COURSE DESCRIPTION – ACADEMIC YEAR 2017/2018

Course title	**Requirements and Design of Software Systems**
Course code	72016
Scientific sector	INF/01
Degree	Master in Computer Science (LM-18)
Semester	1
Year	1
Credits	8
Modular	No

Total lecturing hours	48
Total lab hours	--
Total exercise hours	24
Attendance	Not compulsory, but non-attending students have to contact the lecturer at the start of the course to agree on the modalities of the independent study.
Prerequisites	Basic courses in Programming and Software Engineering. Familiarity with UML and software modelling.
Course page	https://ole.unibz.it/

Specific educational objectives	The course belongs to the type "caratterizzanti – discipline informatiche" in the curriculum “Software and IT Management”. The course objective is to familiarize students with tools and techniques to acquire and analyze software requirements, and to define and design a software system. Emphasis is given to traceability of requirements to architecture, to justification of design decisions based on collected requirements, to consequences of design decisions. Students will acquire skills and competencies resulting from the conception, negotiation, documentation and maintenance of software requirements in a specific domain and environment. Requirements analysis aims at reviewing, assessing, prioritizing, and balancing the software requirements by developing technical specifications for building a system that will meet the needs of the stakeholders. Design of software systems aims at identifying or building software components that define the characteristics and quality of a system. The students are exposed to problem-solving techniques that allow the synthesis of software solutions satisfying the requirements of the system.

Lecturer	Claus Pahl
Contact	Piazza Domenicani 3, Room 1.11, Claus.Pahl@unibz.it
Scientific sector of lecturer	INF/01
Teaching language	English
Office hours	During the lecture times, and by arrangement by email. Faculty of CS, Piazza Domenicani 3, Office 1.11
Lecturing Assistant (if any)	TBD
Contact LA	TBD
Office hours LA	TBD
List of topics	• Requirements elicitation and specification

	• Requirements management • Traceability and validation of requirements • Quality measures of software requirements • Non-functional requirements • Architectural Patterns • Design Patterns
Teaching format	Frontal lectures, exercises; team projects.

Learning outcomes	Knowledge and understanding • Knowing foundations, techniques and methods for designing and implementing information systems, both basic software and application software, with special emphasis on engineering principles applied to conception, creation, development, test and maintenance of software systems • Being able to work with great autonomy, also taking responsibility of projects and structures Applying knowledge and understanding • Be able to apply engineering principles in different domains of different complexity, both those typically IT related and those non-IT related, where software technology has great relevance, such as, for example, in logistics and in medicine • Be able to define an innovative technical solution to an application problem, which respects constraints and requirements of technical, functional and organizational kind. • Be able to synthesize knowledge acquired from reading and studying scientific and technical documentation and develop reports and presentations. Making judgments • Be able to plan and re-plan a technical project activity, and to bring it to completion by meeting the defined deadlines and objectives • Be able to identify work goals, compatible with available time and resources. Communication skills • Be able to structure and prepare scientific and technical documentation describing project activities • Be able to interact and collaborate with peers and experts in the realization of a project or a research activity Ability to learn • Be able, in the context of a problem-solving activity, to extend even incomplete knowledge taking into account the final objectives of the project.

Assessment	• Project (50% of exam mark) • Final written exam (50% of exam mark)
Assessment language	English

Evaluation criteria and criteria for awarding marks	- Project assessment (50% of exam mark): project requirements and project design size, internal coherence and completeness of the documents, adequateness and mastery of techniques, assessed on the written document (this mark is the same for all group participants); awareness and personal contribution to group work, assessed at project presentation (this mark is individual, it confirms or decreases the group mark up to 1/3) - Final written exam (50% of exam mark): clarity of answers, knowledge of the proposed topics both as definitions and as examples, use of adequate technical terminology, ability to summarize, evaluate, and establish relationships among topics.

Required readings	The course will be based on lecture notes. Subject Librarian: David Gebhardi, David.Gebhardi@unibz.it
Supplementary readings	Selected chapters from the following ebooks (available with unibz credentials from the library website): - Lauesen, Software Requirements, Styles and techniques. Addison Wesley, available in the library, code ST 230. - Bass, Clements, Katzan, Software Architecture in practice, http://proquest.safaribooksonline.com/book/software-engineering-and-development/9780132942799 - Pattern oriented software architecture for dummies, http://proquest.safaribooksonline.com/book/information-technology-and-software-development/9781119963998 Open educational resources, representing alternative or supplementary materials, shall be linked to the course website.
Software used	Software Modelling (e.g. Argo UML, Papyrus, StarUML).