LINGI2251

2015-2016

Software engineering: development methods

5.0 credits	30.0 h + 30.0 h	2q

Teacher(s) :	Pecheur Charles ;
Language :	Anglais
Place of the course	Louvain-la-Neuve
Inline resources:	> http://icampus.uclouvain.be/claroline/course/index.php?cid=LINGI2251
Main themes :	-- The what, why and how of software engineering; -- Software life-cycle phases (introduction to requirements engineering, -- Software architecture and design, software construction, software -- verification and validation, deployment, software evolution and maintenance); -- Software process models (e.g., waterfall, incremental, agile); -- Software process capability maturity models; -- Software quality concepts and quality assurance; -- Software modelling (requirement models, functional, architectural and behavioural models); -- Program specification and correctness. -- Software verification and validation, testing.
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: -- INFO1.1-3 -- INFO2.1-5 -- INFO6.1-3 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 -- SINF2.1-5 -- SINF6.1-3 Students completing successfully this course will be able to: -- describe main issues raised by large software projects; -- master systematic approaches to software development; -- explain the concept of a software life-cycle and provide an example, illustrating its phases, including the various products that are produced; -- assess the impact of the decisions made at various phases of this lifecycle; -- compare common software process models with respect to their value for developing particular classes of software systems; -- explain the purpose of process maturity models and determine the maturity of a given process; -- describe fundamental challenges of and common techniques used for requirements engineering and apply them for a simple software system; --

	define software quality, discuss the various qualities that a large-scale complex software application should exhibit and describe the role of quality assurance activities in the software process; -- model rigorously the design of a product in order to ensure the quality of the final product and its development process; -- for a simple software system, propose an appropriate software architecture and discuss its advantages and disadvantages; -- specify program components and use these specifications to verify their correct behaviour. -- apply a variety of testing and other verification techniques to verify program correctness. *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 :	-- Asssignments and reports, 40% of the final grade. -- Theory: oral exam, 60% of the final grade. A list of questions is provided at the end of the course. The asssignments and reports can only be presented during the quadrimester of the course. They cannot be represented in subsequent exam sessions.
Teaching methods :	The course will consist of lectures and lab sessions. One or two lectures will consist of presentations by industry specialists. In the lab sessions, students will apply software engineering techniques and tools to sample software applications. During the quadrimester, students will have to complete several assignments and/or quizzes, applying the tools and techniques seen in the lab sessions.
Content :	Software engineering is the discipline concerned with the application of theory, knowledge, and practice to effectively and efficiently build reliable software systems that satisfy the requirements of customers and users. This discipline is applicable to small, medium, and large-scale systems. It encompasses all phases of the lifecycle of a software system, including requirements elicitation, analysis and specification; design; construction; verification and validation; deployment; and operation and maintenance. Whether small or large, following a traditional disciplined development process, an agile approach, or some other method, software engineering is concerned with the best way to build good software systems.
Bibliography :	Shari Lawrence Pfleeger and Joanne Atlee, Software Engineering: Theory and Practice, 4th edition, Pearson, 2010. Lectures slides will be available on the course website.
Other infos :	Background: -- LINGI1101 : mathematical logic -- LSINF1121 : master of object-oriented programming, algorithms and data structures -- LSINF1225 : participating in the implementation of a small-size software project
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 Biomedical Engineering	GBIO2M	5	-	
Master [120] in Computer Science and Engineering	INFO2M	5	-	
Master [120] in Computer Science	SINF2M	5	-	