Università degli Studi di L'Aquila - Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Course catalogue

Page compiled on 26/10/2020

Programme of Course "Analysis and Testing of Component-based Systems"

- Code: F0175
- Type of course unit: Compulsory (Master Degree in Computer Science curriculum General)
- Level of course unit: Postgraduate Degrees
- Semester: 1

Number of ects credits: (Master Degree in Computer Science) 6 (workload 150 hours)

Teachers: Henry Muccini (henry.muccini@di.univaq.it)

1	**Course objectives**	KNOWLEDGE: This course introduces advanced concepts on Software Architecture. The first part of this course will provide advanced basic and advanced knowledge on software architecture, together with examples, and with a specific focus on architecture description language, and multi-view modeling. The second part will focus on architectural design decisions, architectural patterns, and architecture for adaptive systems. An objective is also to gain familiarity with software languages and tools which make easier the specification of component-based systems and architectures ABILITY (ability to do): From the perspective of the "ability students will gain", the main objective of this course is to acquire a good knowledge on both theory and practice of Software Architecture and their usage in practice. At the end of this course, students will be able to correctly model a Software Architecture by using the appropriate tools.Through projects, students will practice the theoretical concepts previously described. BEHAVIOR (ability to be): at the end of the learning process, the students will be conscious of how architectural choices impact on the quality of the developed software system.
2	**Course content and learning outcomes (dublin descriptors)**	Topics of the module include: • Software Architecture • Architecture Description Languages • IEEE/IEC 42010 Standard and multi-view modeling • Architectural Styles • Architectural Design Decisions • Tool Support On successful completion of this module, the student should : • o gain familiarity with software tools which make easier the specification of software architectures; o acquire a good knowledge on both theory and practice of Software Architecture and their usage; o be able to correctly model a Software Architecture with the appropriate tools; o be able to apply the theoretical concepts to concrete projects; o be conscious of how architectural choices impact the quality of the developed software system; o demonstrate the ability to analyze, plan, and produce a complete software application. o be able to articulate the software system development life cycle and its characteristics. o be able to analyze needs and constraints of complex software system design. o be able to write a project proposal, a project report, and an operating manual. o be able to collaborate with team members to implement a large software system. o understand and apply modern software production practice and tools such as component based programming, integrated development environment, and version control.
3	**Course prerequisites**	Students should have attended the "Ingegneria del Software I" course.
4	**Teaching methodsand language**	Lectures, Classroom discussion Mandatory readings Language: English Ref. Textbooks: -Vari, Research papers provided by the lecturer. -Len Bass, and Paul Clements, and Rick Kazman, Software Architecture in Practice (2nd Edition). Addison-Wesley Professional; 2 edition. April 2003. **Language**: English **Reference textbooks** • Vari, ***Research papers provided by the lecturer***. • Len Bass, and Paul Clements, and Rick Kazman, ***Software Architecture in Practice***

		(2nd Edition). Addison-Wesley Professional; 2 edition. April 2003.
5	**Assessment methods**	Pre-Assessment: There is no formal pre-assessment. Still, the above mentioned pre-requisites are fundamental and their fulfillment is verified throughout questions and discussions in class. Additional lectures or short seminars are provided by the teacher in case significant problems are detected. Formative Assessment: The formative assessment is performed via interactive interaction between teacher and students during lectures. Students are aware since the beginning of the Course that they will be involved (in turns) in: - Questioning and discussion, by means of open oral questions to the class or to single students. The all students are required to analyze a given problem, and provide their own solutions. - Short seminars: students may be assigned personalized homework, that they will have to illustrate to the class by means of 30 minutes' short seminars. Summative Assessment that consists of a written project, followed by an oral exam. The project is aimed at: (1) verification of theoretical competences, and in particular of knowledge and comprehension of Course contents (2) verification of skills in understanding and solving significant exercises, and in explaining the proposed solutions. This in order to verify the ability of application of techniques learnt during the Course, of analysis of problems and synthesis of suitable solutions, and of evaluation of alternative solutions. Criteria of evaluation will be: the level of knowledge and practical ability; the property of use of the technical/mathematical language; the clarity and completeness of explanations. The oral exam will occur within one week after the project submission and will typically cover the areas of the project that need clarification plus, additional subjects not covered by the project. The oral test is mandatory. Assessment breakdown: 75% project, plus 25% oral examination. The project consists in: (a) Definition of the architecture of a given application, 30% of total marks; (b) Definition of the multi-views models and architectural styles, 30% of total marks; (c) Specification of the architecture by using two modeling languages taught during the course, 40% of total marks. The oral test (max 1 hour) consists of one-two questions for each serious mistake in the project and two or three questions related to the contents not covered by the project. The final mark of the 12 CFU Course "Advanced Software Engineering " is obtained as the average among the marks of the 6 CFU Modules "Analysis and Testing of Component-based Systems" and "Software Engineering II".