COURSE DESCRIPTION – ACADEMIC YEAR 2023/2024

Course title	**Contemporary Software Development**
Course code	76051
Scientific sector	INF/01
Degree	Master in Software Engineering (LM-18)
Semester	1
Year	1
Credits	6
Modular	No

Total lecturing hours	40
Total exercise hours	20
Attendance	Not compulsory but strongly suggested. Non-attending students must contact the lecturer at the start of the course to agree on the modalities of the independent study.
Prerequisites	Students should have a good knowledge of software development in an object-oriented programming language, possess knowledge of software development processes as well as Agile methodologies.
Course page	https://ole.unibz.it/

Specific educational objectives	The course type is "attività formativa caratterizzante" and belongs to the subject area "informatica". The course is designed to give specific professional skills. Students will learn how to apply software development techniques and tools that are used in contemporary software engineering environments. These include virtualization and container infrastructures, microservices, the automation of the software engineering process using DevOps, continuous integration/deployment/delivery, log and configuration management and monitoring.

Lecturer	Andrea Janes
Contact LA	andrea.janes@unibz.it
Scientific sector of lecturer	/
Teaching language	English
Office hours	During the lecture time span, Monday or Friday 13:00-14:00, arrange beforehand by email.
Lecturing Assistant (if any)	Same as lecturer
Contact LA	/
Office hours LA	/
List of topics	• Software development environments • Configuration management • Software artifact management • Design and programming techniques in practice • Tools and techniques for process management and quality assurance • Continuous integration
Teaching format	Frontal lectures and laboratory exercises

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 information systems for industrial production and company business; **Applying knowledge and understanding** • D2.2 know how to design and carry out experimental analyses of software systems in order to acquire measurements of their behaviour and evaluate experimental hypotheses in different application fields, such as business, industry or research; • D2.4 ability to define an innovative technical solution to an application problem that respects technical, functional and organisational constraints and requirements; **Making judgments** • D3.2 ability to plan and re-plan a technical project activity and to carry it out within the defined deadlines and objectives; **Communication skills** • D4.3 ability to co-ordinate the work of a project team and to identify activities aimed at achieving the project objectives; **Learning skills:** • D5.2 ability to independently keep up to date with developments in the most important fields of information technology.

Assessment	Assessment is based on the final exam. To allow students to test and learn the presented technologies and methods during the semester, exercises are offered and discussed in the lab. The final exam will assess the understanding of the theoretical concepts as well as the ability to solve the problems discussed in the lab. Both, attending and non-attending students will be assessed through the final exam. Also, both, attending and non-attending students can download the optional weekly assignments from the course web page.
Assessment language	English
Assessment typology	Monocratic
Evaluation criteria and criteria for awarding marks	For both, attending and non-attending students, the assessment is based on the final exam (up to 30 points). The final mark is the obtained final exam score. Relevant for assessment is the solution of the given task and the ability to explain the adopted strategy to reach the solution as well as the clarity of answers, mastery of language, ability to summarize, evaluate, and establish relationships between topics.

Required readings	Lecture notes will be handed out during the course. Subject Librarian: David Gebhardi, David.Gebhardi@unibz.it

Supplementary readings	• Robert C Martin: Clean Architecture: A Craftsman's Guide to Software Structure and Design. Pearson (2017) • Vaughn Vernon: Domain-Driven Design Distilled. Addison-Wesley Professional (2016)
Software used	For the approaches discussed in the lectures, the lab introduces tools that support these topics. The software used includes Microsoft Visual studio Code, GIT, Docker, Maven, npm, locust, Grafana, SonarQube. The provided examples are written in Java and Python.