lingi2255

2019

UCLouvain

Software engineering project

In view of the health context linked to the spread of the coronavirus, the methods of organisation and evaluation of the learning units could be adapted in different situations; these possible new methods have been - or will be - communicated by the teachers to the students.

6 credits	30.0 h + 30.0 h	Q1

Teacher(s)	Mens Kim ;
Language :	English
Place of the course	Louvain-la-Neuve
Main themes	In general, the best way for students to learn about software engineering is by participating in a large project. In such a project, representative of a realistic industrial project, students will be required to work in teams to develop a software system throughout its entire life cycle. An important part of software engineering is dedicated to an efficient communication and collaboration between team members and external actors such as the client and end user. By working in larger teams, the projects can be of sufficient difficult to force students to develop and apply their technical and communication skills. The best way to learn how to apply software engineering theory and knowledge is in the practical environment of a realistic development project. This practical project will be accompanied by a few theoretical sessions where the students will get acquainted with: • software development models (waterfall, incremental, agile) and the software lifecycle activities; • practical approaches to requirements engineering; • analysis and design of technical software and the corresponding modelling notations; • describing the data of a software system using, for example, object role modelling or entity-relationship diagrams; • project management and planning techniques.
Aims	At the end of this learning unit, the student is able to: 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: • INFO2.1-5 • INFO4.1-4 • INFO5.1-6 • INFO6.1, INFO6.4 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: • INFO2.1-4 • INFO4.1-4 • INFO5.1-3 • INFO6.2-4 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 different phases of the software development life cycle (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	**Due to the COVID-19 crisis, the information in this section is particularly likely to change.** Course evaluation will be based on: • Individual participation in group work and weekly group meetings with an assistant or tutor, who plays the role of project manager; • The realization of three intermediate prototypes with corresponding technical reports; • The final report, the delivered system and its documentation, as well as a presentation and demonstration of the final product to the customer. **Warning:** Since this course is based on your participation in a team project throughout the entire year, there will be no opportunity to do a second session for this course.
Teaching methods	**Due to the COVID-19 crisis, the information in this section is particularly likely to change.** Development (analysis, design, implementation, validation, documentation, integration and deployment) of a large software system for a client, in teams of 6 to 8 students supervised by a project manager. Weekly meetings will be held with the project manager, and different prototypes and reports will need to be produced throughout the project.
Content	This software engineering project consists of the development (analysis, design, implementation, validation, documentation, integration and deployment) of a realistic and non-trivial software application, if possible proposed by and with the participation of a real client, under semi-professional working conditions. The topic of the application to be constructed is proposed by an industrial partner or a non-profit organization that participates in the organisation of this course. Teams of 6 to 8 students (required to achieve such a large project), will collaborate, supervised by a project manager. Weekly meetings will be held with the project manager (an assistant or tutor) to present the progress and difficulties encountered, to evaluate alternatives, and discuss the distribution and planning of the work among team members. The application to be developed will most likely be a web application, but the choice of the programming language, the environment, the application framework, and the development tools will depend on the requirements of the project client.
Inline resources	*http://moodleucl.uclouvain.be/course/view.php?id=7599*
Bibliography	**French** Des lectures supple#mentaires seront sugge#re#es dans le plan de cours qui de#crit les produits livrables et l'organisation du projet. Les supports de cours pertinents, des slides et des informations pratiques seront accessibles sur Moodle, qui sera e#galement le principal moyen de communication entre l'enseignant et les e#tudiants. **English** Additional reading material will be suggested in the course plan which describes the deliverables and organisation of the project. All relevant course material, slides and practical information will be available on Moodle, which will also be the main means of communication between the teacher and the students.
Other infos	Prerequisites: • Have good knowledge of and experience with the concepts of object-oriented programming, algorithms and data structures. • Have participated in the development of a small to medium-sized software system.
Faculty or entity in charge	INFO

Programmes containing this learning unit (UE)

Program title	Acronym	Credits	Prerequisite	Aims
Master [120] in Computer Science and Engineering	INFO2M	6		🔍
Master [60] in Computer Science	SINF2M1	6		🔍
Master [120] in Computer Science	SINF2M	6		🔍