

Software engineering - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Software engineering
Kod przedmiotu	11.3-WE-INFP-SoftEng-Er
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	Program Erasmus pierwszego stopnia
Semestr rozpoczęcia	semestr zimowy 2020/2021

Informacje o przedmiocie	
Semestr	4
Liczba punktów ECTS do zdobycia	4
Typ przedmiotu	obowiązkowy
Język nauczania	angielski
Sylabus opracował	- dr inż. Tomasz Gratkowski - mgr inż. Marcin Andrzejewski - dr inż. Michał Doligalski

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	30	2	-	-	Zaliczenie na ocenę
Projekt	30	2	-	-	Zaliczenie na ocenę

Cel przedmiotu

- Familiarize students with methods of design, analysis and methods of testing programs.
- Teach students the fundamental skills in specification requirements, planning, documentation of IT projects.
- Familiarize students with with tools for object-oriented design and project management.

Wymagania wstępne

Algorithms and data structures, Principles of programming, Object-oriented programming

Zakres tematyczny

Introduction to software engineering. Why engineering software is different? Software lifespan and maintenance. Lifecycle models with specified project phases. Information systems. System and software design. Models for information systems. Software process. Requirements analysis and specification. Guidelines and forms for specification. Design. Purpose of design. Fundamental design concepts. Design strategies. Design quality metrics. Reliability and system security. Implementation. Review of structural programming. Error handling and defensive programming. Aids to maintainability. Coding for performance. Testing. Reasons for testing. Black box and structural testing. Testing strategies. Tools for testing Computer Aided Software Engineering tools. Upper and Lower CASE, CASE workbenches.

Metody kształcenia

Lecture, project

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Understands software distribution and maintenance problems, can work and communicate in a programming team		sprawdzian z programami punktowymi	- Wykład - Projekt
Can develop a project plan, documentation of requirements, requirement specification as well as functional and program specification, can also evaluate the quality of a project using appropriate tools		- obserwacje i ocena umiejętności praktycznych studenta - projekt	Projekt
Can define and characterize basic software production cycles		- obserwacje i ocena umiejętności praktycznych studenta - projekt	- Wykład - Projekt

Warunki zaliczenia

Lecture - obtaining a positive grade in written exam.

Project - a condition of pass is to obtain positive marks from all project tasks and preparation written report of project.

Calculation of the final grade: = lecture 50% + project 50%.

Literatura podstawowa

1. Ian Sommerville: Software Engineering (10th Edition), Pearson, 2015
2. Joel Spolsky: The Best Software Writing I: Selected and Introduced, Apress, 2005
3. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, Second Edition, Addison-Wesley Professional, 2003
4. Steve McConnell: Code Complete: A Practical Handbook of Software Construction, Second Edition 2nd Edition, Microsoft Press, 2004
5. Martin Fowler, Kent Beck, John Brant, William Opdyke, and Don Roberts: Refactoring - Improving the Design of Existing Code, Addison-Wesley Professional, 1999
6. Eric Freeman, Bert Bates, Kathy Sierra, Elisabeth Robson: Head First Design Patterns: A Brain-Friendly Guide 1st Edition, O'Reilly Media, 2004

Literatura uzupełniająca

Uwagi

Zmodyfikowane przez dr inż. Tomasz Gratkowski (ostatnia modyfikacja: 26-04-2020 20:54)

Wygenerowano automatycznie z systemu SylabUZ