

<b>STUDY MODULE DESCRIPTION FORM</b>		
Name of the module/subject <b>Software Development Studio 2</b>		Code <b>1010512321010513908</b>
Field of study <b>Computing</b>	Profile of study (general academic, practical) <b>general academic</b>	Year /Semester <b>1 / 2</b>
Elective path/specialty <b>Software Engineering</b>	Subject offered in: <b>English</b>	Course (compulsory, elective) <b>obligatory</b>
Cycle of study: <b>Second-cycle studies</b>	Form of study (full-time, part-time) <b>full-time</b>	
No. of hours Lecture: - Classes: - Laboratory: - Project/seminars: <b>60</b>		No. of credits <b>4</b>
Status of the course in the study program (Basic, major, other) <b>major</b>		(university-wide, from another field) <b>from field</b>
Education areas and fields of science and art <b>technical sciences</b>		ECTS distribution (number and %) <b>4 100%</b>
<b>Responsible for subject / lecturer:</b> Miroslaw Ochodek, PhD email: Miroslaw.Ochodek@put.poznan.pl tel. 61 6652944 Institute of Computing Science Piotrowo 2 Str., 60-965 Poznan		
<b>Prerequisites in terms of knowledge, skills and social competencies:</b>		
1	<b>Knowledge</b>	Student starting this module should have a basic knowledge regarding software engineering especially regarding software project management (K_W4), including a basic knowledge of software development project lifecycle (K_W7).
2	<b>Skills</b>	Should have skills allowing solving basic problems related to analyzing risk (K_U11), functional test planning and execution (K_20), should be able to prepare a requirements specification document for a software product (K_U22-23), and have skills that are necessary to acquire information from given sources of information.
3	<b>Social competencies</b>	Student should understand the need to extend his/her competences / has the willingness to work in a team. In addition, in respect to the social skills the student should show attitudes as honesty, responsibility, perseverance, curiosity, creativity, manners, and respect for other people.
<b>Assumptions and objectives of the course:</b> 1. Provide knowledge, supported with the presentation of real cases, regarding software engineering related to software project management, requirements engineering, software architecture, which is necessary to fulfill the roles of: project manager, analyst, architect., 2. Develop student's skills in solving problems related to software project management, requirements engineering, software architecture by involving students in a software project developed for a real customer. Within the course, the main focus is on developing student's skills related to planning and management of construction stages (so called releases), closing-up a project, and transition of project outcomes. 3. Develop student's teamwork skills.		
<b>Study outcomes and reference to the educational results for a field of study</b>		
<b>Knowledge:</b> 1. has well-established theoretical knowledge of software engineering regarding project management, functional and non-functional requirements analysis, software architecture, acceptance testing, software system transition, etc. - [K_W4] 2. has detailed theoretical knowledge related to the selected areas of computer science including: software project management according to PRINCE2, eXtreme Programming, and XPrince; functional requirements specification with the use of use cases; organizing and moderating architecture analysis workshops, and designing acceptance tests for software systems. - [K_W5] 3. has basic knowledge regarding life-cycle of software systems, especially regarding the life-cycle of the software development stages. - [K_W7] 4. knows the fundamental methods, techniques and tools employed to solve complex engineering tasks related to management and planning of software projects, requirements elicitation, GUI design, software and process measurement, and acceptance testing. - [K_W8]		
<b>Skills:</b>		

<p>1. is able to acquire, combine, interpret and evaluate information from literature, databases and other information sources (in mother tongue and English); draw conclusions, and formulate opinions based on it. - [K_U1]</p> <p>2. is able to: participate and moderate software project meeting (in mother tongue and English) regarding various aspects of software development, present assumptions, constraints, current status and the results of a software project, negotiate with a client, use software tools supporting communication among team members (e.g., Redmine). - [K_U2]</p> <p>3. is able to plan and arrange self-education process especially regarding new methods, techniques, and technologies used in software development. - [K_U5]</p> <p>4. is able to understand the content of books and papers in English, can play the role of a moderator during a project meeting in English, can present the assumptions, the results of a project and negotiate with a client in English. - [K_U6]</p> <p>5. is able to use Information and Communication Technologies that are commonly employed in IT projects when participating in a software project. - [K_U7]</p> <p>6. is able to combine knowledge from different areas of computer science (and if necessary from other scientific disciplines) to formulate and solve engineering tasks; and use system approach that also incorporates nontechnical aspects. - [K_U10]</p> <p>7. is able to conduct risk analysis of an IT project and manage the risk regarding the software project he/she participate in. - [K_U11]</p> <p>8. is able to assess usefulness and possibility of employing new developments (methods and tools) and new IT products. - [K_U13]</p> <p>9. is able to develop an object-oriented model of a simple software system (e.g., in UML notation). - [K_U17]</p> <p>10. is able to systematically design, plan and execute acceptance tests regarding functional and non-functional aspects of a software product. - [K_U20]</p> <p>11. is able to propose enhancements (improvements) to existing technical solutions. - [K_U21]</p> <p>12. is able to evaluate usefulness of methods and tools (also to identify their limitations) used to solve engineering tasks, i.e., building IT systems or their components. - [K_U24]</p> <p>13. is able to design (according to a provided specification which includes also non-technical aspects) a complex device, an IT system, or a process; and is able implement it (at least partially) using appropriate methods, techniques, and tools (including adjustment of available tools or developing new ones). - [K_U27]</p>
<p><b>Social competencies:</b></p> <p>1. is able to collaborate and cooperate in a team performing different roles including the roles of project manager, analyst, architect. - [K_K5]</p> <p>2. is able to correctly assign priorities to own tasks and tasks performed by others. - [K_K6]</p> <p>3. is able to think and act in an entrepreneurial way. - [K_K8]</p>

<p><b>Assessment methods of study outcomes</b></p>
<p>Formative assessment:</p> <ul style="list-style-type: none"> <li>- based on the regular assessment of the provided information regarding the activities performed within the project tasks,</li> <li>- based on the regular assessment of the current status of software development project tasks,</li> <li>- based on the regular assessment of the conduction and organization of software project meetings (meetings with team members, the representatives of client, user, supplier);</li> <li>- based on the regular assessment of the project documentation.</li> </ul> <p>Summative assessment:</p> <ul style="list-style-type: none"> <li>- based on the assessment of the student's preparation to the project classes (multiple choice question test at the beginning of each class) - K_W4, K_W5, K_W7, K_W8, K_U1, K_U6;</li> <li>- based on the assessment of the quality of the created project documentation (management and planning documents, requirements specification documents, design documents of software architecture, test specifications, and other technical documentation) - K_W4, K_W5, K_W7, K_W8, K_U1, K_U5, K_U6, K_U10, K_U13, K_U17, K_U21, K_U24, K_U27;</li> <li>- based on the assessment of project planning and following a project schedule - K_U27, K_K6, K_K8;</li> <li>- based on the assessment of applying good software engineering practices in a software project the student participates in, including the responsibilities assigned to the roles performed by the student - K_W4, K_W5, K_W7, K_W8, K_U2, K_U5, K_U7, K_U18, K_U20, K_5;</li> <li>- based on the assessment of the effectiveness of communication among team members based on the task log in a task management tool - K_U2, K_U7;</li> <li>- based on the assessment of the quality of risk log and the effectiveness of risk mitigation actions - K_U11;</li> <li>- based on the assessment of the quality of development team project meetings and meetings with the representatives of customer, user, and supplier - K_U2, K_U6, K_U7;</li> <li>- based on the assessment of the completeness and quality of the software system developed during a project based on the acceptance tests and the opinion of the representatives of customer, user, and supplier - K_U20, K_U27.</li> </ul>
<p><b>Course description</b></p>

The course is the continuation of the Software Development 1 course given in the previous semester and is about further development of a software project for a real customer.

The projects are run according to the guidelines of XPrince project management methodology which is based on three well-known methodologies: PRINCE2, Rational Unified Process, and eXtreme Programming. Each student can play one or more of the following project roles: project manager, analyst, and architect. The students of first cycle in computer science play roles of software developers.

The following project stages are covered within the scope of the course: software architecture refinement, software development in releases (each release is divided into two iterations and a transition of a working version of a system), and closing up a project.

The following aspects are covered within the course:

- Team structure (steering committee, responsibilities of the roles of project manager, analyst, architect);
- Software project life cycle (XPrince, PRINCE2, eXtreme Programming - initial stages of a project and the tasks that should be completed in each stage);
- Project management, controlling a stage (planning a release with the use of planning game, task delegation in a team, change management, project plan development);
- Monitoring the project progress and decision making based on facts (measurement programs and systems, Goal-Question-Metric - GQM);
- Elicitation and analysis of requirements (use cases, non-functional requirements, GUI workshops, requirements elicitation workshops);
- Quality assurance (acceptance tests regarding functional and non-functional aspects of a system, measurement program);
- Software architecture (refinement of architecture designs as a response to the changes of requirements, technologies of software development);
- Reflection workshops;
- Transition of a software system (formal acceptance meetings, preparation to transit software system to maintenance organization units),
- Risk management (identification, analysis and mitigation);

Learning methods:

1. multimedia presentation, multimedia showcase, discussion, teamwork, workshop, case studies, tutorial.

**Basic bibliography:**

1. Office of Government Commerce, An Introduction to PRINCE2: Managing and Directing Successful Projects, The Stationary Office, ISBN-13: 9780113311880, 2009
2. Writing Effective Use Cases, A. Cockburn, Addison-Wesley, Boston, 2001

**Additional bibliography:**

1. Software Requirements 2nd edition, K. E. Wiegers, Microsoft Press, Redmond, WA, USA, 2003
2. 7 Habits of Highly Effective People, S. Covey, Free Press, London, 2004
3. Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice (2nd Edition), ISBN-13: 978-0321154958, Addison-Wesley Professional, 2003

**Result of average student's workload**

Activity	Time (working hours)
1. participating in project classes: 15 x 4 hours,	60
2. preparing to project classes and development of project documentation (15 x 1 hour),	15
3. consulting issues related to the subject of the course; especially related to classes and projects,	4
4. participating in project team meetings (4 x 2 hours),	8
5. participating in meetings with steering committee (4 x 2 hours),	8
6. project management (10 x 1 hour),	10
7. studying literature / learning aids (10 pages = 1 hour), 100 pages.	10

**Student's workload**

Source of workload	hours	ECTS
Total workload	115	4
Contact hours	80	3
Practical activities	105	4