterze projektowym studenci realizują projekt badawczy, który wymaga użycia przynajmniej jednej z następujących metod empirycznych: kontrolowanego eksperymentu, studium przypadku, badania ankietowego, lub meta-analzy. Celem każdego projektu jest udzielenie odpowiedzi na określone pytanie badawcze (jeśli to możliwe z zakresu zarządzania jakością oprogramowania). Rezultaty projektu zostają zawarte w raporcie oraz są prezentowane w trakcie zajęć.

Tematyka zajęć

Program zajęć obejmuje następujące zagadnienia:

1. Seria zajęć wykładowo-warsztatowych z zakresu zarządzania jakością oprogramowania:

- Wprowadzenie do zarządzania jakością oprogramowania

- Modele jakości (SQUALE)

- Przeglądy i inspekcje kodu oraz artefaktów

- Jakość produktów i procesów (CMMI, ISO 9000, ISO 27000, podejścia do poprawy procesów)

- Metodyki Lean, Kanban, TQM

- Systemy miar i celów (GQM+Strategies, KPI, OKR)

- Zarządzanie usługami IT (ITIL)

- Skalowanie metodyk zwinnych

2. Seria zajęć wykładowo-warsztatowych z zakresu empirycznej inżynierii oprogramowania:

- Wprowadzenie do empirycznej inżynierii oprogramowania

- Metody badawcze: kontrolowany eksperyment, studium przypadku, badania ankietowe, meta-analiza

- Prezentacje artykułów naukowych (w kontekście wykorzystania empirycznych metod badawczych)

3. Realizacja mini-projektu badawczego:

- Przeprowadzenie kontrolowanego eksperymentu z udziałem uczestników kursu dotyczącego jakości produktów lub procesów

- Przygotowanie raportu z badania

- Wzajemna ocena raportów między grupami

Metody dydaktyczne

Prezentacja multimedialna, moderowana dyskusja, studium przypadku, ćwiczenia praktyczne (rozwiązywanie zadań podanych przez prowadzących) oraz realizacja projektu.

Literatura

Podstawowa

1. C. Wohlin, P. Runeson, M. Host, M. Ohlsson, B. Regnell, and A. Wesslen: Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers, 2000.

2. Gordon G. Schulmeyer: Handbook of Software Quality Assurance, ISBN-13: 978-1596931862, Artech House Publishers, 2007.

Uzupełniająca

1. Ochodek, Mirosław, et al. Improving the reliability of transaction identification in use cases. Information and Software Technology 53.8 (2011): 885-897.

2. Ochodek, Mirosław, and Sylwia Kopczyńska. Perceived importance of agile requirements engineering practices - A survey. Journal of Systems and Software 143 (2018): 29-43.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	125	5,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwii/egzaminu, wykonanie projektu)	65	2,50