STUDY MODULE DESCRIPTION FORM

Name of the module/subject **Software engineering**		Code **1010331561010330109**
Field of study **Information Engineering**	Profile of study (general academic, practical) **(brak)**	Year /Semester **3 / 6**
Elective path/specialty **Security of Information Technology (IT)**	Subject offered in: **Polish**	Course (compulsory, elective) **obligatory**
Cycle of study: **First-cycle studies**	Form of study (full-time,part-time) **full-time**	
No. of hours Lecture: **30** Classes: **-** Laboratory: **-** Project/seminars: **15**		No. of credits **4**
Status of the course in the study program (Basic, major, other) **(brak)**	(university-wide, from another field) **(brak)**	
Education areas and fields of science and art **technical sciences** **Technical sciences**		ECTS distribution (number and %) **4 100%** **4 100%**

Responsible for subject / lecturer:

dr inż. Andrzej Sikorski
email: andrzej.sikorski@put.poznan.pl
tel. 6653958
Wydział Elektryczny
ul. Piotrowo 3A 60-965 Poznań

Responsible for subject / lecturer:

dr inż. Adam Meissner
email: adam.meissner@put.poznan.pl
tel. 61 665 37 24
Faculty of Electrical Engineering
ul. Piotrowo 3A 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	Student has a theoretical and practical knowledge on software engineering. Student is knowledgeable with the state of art and modern trends in software engineering and computing. Knowledge of Visual Paradigm. computer science fundamentals with emphasize on OOP, fundamental algorithms (e.g. as given in AOCP vol.1) Model relacyjny.
2	**Skills**	Software engineering as covered in previous semester (class, use case and requirements diagrams) Proficiency in C,C++,C# or java. Podstawy baz danych. data base basics.
3	**Social competencies**	Ability to work in a team. Trustworthiness, loyalty and discretion.

Assumptions and objectives of the course:

Familiarizing a student with selected methods of software modelling and design and also with methods of software testing, validation and verification.

Knowledge of OOP and advanced programming & modeling techniques.

The impact of modeling on software quality.

Proficiency in UML modeling.

The main objective is to provide necessary knowledge and to support student project and lab work.

Study outcomes and reference to the educational results for a field of study

Knowledge:

1. Knowledge of Software Engineering and CASE tools (ie. Visual Paradigm) - [K_W12]
2. Konowledge of latest tools, technologies and trends within IT industry. - [K_W19]

Skills:

1. Knowledge aquisition from API, tools and software framework documentation. - [K_U16]
2. Ability to map the requirement to the functionality and strucutre offered by software tools. - [K_U03]

Social competencies:
1. Reliability and dependability. Understanding of the software modeling importance. - [K_K07]
2. Responsibility for the work results. - [K_K04]

Assessment methods of study outcomes
Examination. UML and coding assigments.
Seminary or mid-term exam.
Challenges offered by the lecturer.
Solution of technical problems presented within the lecture.

Course description
Course update 2017: Scrum methodology.
Dynamic UML diagrams: state, timing, interaction, sequence and activity.
Concurrent programming design patterns. UML specification of high level synchronization objects.
Real time system modeling. Relational design and modeling. Relational modeling. Relational division, semi anti-join, SQL query re-writing.
Formal methods. Agile programming and extreme programming. Scrum methodology. Methods of software validation, verification and testing.
Teaching methods:
- lectures supported by slides and examples presented on the table
- projects - a usage of tools enabling students to perform tasks at home, reviewing student project documentation with a discussion of common errors.

Basic bibliography:

1. Bath G., McKay J., The Software Test Engineer's Handbook, Rocky Nook, 2011
2. Paulish D.J., Architecture-Centric Software Project Management: A Practical Guide, Addison-Wesley Professional, 2001
3. Schwaber K., Sutherland J., The Scrum Guide TM. The Definitive Guide to Scrum: The Rules of the Game, July 2016, http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf
4. Shore S., Warden S., The Art of Agile Development, O'Reilly Media, 2007

Additional bibliography:

1. Jeffries R., Extreme Programming Adventures in C#, Microsoft Press, 2004
2. Rad N.K., Turley F., The Scrum Master Training Manual. A Guide to Passing the Professional Scrum Master (PSM) Exam, Management Plaza, 2013, https://mplaza.pm/downloads/Scrum%20Training%20Manual.pdf
3. Sutherland J., Jeff Sutherland?s Scrum Handbook, Scrum Training Institute Press, 2010, http://www.ugrad.cs.ubc.ca/~cs310/2014W1/slides/Sutherland_Scrum_Handbook.pdf

Result of average student's workload

Activity	**Time (working hours)**
1. Lecture	30
2. Individual activity	20
3. Project labs	15

Student's workload

Source of workload	**hours**	**ECTS**
Total workload	65	4
Contact hours	45	3
Practical activities	15	1