Syllabus

Course Description

Course Title	Responsible Software Engineering
Course Code	76110
Course Title Additional	
Scientific-Disciplinary Sector	IINF-05/A
Language	English
Degree Course	Master in Software Engineering
Other Degree Courses (Loaned)	
Lecturers	Dott. Alma Sheko, Alma.Sheko@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/55199
Teaching Assistant	
Semester	First semester
Course Year/s	2
CP	6
Teaching Hours	40
Lab Hours	20
Individual Study Hours	90
Planned Office Hours	18
Contents Summary	- Engineering Challenges in Responsible Software Systems - Software Safety Engineering and Risk Analysis - Fairness, Bias, and Algorithmic Accountability - Sustainable and Energy-Efficient Software Engineering - Explainability, Transparency, and Human Oversight - Privacy Engineering and Secure Software Design
Course Topics	The course covers the principles and practices of responsible software engineering throughout the software lifecycle. Specific topics include: - Responsible software engineering principles and the broader

	societal impact of software systems - Stakeholder identification, responsible requirements engineering, and trade-offs between technical and societal requirements - Software safety concepts, hazards, risk identification, risk assessment, and mitigation strategies - Fairness in software and AI-based systems, sources of bias, fairness metrics, and approaches to bias detection and mitigation - Algorithmic accountability and responsibilities of software engineers and organizations - Environmental impact of software systems, green software principles, energy-efficiency measurement, and sustainable software design - Explainability and transparency in automated and AI-supported systems - Human oversight, human-in-the-loop approaches, and responsible decision-making - Privacy by design and data protection principles in software development - Secure software design, security risks, and integration of security throughout the software development lifecycle - Analysis of real-world cases involving safety, fairness, privacy, security, sustainability, and accountability - Application of responsible software engineering principles through practical exercises and a course project
Keywords	Responsible Software Engineering; Software Safety; Risk Analysis; Fairness; Algorithmic Bias; Algorithmic Accountability; Sustainable Software Engineering; Green Software; Explainability; Transparency; Human Oversight; Privacy Engineering; Secure Software Design
Recommended Prerequisites	Basic knowledge of software engineering principles and the software development lifecycle is recommended. Familiarity with software requirements, design, and programming concepts is helpful but not mandatory.
Propaedeutic Courses	
Teaching Format	The course is delivered through in-person lectures and practical exercises. Lectures introduce the theoretical concepts and principles of responsible software engineering, supported by real-world examples and case studies. Practical exercises focus on case-study analysis, discussion, problem-solving activities, and the

	application of responsible software engineering principles to concrete scenarios. Students will also develop a course project to apply the concepts covered during the course.
Mandatory Attendance	Attendance is not mandatory but strongly recommended. Non-attending students should contact the lecturer at the start of the course to agree on the modalities of the independent study.
Specific Educational Objectives and Learning Outcomes	Knowledge and understanding D1.5 know the fundamentals, techniques, and methods of design, customisation and implementation of software to support the automation of new-generation software systems for industrial production, company business, education, and society. D1.7 know the various fields of application of Software Engineering also with reference to the local, national, and international economic-social context. D1.8 ability to read, understand, and elaborate on specialist scientific documentation, such as conference proceedings, articles in scientific journals, technical manuals. Applying knowledge and understanding 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. Making judgements D3.6 ability to identify the various roles of software engineering in society and its social and environmental impact on society. Communication skills D4.4 ability to prepare and deliver presentations with technical content in English for diverse audiences. Learning skills D5.1 ability to independently extend the knowledge acquired during the course of study by reading and understanding scientific and technical documentation in English.
Specific Educational Objectives and Learning	

Outcomes (additional info.)	
Assessment	The assessment consists of a final written exam (60% of the final grade) and a course project (40% of the final grade). The written exam assesses students' understanding of the main concepts and principles of Responsible Software Engineering and their ability to analyse problems and real-world scenarios from a responsible software engineering perspective. The course project requires students to apply the principles and methods covered during the course to a concrete software system or case study, addressing relevant aspects such as safety, fairness, sustainability, transparency, privacy, security, and accountability. The same assessment methods apply to attending and non-attending students.
Evaluation Criteria	Written exam (60%): evaluation is based on the correctness and completeness of the answers, understanding of the key concepts and principles, appropriate use of terminology, and the ability to analyze responsible software engineering problems and justify proposed solutions. Course project (40%): evaluation is based on the appropriate application of responsible software engineering principles, quality and depth of the analysis, identification of relevant risks and trade-offs, justification of design decisions and proposed solutions, and clarity and overall quality of the project work.
Required Readings	Barrett, D. J. (2024). *Responsible Software Engineering*. O'Reilly Media. O'Regan, G. (2025). *Ethical and Legal Aspects of Computing: A Professional Perspective from Software Engineering*. Springer. Selected articles, case studies, and lecture materials will also be provided during the course.
Supplementary Readings	Curri, A. (2024). *Building Green Software: A Sustainable Approach to Software Development and Operations* (1st ed.). O'Reilly Media.
Further Information	Lecture slides, selected articles, case studies, and additional

	learning materials will be made available to students through the university's online learning platform.
Sustainable Development Goals (SDGs)	Responsible consumption and production, Industry, innovation and infrastructure