

STUDY MODULE DESCRIPTION FORM		
Name of the module/subject Software Development Studio 2		Code 1010512321010513908
Field of study Computing	Profile of study (general academic, practical) general academic	Year /Semester 1 / 2
Elective path/specialty Software Engineering	Subject offered in: Polish	Course (compulsory, elective) obligatory
Cycle of study: Second-cycle studies	Form of study (full-time, part-time) full-time	
No. of hours Lecture: - Classes: - Laboratory: - Project/seminars: 60		No. of credits 5
Status of the course in the study program (Basic, major, other) major		(university-wide, from another field) from field
Education areas and fields of science and art technical sciences Technical sciences		ECTS distribution (number and %) 5 100% 5 100%
Responsible for subject / lecturer: dr inż. Mirosław Ochodek email: Mirosław.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	Student shall have a general knowledge regarding software engineering and detailed knowledge regarding project management with Scrum.
2	Skills	Student shall have skills related to project management with Scrum (especially, Product-Backlog management, sprint and release planning, organizing and conducting event meetings defined by Scrum).
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:		
<ul style="list-style-type: none"> - 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 managing a software development team, requirements engineering (maintaining Product Backlog), and software architectures. - Develop students' teamwork skills. - This module extends the Software Development Studio 1 module. 		
Study outcomes and reference to the educational results for a field of study		
Knowledge:		
<ol style="list-style-type: none"> 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 about software-development process proposed by Scrum - [K2st_W5] 		
Skills:		

1. is able to acquire knowledge necessary to conduct a software-development project (e.g., knowledge related to building effective teams) - [K2st_U1]
2. is able to use task-management tools (e.g., Jira) to organize the work of a project team - [K2st_U2]
3. is able to integrate technical and domain knowledge while talking with users and/or customers to orientate future development of 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 select appropriate software effort estimation methods to plan scopes of releases and sprints (e.g., Planning Poker) - [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 evaluate code quality and propose improvements - [K2st_U8]
8. is able to design a software products according to presented requirements - [K2st_U11]
9. 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 three criteria (the weighted average percentage points):

- active participation in course classes (average grade is taken => 0-100%, weight 0.1)
- knowledge test (mainly prepared based on the scrum.org certification tests) (average grade is taken => 0-100%, weight 0.3)
- project development (the average from weekly assessments => 0-100%, weight 0.6)

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

This module extends the Software Development Studio 1 (SDS 1) module and aims at further development of the capstone software project (started within the SDS 1) aiming at solving the real problem defined by an external customer. The projects is 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:

- team roles and responsibilities
- project management and stage management (planning releases, task delegation, change management, planning and reviewing releases)
- monitoring the progress of a project and fact-based decision making,
- software quality assurance (acceptance testing)
- software architecture (maintenance of software architecture),
- reflection workshops,
- software delivery (formal product acceptance, release preparation),
- risk management (identification, analysis, and responding).

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

Kopczyńska, Sylwia, Jerzy Nawrocki, and Mirosław 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	5	
3. consulting issues related to the subject of the course; especially related to classes and projects	5	
4. participating in project team meetings	20	
5. project management	20	
6. development and update of project documentation	10	
7. studying literature / learning aids	5	
Student's workload		
Source of workload	hours	ECTS
Total workload	125	5
Contact hours	62	3
Practical activities	110	4