(faculty stamp)

COURSE DESCRIPTION

1. Course title: SOFTWARE ENGINEERING	**2. Course code:** SE
3. Validity of course description: 2017/2018	
4. Level of studies: 1st cycle of higher education	
5. Mode of studies: intramural studies	
6. Field of study: MACROFACULTY	
7. Profile of studies: general academic	
8. Programme:	
9. Semester: VII	
10. Faculty teaching the course: Faculty of Automatic Control, Electronics and Computer Science	
11. Course instructor: dr inż. Przemysław Szmal	
12. Course classification: specialization courses	
13. Course status: compulsory	
14. Language of instruction: English	
15. Pre-requisite qualifications: completed courses: Fundamentals of Computer Programming, Computer Programming	
16. Course objectives: The course is aimed at a presentation of selected topics falling within the scope of Software Engineering particular attention to those that relate to the life cycle of limited size software units. Among other thing students become generally familiar with the design issues of information systems using modern method and CASE tools	

17. Description of learning outcomes:

Nr	Learning outcomes description	Method of assessment	Teaching methods	Learning outcomes reference code
1	General knowledge in the scope of Software Engineering	Test	Lecture	K1A_W14, K1A_W18
2	Ability to model IT (Information Technology) systems	Project task	Project	K1A_U21, K1A_U24
3	Ability to design software	Project task	Project	K1A_U24
4	Ability to use CASE tools	Project task	Project	K1A_U24
5	Ability to test and improve the performance of software	Laboratory task	Laboratory	K1A_U15

18. Teaching modes and hours
Lecture 30 h**/ BA/MA Seminar** – **/ Class** – **/ Project** – **/ Laboratory** 30 h

19. Syllabus description:

Lectures:
Introduction, origin and object of concern of Software Engineering. Software Engineering vs. Systems Engineering. I(nformation) T(echnology) project management. Software life-cycle models. Basics of agile development. Strategy phase. Requirements definition phase. Application of the CASE tools in the strategy and the requirements definition phase. Basics of the analysis phase. UML language and methodology. Building the object model. System design and Implementation. Software testing and reliability.

Laboratory:

1. The process of creating a software development team.
2. The choice and adaptation of software development methodology.
3. The process of gathering and engineering the requirements.
4. Use-case modelling.
    a. Use-case workshop. Use-cases against requirements matching.
5. Modelling the system architecture.
    a. Structural modelling.
    b. Behavioural modelling.
6. Group programming.
    a. Common code repository.
    b. Issue trackers.
    c. The process of assuring the software quality.
7. Time optimisation

20. Examination: no

21. Primary sources:

Sommerville, I: Software Engineering, (9th Edition),
https://edisciplinas.usp.br/mod/resource/view.php?id=1094198
Cockburn, Alistair. *Crystal clear: a human-powered methodology for small teams*. Pearson Education, 2004.
Sacha, K., Inżynieria oprogramowania, Wydawnictwo Naukowe PWN, Warszawa, 2010
Wrycza, S., Marcinkowski, B., Wyrzykowski, K., Język UML 2.0 w modelowaniu systemów informatycznych, Helion, Gliwice, 2006.
Dąbrowski, W., Stasiak, A., Wolski, M., Modelowanie systemów informatycznych w języku UML 2.1, PWN , Warszawa 2009.
Booch, G., Rumbaugh, J., Jacobson, I., The Unified Modeling Language User Guide, http://meusite.mackenzie.com.br/rogerio/the-unified-modeling-language-user-guide.9780201571684.997.pdf
Szmal, P. (ed.), Inżynieria programowania. Metody i ćwiczenia laboratoryjne, Wydawnictwo Politechniki Śląskiej, Gliwice, 2003

22. Secondary sources:

Pressman, R.S., Software Engineering: A Practitioner's Approach (7th Edition),
http://freebooks.dlworld.info/2012/10/download-software-engineering.html
Hunt, A., Thomas, D., The Pragmatic Programmer: From Journeyman to Master,
http://zsiie.icis.pcz.pl/ksiazki/Pragmatic%20Programmer.pdf
Stoica, Marian, Marinela Mircea, and Bogdan Ghilic-Micu. "Software Development: Agile vs. Traditional." Informatica Economica 17.4 (2013): 64.
Brhel, Manuel, et al. "Exploring principles of user-centered agile software development: A literature review." Information and Software Technology 61 (2015): 163-181.
Górski, J. (ed.), Inżynieria oprogramowania w projekcie informatycznym", wyd. II rozszerzone. Mikom, Warszawa 2000
Jaszkiewicz, A., Inżynieria oprogramowania, Helion, Gliwice 1997
Schneider, G., Winters, J., Stosowanie przypadków użycia, WNT, Warszawa 2004

23. Total workload required to achieve learning outcomes

Nr	Teaching mode:	Contact hours / Student workload hours
1	Lecture	30 / 15
2	Classes	- / -
3	Laboratory	30 / 45
4	Project	- / -
5	BA / MA Seminar	- / -
6	Other	- / -
	Total number of hours	60 / 60

24. Total hours: 120

25. Number of ECTS credits: 4

26. Number of ECTS credits allocated for contact hours: 2

27. Number of ECTS credits allocated for in-practice hours (laboratory classes, projects): 1

26. Comments: –

Approved:

…….. ……..

(date, instructor's signature) (*date , the Director of the Faculty Unit signature)*