STUDY MODULE DESCRIPTION FORM

Name of the module/subject	Code
Software Development Studio 1	**1010512311010513906**

Field of study	Profile of study (general academic, practical)	Year /Semester
Computing	**general academic**	**1 / 1**
Elective path/specialty	Subject offered in:	Course (compulsory, elective)
Software Engineering	**English**	**obligatory**

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

No. of hours					No. of credits
Lecture: **-**	Classes: **-**	Laboratory: **-**	Project/seminars: **60**		**5**

Status of the course in the study program (Basic, major, other)	(university-wide, from another field)
major	**from field**

Education areas and fields of science and art	ECTS distribution (number and %)
technical sciences	**5 100%**
Technical sciences	**5 100%**

Responsible for subject / lecturer:

dr inż. Miroslaw Ochodek
email: Miroslaw.Ochodek@cs.put.poznan.pl
tel. 61 665 2944
Wydział Informatyki
ul. Piotrowo 3, 60-965 Poznań

Responsible for subject / lecturer:

mgr inż. Sylwia Kopczyńska
email: Sylwia.Kopczynska@cs.put.poznan.pl
tel. 61 665 2944
Wydział Informatyki
ul. Piotrowo 3, 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	Learning objectives concerning knowledge defined by the resolution of the PUT Academic Senate no. 42 from Apr 24 2017 that are verified in the admission process to the second cycle studies (i.e, K1st_W1 - 8).
2	**Skills**	Learning objectives concerning skills defined by the resolution of the PUT Academic Senate no. 42 from Apr 24 2017 that are verified in the admission process to the second cycle studies (i.e, K1st_U2 - 14).
3	**Social competencies**	Student shall understand necessity of continuous development of skills and show attitudes as honesty, responsibility, perseverance, curiosity, creativity, manners, and respect for other people.

Assumptions and objectives of the course:

- Provide basic knowledge, also by presenting real-life cases, regarding software engineering, and especially related to software project management and requirements engineering that is necessary to fulfill the management and analyst in a software development project,

- Developing skills related to developing software products (especially, skills related to project management, requirements engineering, and architecture) by participation in a capstone project, which goal is to solve a problem of real customers. The main focus is on developing skills related to the preparation of business case, project initiation, elicitation and documentation of requirements, and software architectures.

- Develop students? teamwork skills.

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

Knowledge:

1. has organized and well-formed theoretical general knowledge regarding agile software development methods (e.g., The Agile Manifesto, the most popular agile methods) - [K2st_W2]

2. has advanced and detailed knowledge regarding Scrum - [K2st_W3]

3. has advanced and detailed knowledge regarding planning software-product development (e.g., Product-Backlog management, the Kano model) - [K2st_W3]

4. has advanced and detailed knowledge about software-development process proposed by Scrum - [K2st_W5]

Skills:

1. is able to acquire knowledge necessary to conduct a software-development project (technical and domain knowledge) - [K2st_U1]
2. is able to use task-management tools (e.g., Jira) and customize them depending on project needs - [K2st_U2]
3. is able to integrate technical and domain knowledge to specify requirements for a software product - [K2st_U5]
4. is able to evaluate usefulness of technical solutions for the development of a software product - [K2st_U6]
5. is able to use Planning Poker to estimate effort - [K2st_U7]
6. is able to use reflection workshops to analyze and improve processes and ways-of-working in a software development project - [K2st_U8]
7. is able to design a software products according to presented requirements - [K2st_U11]
8. is able to collaborate within a software projects while fulfilling management and analytical roles - [K2st_U15]

Social competencies:

1. is aware of directions and intensity of modern software-development methodologies - [K2st_K1]
2. understands the necessity of continuous development of methods and ICT technologies and the necessity of adapting them to the context of a software-development project - [K2st_K2]

Assessment methods of study outcomes

Learning objectives are verified by:

Formative assessment:

- based on the regular assessment of the provided information regarding the activities performed within the project tasks
- based on the regular assessment of the current status of software development project tasks

Summative assessment:

Performed based on four criteria (the average percentage points):

- active participation in course classes (average grade is taken => 0-100%)
- knowledge test (mainly prepared based on the scrum.org certification tests) (average grade is taken => 0-100%)
- application of good practices (their selection depends on a project context) (0-100%)
- the quality of delivered products (min. business case and software requirements specification; other products selected based on the project context) (0-100%)

The final grade is determined using the following scale:

- (90%, 100%] ? 5.0
- (80%, 90%] ? 4.5
- (70%, 80%] ? 4.0
- (60%, 70%] ? 3.5
- (50%, 60%] ? 3.0
- (0%, 50%] ? 2.0

Course description

During the course a student takes part in a capstone software project aiming at solving the real problem defined by an external customer. The course covers the initial stages of a software development project and is continued in the following semester within the Software Development Studio 2 course.

The projects are run according to Scrum. A student can fulfill the Scrum Master, Product Owner or architect roles, while students of the first cycle studies in computer science play the roles of software developers.

The following aspects are covered within the course:

- agile software development methods with a special focus on Scrum (roles, artifacts, events)
- planning and managing product development (Product Backlog, the Kano model)
- applying Design Thinking to software development

More information about the teaching method used during the course could be found in the paper:

Kopczyńska, Sylwia, Jerzy Nawrocki, and Miroslaw Ochodek. Software development studio: bringing industrial environment to a classroom. Proceedings of the First International Workshop on Software Engineering Education Based on Real-World Experiences. IEEE Press, 2012.

Basic bibliography:

1. K. Schwaber, J. Sutherland, The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game, http://www.scrumguides.org, (available online), 2017

Additional bibliography:

1. Nawrocki, Jerzy, et al. Agile requirements engineering: A research perspective. International Conference on Current Trends in Theory and Practice of Informatics. Springer, Cham, 2014.

Result of average student's workload

Activity	Time (working hours)
1. participating in project classes	60
2. preparing to project classes	10
3. consulting issues related to the subject of the course; especially related to classes and projects	5
4. participating in project team meetings	15
5. project management	10
6. development of project documentation	20
7. studying literature / learning aids	5

Student's workload

Source of workload	hours	ECTS
Total workload	125	5
Contact hours	62	3
Practical activities	113	4