Syllabus

Course Description

Course Title	Verification and Reliability
Course Code	76096
Course Title Additional	
Scientific-Disciplinary Sector	INFO-01/A
Language	English
Degree Course	Master in Software Engineering
Other Degree Courses (Loaned)	
Lecturers	Prof. Barbara Russo, Barbara.Russo@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/2242
Teaching Assistant	
Semester	Second semester
Course Year/s	1
CP	6
Teaching Hours	40
Lab Hours	20
Individual Study Hours	90
Planned Office Hours	
Contents Summary	• Dependable systems • Verification by design • Search Based Software Testing • Software Reliability • Models of reliability growth • AI for Software verification and reliability.
Course Topics	The course focuses on key principles and practices for developing dependable software systems. It consists of three connected parts. In the first part of the course, we overview concepts and

	techniques to verify software systems, with particular focus on testing. We introduce advanced topics such as search based software testing, providing practical examples. The second part is dedicated to reliability with specific stress on modeling reliability growth as introduced by Musa and Okumoto. We overview the stochastic processes underlying them and introduces the mathematical instruments (e.g., Poisson processes and Markov Chain) to estimate time and number of failures of a system over time. The last part of the course focuses on AI-based techniques for software verification (e.g., defect prediction) and reliability (e.g., AutoML).
Keywords	Dependable Software Systems, Software Verification, Software Reliability, AI for Software Engineering.
Recommended Prerequisites	Students are familiar with practices and methods of modern software product development and probbility and statistics and are able to develop software programs in autonomy. Pre-requisite material can be taught in the following courses: • Continuous Integration and Delivery • Software design and implementation.
Propaedeutic Courses	
Teaching Format	Frontal lectures, exercises, and project development solo or in team.
Mandatory Attendance	Yes (75%).
Specific Educational Objectives and Learning Outcomes	Knowledge and understanding D1.2 be able to analyse and solve even complex problems in the area of Software Engineering with particular emphasis on the use of empirical evaluation studies, methods, techniques, and technologies. D1.3 have an in-depth knowledge of the scientific method of investigation applied to even complex systems and innovative technologies that support Software Engineering and its various fields of applications. Applying knowledge and understanding D2.1 know how to apply the fundamentals of data analysis for the construction of frameworks, models, techniques, and tools for the evaluation and prediction of characteristics of applications and software systems.

	D2.2 know how to design and carry out empirical studies of software systems in order to acquire measurements of their behaviour and evaluate experimental hypotheses in different application fields, such as business, industry, education, or research. D2.5 ability to extend and modify an existing technical solution or theoretical model in an original way, taking into account changing conditions, requirements and the evolution of technology. Making judgements D3.1 ability to independently select documentation from various sources, including technical books, digital libraries, technical scientific journals, web portals, or open source software and hardware tools. D3.5 ability to work with broad autonomy, taking responsibility for projects and structures. Communication skills D4.1 ability to present the contents of a scientific/technical report in a set time in front of diverse audiences, including non-specialists. D4.2 ability to structure and draft scientific and technical descriptive documentation of project activities for diverse audiences. D4.4 ability to prepare and deliver presentations with technical content in English for diverse audiences. D4.6 ability to carry out research and projects in a working group. Learning skills D5.2 ability to independently keep up to date with developments in the most important fields of information technology. D5.3 ability to extend incomplete knowledge with regard to the final objective of the project, in the context of a problem-solving activity.
Specific Educational Objectives and Learning Outcomes (additional info.)	The course belongs to the type "caratterizzanti – discipline informatiche". By the end of this course, students will be able to understand the foundational principles of dependable software systems and the key challenges involved in their development. They will be able to describe various software verification and testing techniques,

	including both conventional and advanced approaches. In addition, students will demonstrate knowledge of software reliability concepts and learn how reliability evolves over time. They will also be able to apply basic mathematical and statistical tools to analyze software failing behavior and reliability and explore the power of AI in software engineering tasks.
Assessment	Student evaluation in the Verification and Reliability course will include both lab assessments and a written exam. The lab assessment will measure students' ability to independently apply and extend empirical evaluation methods and technologies to solve complex software engineering problems (D1.2, D1.3), design and conduct empirical studies to gather and analyze software system measurements (D2.1, D2.2), and creatively modify existing models or solutions in response to evolving requirements (D2.5). Students must demonstrate autonomy and responsibility in managing project tasks (D3.5) and independently select appropriate technical documentation and resources (D3.1). The written exam will assess proficiency in communicating technical content effectively to diverse audiences through well-structured scientific documentation and presentations (D4.1, D4.2, D4.4), as well as the ability to collaborate in research projects within teams (D4.6). Additionally, students will be evaluated on their capacity to stay current with advances in information technology (D5.2) and to extend incomplete knowledge critically in problem-solving contexts (D5.3).
Evaluation Criteria	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.
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	Kent Beck: Test Driven Development by Example, Addison-Wesley Verlag Taehyoun Kim, Duksan Ryu, and Jongmoon Baik, (2024) Automated Machine Learning for Enhanced Software Reliability Growth Modeling : A Comparative Analysis with Traditional SRGMs QRS.
Further Information	
Sustainable Development Goals (SDGs)	Responsible consumption and production, Industry, innovation and infrastructure