STUDY MODULE DESCRIPTION FORM

Name of the module/subject **Project management**		Code **1010512311010510631**
Field of study **Computing**	Profile of study (general academic, practical) **general academic**	Year /Semester **1 / 1**
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				No. of credits **4**
Lecture: **20**	Classes: **10**	Laboratory: **30**	Project/seminars: **-**	

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

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 hab. inż. Jerzy Nawrocki, prof. nadzw. email: jerzy.nawrocki@put.poznan.pl tel. 665-2980 Wydział Informatyki ul. Piotrowo 3 60-965 Poznań	Responsible for subject / lecturer: dr hab. inż. Jerzy Nawrocki, prof. nadzw. email: jerzy.nawrocki@put.poznan.pl tel. 665-2980 Wydział Informatyki ul. Piotrowo 3 60-965 Poznań

Prerequisites in terms of knowledge, skills and social competencies:

1	**Knowledge**	Knows at least one programming language.
2	**Skills**	Is able to write and run simple programs in the selected programming language.
3	**Social competencies**	Is ready to expand his knowledge, skills and social competences in the field of project management.

Assumptions and objectives of the course:

The aim of the course is to present the basic principles and practices related to the effective management of IT / software development projects implementing products and services.

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

Knowledge:

1. Has advance and detailed knowledge regarding how to formulate the goal of an IT / software development project, functional and nonfunctional requirements, effort estimation methods, and agile software development. - [K2s_W3]

2. Knows software development cycle and team structure of different software development methodologies. - [K2s_W5]

3. Knows economic and social conditions of IT/ software development projects, including related risk management approaches. - [K2s_W8]

4. Knows psychologic conditions of IT / software development projects, including the basic motivation theories. - [K2s_W8]

Skills:

1. Is able to point out information and communication techniques useful for managing an IT project. - [K2s_U2]

2. Is able to discuss economic, social and psychological aspects of an IT project. - [K2s_U5]

3. Is able to indicated the method of effort estimation that is suited for a given IT project. - [K2s_U7]

4. Is able to assess the suitability of methods and tools for software quality control (manual tests, automated tests, code reviews, formal methods) to a given context of an IT project. - [K2s_U9]

5. Can present a requirements specification in the form of use cases and can point out architectural style suitable for a given IT project. - [K2s_U11]

6. Is able to present the basic cooperation techniques and practices in a small team, including Scrum. - [K2st_U15]

Social competencies:

1. Understands that knowledge and skills quickly become old-fashioned in IT and it also applies to software development methodologies. - [K2st_K1]

2. Understands the significance of the most up-to-date knowledge regarding project management. - [K2st_K2]

Assessment methods of study outcomes

Achieving learning outcomes is verified based on an during lectures (through an interactive form), during laboratories through tasks and tests, and during exercises through tests (there is no exam).

Formative assessment:

-          Each test is graded on the scale of 0 to 100% of correct answers.

-          Each set of tasks is rated on a scale of 0-100 points.

Summative assessment:

1. For lectures and exercises:

The average of the grades from the tests is the basis for the final grade. Mapping of the average grade from the tests, T, to the final grade is made according to the following rule:

  90% ? T   : grade 5.0 (A)

  80% ? T < 90%: ocena 4.5 (B)

  70% ? T < 80%: grade 4.0 (C)

  60% ? T < 70%: grade 3.5 (D)

  50% ? T < 60%: grade 3.0 (E)

      T < 50%: grade 2.0 (F)

2. For laboratories:

  Calculated on the basis of the total number of points for:

  A) (40 points) tests

  B) (60 points) tasks performed on laboratories and homework

  The mapping of the total number of points to the final grade is made according to the following rule:

  90% < T   : grade 5.0 (A)

  80% < T ? 90%: ocena 4.5 (B)

  70% < T ? 80%: grade 4.0 (C)

  60% < T ? 70%: grade 3.5 (D)

  50% < T ? 60%: grade 3.0 (E)

      T ? 50%: grade 2.0 (F)

Course description

Lectures cover the following topics:

Review of project management methodologies. Risk management. Communication and issue management. The goal of an IT / software development project. Outline of the project (Project Brief). Functional requirements and use cases. Non-functional requirements. Software architecture. Software quality control methods. User Interface. Software testing. Effort estimation and project planning. Change management. Man in an IT project

The laboratories and exercises serve to consolidate the knowledge presented at lectures.

Basic bibliography:

1. Ken Schwaber and Jeff Sutherland, The Scrum Guide, 2017,   https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf

2. Stephen Covey, 7 habits of highly effective people

Additional bibliography:

1. Managing Successful Projects with PRINCE2, OGC

2. Jerzy Nawrocki et al., Balancing agility and discipline with XPrince, LNCS 3943, 266-277, 2006

3. Jerzy Nawrocki et al., Agile requirements engineering: A research perspective (invited lecture), LNCS 8327,   40-51, 2014

Result of average student's workload

Activity	Time (working hours)

1. Participation in lectures	20
2. Participation in excercies	10
3. Participation in laboratories	30
4. Preparation to classes	20
5. Literatue studies and own work	30

Student's workload		
Source of workload	**hours**	**ECTS**
Total workload	110	4
Contact hours	60	2
Practical activities	40	1