Poznan University of Technology
Faculty of Electrical Engineering

European **C**redit **T**ransfer **S**ystem

STUDY MODULE DESCRIPTION FORM

Name of the module/subject **Software engineering**	Code **1010334551010330109**

Field of study **Information Engineering**	Profile of study (general academic, practical) **(brak)**	Year /Semester **3 / 5**
Elective path/specialty **-**	Subject offered in: **Polish**	Course (compulsory, elective) **obligatory**

Cycle of study: **First-cycle studies**	Form of study (full-time,part-time) **part-time**

No. of hours Lecture: **12** Classes: **-** Laboratory: **8** Project/seminars: **-**	No. of credits **3**

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**	ECTS distribution (number and %**)** **3 100%**

Responsible for subject / lecturer:

dr hab. inż. Barbara Begier
email: Barbara.Begier@put.poznan.pl
tel. (61) 665-3724
Wydział Elektryczny
ul. Piotrowo 3A 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	Basic knowledge learnt at high school. Student has theoretical and partially practical knowledge concerning: programming constructions, implementation of algorithms, programming styles, verification of software correctness, formal languages, compilers, and platforms.
2	**Skills**	Student is able to find information from professional literature, databases and other sources; he/she can also integrate and correctly interpret the gained information and then to conclude and formulate his/her own opinions.
3	**Social competencies**	Student is aware of an importance of non-technical aspects and then consequences of software engineer's activities; he/she understands his/her responsibility for his/her decisions.

Assumptions and objectives of the course:

The aim of the two-semester course of software engineering is to present an engineering approach to software development. During the first semester students are taught to build a software object model using the UML standard. An overview of software life cycle models is presented.

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

Knowledge:

1. Student has basic knowledge concerning software enegineering: concept of MDA (Model Driven Architecture), object modeling using the UML standard, quality of a software process and product. - [K_W12]

2. Student is knowledgeable with the state of art and modern trends in software engineering and computing. - [K_W19]

Skills:

1. Student is able to formulate requirements, to build an object model, and assess a simple information system, its functions, and components. - [K_U16]

2. Student is able to prepare and present a short presentation about his/her own engineering solution. - [K_U04]

Social competencies:

1. Student has a broaded awareness of an importance of non-technical aspects and then consequences of software engineer - [K_K02]

2. Student is aware of his/her responsibility for the work done. He/she points out his/her readyness to work in team work and to be responsible for results of tasks realized in team. - [K_K04]

Assessment methods of study outcomes

The content of lectures presented in the first semester of the software engineering course is a subject of an exam after the second semester of this course. After the first semester student's work is assessed on a base of his/her activity in classes and results of a test.

Student's work in laboratories is assessed on the base of partial marks given for each UML diagram and other artefact (requirements document).

Course description

Lectures. Field of software engineering. Concept of MDA (Model Driven Architecture). Assumptions and elements of the UML standard: modeling of use cases, classes, bjects, interfaces, stereotypes, derived elements, packages, components. Modeling an object behavior using: statechart, activity diagram, interaction diagrams. Primary and supporting processes, including documenting, in software development. Overview of software life cycle models: waterfall, RAD, pyramid, V, spiral, WinWin, incremental, and iterative-incremental model. Specification of requirements. Repository. Overviews and software inspections. Process-oriented approach recommended in ISO 9000. Capability Maturity Model for Software. Key areas assigned to maturity levels in the CMM model.

Laboratories. Specifying software requirements. Development of software object model (use cases, objects, and classes) using the UML 2.0 standard.

Basic bibliography:

1. . Booch G., Jacobson I., Rumbaugh J., The Uified Modeling Language User?s Guide, Addison-Wesley, Boston.

2. Wrycza St., Marcinkowski B., Wyrzykowski K., Język UML 2.0 w modelowaniu systemów informatycznych, Helion, Gliwice 2005 (and further editions).

Additional bibliography:

1. Begier B., Inżynieria oprogramowania - problematyka jakości, Wydawnictwo Politechniki Pozn., Poznań 1999.

2. Hamlet D., Maybee J., Podstawy techniczne inżynierii oprogramowania, WNT, Warszawa 2003

3. Metody wytwarzania oprogramowania (red. S. Szejko), MIKOM, Warszawa 2002.

4. . Pressman R., Software engineering: A Practitioner?s Approach, McGraw-Hill Co. Inc., 2004.

5. Pilone D., Pitman N., UML 2.0 almanach, Helion, Gliwice 2007.

Result of average student's workload

Activity	Time (working hours)
1. Participation in lectures	12
2. Participation in labs	8
3. Constuction of an object model, preparation to pass a test after the first part of software engineering course	20
4. Consultation, test	15

Student’s workload

Source of workload	hours	ECTS
Total workload	55	3
Contact hours	25	1
Practical activities	20	2