COURSE DESCRIPTION – ACADEMIC YEAR 2019/2020

Course title	**Verification and Reliability for Dependable Systems**
Course code	76058
Scientific sector	INF/01
Degree	Software Engineering for Information Systems (LM-18)
Semester	2
Year	1
Credits	6
Modular	No

Total lecturing hours	40
Total exercise hours	20
Attendance	Not compulsory
Prerequisites	Students are familiar with practices and methods of modern software product development and statistics and are able to develop software programs in autonomy. Pre-requisite material can be taught in the following courses: • Contemporary Software Development • Requirements and Design for Dependable Systems
Course page	https://ole.unibz.it/

Specific educational objectives	The course belongs to the type caratterizzanti – discipline informatiche. The course defines principles and practices of verification and reliability of software systems that have dependable characteristics. Verification methods aim at checking that the system meets prescribed software specifications. Reliability aims at observing and predicting the capability of a system to operates according to its specifications over a given period of time. The goal of the course is to prepare the students to recognize the quality characteristics of a system and to develop and maintain a system accordingly.

Lecturer	Barbara Russo
Contact	Piazza Domenicani 3, Room 1.16, barbara.russo@unibz.it, 0471-016170
Scientific sector of lecturer	INF/01
Teaching language	English
Office hours	By previous appointment via e-mail.
Lecturing Assistant (if any)	Matteo Camilli
Contact LA	Piazza Domenicani 3, Room 1.13, Matteo.Camilli@unibz.it
Office hours LA	By previous appointment via e-mail.
List of topics	• Dependable properties of systems • Software and software systems testing • Techniques for verification of software systems • Advances in test design and implementation • HW and SW reliability and their models
Teaching format	Frontal lectures, exercises, and project development solo or in team.

Learning outcomes	**Knowledge and understanding** D1.2 To be able to analyze and solve even complex problems in the area of Software Engineering for Information Systems with particular emphasis on the use of studies, methods, techniques and technologies of empirical evaluation; D1.3 To know in depth the scientific method of investigation applied to complex systems and innovative technologies that support information technology and its applications; D1.4 To know in depth the principles, structures and use of computer systems for the automation of information systems. **Applying knowledge and understanding** D2.1 To know how to apply the fundamentals of empirical analysis of ICT data to the construction of mathematical models for the evaluation and prediction of characteristics of applications and software systems; D2.2 To be able to design and perform experimental analyses of information systems in order to acquire measures related to their behaviour and to evaluate experimental hypotheses in different fields of application, such as business, industrial or research; D2.5 To be able to extend and modify in an original way an existing technical solution or a formal model taking into account changed conditions, requirements and evolution of the technology. **Making judgments** D3.1 To be able to autonomously select documentation from a variety of sources, including technical books, digital libraries, technical scientific journals, web portals or open source software and hardware tools; D3.5 To be able to work with large autonomy, also assuming responsibility for projects and structures. **Communication skills** D4.2 To be able to present the contents of a scientific/technical report to an audience, including non-specialists, at a fixed time; D4.3 To be able to structure and draft scientific and technical documentation describing project activities; D4.5 To be able to prepare and conduct technical presentations in English; D4.7 To be able to carry out research and projects in collaborative manner; D4.8 To be able to synthesise knowledge gained from reading and studying scientific documentation. **Learning skills** D5.3 In the context of a problem solving activity, to be able to extend knowledge, even if incomplete, taking into account the final objective of the project;

	D5.4 To be able to formulate and validate theories and define new methods through empirical induction and new generation scientific investigation tools.

Assessment	**Knowledge and understanding** The written exam contains exercises and theoretical open questions on verification and reliability. Specifically: 1. The students are required to describe a technique and interpret its use in a specific environment. 2. The students are requested to compare techniques and describe their benefits and limitations **Applying knowledge and understanding** At the lab, students will be evaluated through the project development on 1. Their autonomy and competence in selecting and applying testing techniques on actual software. 2. How they use statistical knowledge for fitting models on data and drawing conclusion out of the resulting findings. 3. How they can design and perform a study on reliability data. The midterm and the final written exam will contain exercises and theoretical open questions on verification techniques and reliability modelling. **Making judgments** At the lab, students will be evaluated on 1. The ability to apply the techniques for verification and reliability 2. The ability to devise a solution and interpret the results **Communication skills** At the written exam, students will be evaluated by the appropriate use of the course registry. At the lab, students will be evaluated on their ability to defend their conclusions. **Learning skills** At the written exam, students will be evaluated by the level of knowledge acquired on the topics of the course At the lab, students will be evaluated on their ability to apply and develop concepts and tools for verification and reliability of dependable software . The assessment is based on a project evaluation (50%) and a written exam (50%). The project work can be hand in no later than one week before the final exam date. To access the written exam students must have passed (18 or more) the project work. In case the project work assessment is positive but the final written exam is not positive, the assignments grade is valid for all three regular exam sessions.
Assessment language	English
Assessment typology	Monocratic commission

Evaluation criteria and criteria for awarding marks	*Final grade: 50% lab assessment + 50% written exam* *Lab assessment must be positive (i.e., 18 or higher) to access the written exam.* *Final grade pass: 18 or higher.* Relevant for the assessment: Lab assessment: ability to apply in autonomy and develop further instruments introduced during the lectures/labs and needed to accomplish tasks and perform little studies with data. Ability to report in a professional manner also using the appropriate terminology and concepts of the course. Written exam: ability to use the appropriate terminology and concepts of the course and to apply them in different context. Ability to understand the assumptions under which different techniques/methods can better perform or be used. Ability to analyze a problem and determine the causes. Ability to synthetize the results and interpret them in a specific context also using mathematical instruments to compare and evaluate models shaping software systems in testing or reliability.

Required readings	Lecture notes and papers will be handed out during the course. **Main reference for testing:** Pezzè & Young, Software Testing and Analysis: Process, Principles and Techniques, Wiley, 2007. University Shelf ST 233 P522 . Chap.1-4, 5-6 8-12 17 **Main reference for reliability:** Lyu, M. (ed.) Handbook of Software Reliability Engineering, IEEE Computer Society Press, 1996 Chapter on SRGM **Main reference for Dynamic Systems:** Rigdon E.S. and Basu A.P. Statistical methods for the reliability of repairable systems Wiley series in probability and statistics. Chapter 1-3 **Main reference to review statistic background** Baron, M. Probability and Statistics for computer Scientists Chapmall and Hall, ISBN: 1584886412 University shelf: 15 ST 340 B265(.07). Chapter 1-3 and chapter 6
Supplementary readings	• Laurie Williams et al. http://openseminar.org/se/modules/7/index/screen.do • Kent Beck: Test Driven Development by Example, Addison-Wesley Verlag
Software used	• R • Latex • Java • Software needed for project development (e.g., testing frameworks) In case is needed, students will develop their own tools to mine software reliability data