LINGI2255

2016-2017

Software engineering project

6.0 credits	30.0 h + 30.0 h	1q

Teacher(s) :	Mens Kim ;
Language :	Anglais
Place of the course	Louvain-la-Neuve
Inline resources:	[> http://moodleucl.uclouvain.be/course/view.php?id=7599](http://moodleucl.uclouvain.be/course/view.php?id=7599)
Main themes :	In general, the best way for students to learn about software engineering is by participating in a project. Such a project should require students to work on a team to develop a software system through as much of its lifecycle as is possible. Much of software engineering is devoted to effective communication among team members and stakeholders. Utilizing project teams, projects can be sufficiently challenging to require students to use effective software engineering techniques and to develop and practice their communication skills. The best way to learn to apply software engineering theory and knowledge is in the practical environment of a project. The practical project will be accompanied by a few theoretical sessions where the students will become acquainted with: -- Software process models (e.g., waterfall, incremental, agile) and the activities within software lifecycles; -- Practical requirements engineering approaches; -- Software analysis and design techniques and notations; -- Describing system data using, for example, object-role modelling or entity-relationship diagrams; -- Project management and planning techniques.
Aims :	Given the learning outcomes of the "Master in Computer Science and Engineering" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: -- INFO2.1-4 -- INFO4.1-4 -- INFO5.1-3 -- INFO6.2-4 Given the learning outcomes of the "Master [120] in Computer Science" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: -- SINF1.M3 -- SINF2.1-4 -- SINF4.1-4 -- SINF5.1-3 -- SINF6.2-4 Given the learning outcomes of the "Master [60] in Computer Science" program, this course contributes to the development, acquisition and evaluation of the following learning outcomes: -- 1SINF1.M3 -- 1SINF2.1-4 -- 1SINF3.1-4 -- 1SINF4.1-3 -- 1SINF5.2-4 At the outcome of this course, the students will have acquired the necessary competences to build a large-scale software system under semi-professional working conditions. More specifically, students having completing this course with success will be able to: --

	Describe the differences among several major process models (e.g., waterfall, iterative, and agile); -- Differentiate among the phases of software development (specification, architecture, design, implementation, validation, documentation); -- Complete, in a rigorous and systematic way, the artefacts produced in these different software life cycle phases; -- Apply a software development methodology currently practiced in industry; -- Work efficiently in a team to develop a medium-to large-scale software system; -- Manage the coordination and communication between the different team members; -- Interact with a client to identify his requirements, to clarify imprecise specifications, and to take into account requested modifications throughout the development process; -- Describe the functional requirements of a software system using, for example, use cases or users stories;' -- Estimate the time and resources needed to complete such a software development project, plan the tasks to be executed and the deliverables to be produced, and respect this planning; -- Use some project management tool to assign and follow the planned software development tasks; -- Put in practice different methods and techniques to assure the quality of the produced software; -- Understand the problems inherent to the development of large software systems having different stakeholders and that consist of multiple components. *The contribution of this Teaching Unit to the development and command of the skills and learning outcomes of the programme(s) can be accessed at the end of this sheet, in the section entitled “Programmes/courses offering this Teaching Unit”.*
Evaluation methods :	The course evaluation will be based on : -- Individual participation to weekly group meetings with the course tutors; -- Three intermediate reports; -- The final report, delivered system and documentation, presentation and demonstration of the final product; -- A small exam consisting of a few more theoretical exam questions. Given that this course is based on the participation in a team project throughout the year, there will be no possibility to do a second session for this course.
Content :	This project will consist of the development (analysis, design, implementation, validation and documentation) of a realistic and non-trivial software application, if possible proposed by and with the participation of an actual client, in conditions of semi-professional work. Teams of 6 to 8 students (necessary to complete a big project), will collaborate, overseen by a project manager. Weekly meetings will be held with the project leader (a teaching assistant or student-tutor) to present the progress and difficulties encountered, to assess alternative options, and to discuss the distribution and planning of work within the team. The proposed software development methodology will be structured according to a waterfall-like model, mainly to be able to explore the different phases of the software development life-cycle (after all, this is a software engineering project), and to facilitate the management of the different groups by their tutors. The application to be developed will most likely be a web-application, but the choice of programming language, environment, application framework and development tools will mostly be left open to the students (for as far as it fits the requirements imposed by the client).
Bibliography :	Suggested additional reading will be given in the course syllabus which describes the project's deliverables and organisation. This syllabus, as well as other relevant course material, slides and practical information will be accessible on Moodle, which will also be the primary means of communication between the teacher(s) and the students.
Other infos :	Background: -- Having a good knowledge of and experience with object-oriented programming concepts, algorithms and data structures. -- Having participated in the development of a small- to medium-scale software system.
Faculty or entity in charge:	INFO

Programmes / formations proposant cette unité d'enseignement (UE)

Intitulé du programme	Sigle	Credits	Prerequis	Acquis d'apprentissage
Master [120] in Computer Science and Engineering	INFO2M	6	-	🔍
Master [60] in Computer Science	SINF2M1	6	-	🔍
Master [120] in Computer Science	SINF2M	6	-	🔍