UCL Université catholique de Louvain

LINGI2251

2012-2013

Software engineering: development methods

5.0 credits	30.0 h + 30.0 h	1q

Teacher(s) :	Van Lamsweerde Axel ;
Language :	Anglais
Place of the course	Louvain-la-Neuve
Prerequisites :	-- mathematical logic as provided by the course INGI1101 -- master of object-oriented programming, algorithms and data structures as provided by the SINF1121 -- participating in the implementation of a small-size software project (for example SINF1124)
Main themes :	-- The software lifecycle: products and processes. -- Introduction to requirements engineering: eliciting, modeling, specifying, analysing, and documenting software requirements. -- Introduction to architectural design: logical vs. physical architecture; hierarchical structuring, modularisation; styles and architectural patterns. -- Specifying modules as work units. Formal specification. -- Test case design for black-box, white-box, and integration testing. -- Documenting decisions at each development step.
Aims :	Students completing successfully this course will be able to -- describe main issues raised by large software projects -- discuss the various qualities a large-scale complex software application should exhibit -- model complex software-intensive systems (model-drive software engineering) -- master systematic approaches to software development -- describe the various types of product and process involved along the software engineering lifecycle -- apply typical approaches for engineering software requirements and for modeling, specifying, designing, validating, and documenting high-quality software -- assess the impact of the decisions made at the various steps of this lifecycle. Students will have developed skills and operational methodology. In particular, they have developed their abilities to -- model rigorously the final product in order to make a rigorous analysis from the design that will ensure the quality of the product and its development process, -- explain how to take into account when designing a product of the volume of resources (particularly human) that will be involved in its development. *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 :	1. Individual quizzes, on mini-project, after each course milestone (30% ingi2255) -- to help you get prepared before meeting project leader -- for auto-evaluation 2. Individual participation during weekly group meetings with project leader(30% ingi2255/sinf2255) (knowledge of required techniques, degree of interaction, initiatives, ideas, ...) 3. Quality of deliverables: project reports & mp; documentation (40% ingi2255/sinf2255) 4. Written exam (January) -- INGI 2251: theory and case study (100% ingi2251)
Teaching methods :	The course is strongly coupled with the development of a large-scale project, by teams, according to the techniques studied (see INGI2255 : Software Engineering Project). It is organized intensively during the very first weeks of the quadrimester to allow the project to start promptly. Subsequent teaching sessions are organized episodically as the various development steps require them. Quizzes are organized regularly to check whether each individual student in a team is working properly. Case study sessions are also scheduled to show how the techniques can be used in another, smaller-size project. The course exercises (2 ECTS) are fully integrated in INGI2255 (Software Engineering Project, 5 ECTS). This introduction to software engineering thus corresponds to a global load of 10 ECTS, that is, third-time work throughout the whole quadrimester.
Content :	-- Software Engineering: WHAT, WHY, HOW -- The software lifecycle -- Lifecycle statics -- Lifecycle dynamics -- Estimating Costs and Schedules -- Specification in the software lifecycle -- Requirements Engineering -- Model-driven elaboration of the Requirement Document -- Specifying software requirements -- Architectural design

	-- Hierarchical software structuring -- Modularisation -- Architectural styles -- Software testing -- Black-box, white-box, integration testing
Bibliography :	-- slides on line -- A. van Lamsweerde, Requirements Engineering, Wiley, 2009 -- F. Brooks, The Mythical Man-Month, Addison-Wesley, 1995 -- S. Pfleeger, Software Engineering: Theory and Practice, 2nd Edition, Prentice-Hall, 2001
Cycle and year of study :	> Master [60] in Computer Science > Master [120] in Biomedical Engineering > Master [120] in Computer Science and Engineering > Master [120] in Computer Science
Faculty or entity in charge:	INFO