

COURSE DESCRIPTION – ACADEMIC YEAR 2023/2024

Course title	Software Quality and Metrics
Course code	76103
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.
Course page	https://ole.unibz.it/
Specific educational objectives	<p>The type of course is "attività formativa affine o integrativa" and belongs to the subject area "informatica".</p> <p>The goal of the course is to familiarize students with advanced tools and techniques for developing a software-based system, based on a systematic measurement process. During the course, students will be introduced to the measurement of software artifacts, processes, and resources based on the representational theory of measurement, will cover the definition and interpretation of quality in software development, and will learn methods and tools for determining and ensuring quality.</p>
Lecturer	Andrea Janes
Contact LA	andrea.janes@unibz.it
Scientific sector of lecturer	INF/01
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	<ul style="list-style-type: none"> • Importance of quality • Software product quality • Software process quality • Software quality metrics • Measurement techniques for monitoring software quality • Total Quality Management tools and techniques and their application to software
Teaching format	Frontal lectures and laboratory exercises

<p>Learning outcomes</p>	<p>Knowledge and understanding D1.1 possess solid knowledge of both the fundamentals and the application aspects of the various fundamental areas of computer science;</p> <p>Applying knowledge and understanding D2.1 know how to apply the fundamentals of empirical analysis of ICT data for the construction of mathematical models for the evaluation and prediction of characteristics of applications and software systems;</p> <p>Making judgments D3.3 ability to define work objectives compatible with the available time and resources.</p>
<p>Assessment</p>	<p>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.</p> <p>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.</p>
<p>Assessment language</p>	<p>English</p>
<p>Assessment typology</p>	<p>Monocratic</p>
<p>Evaluation criteria and criteria for awarding marks</p>	<p>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.</p> <p>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.</p>
<p>Required readings</p>	<p>Lecture notes will be handed out during the course.</p> <p>Subject Librarian: David Gebhardi, David.Gebhardi@unibz.it</p>
<p>Supplementary readings</p>	<ul style="list-style-type: none"> Norman Fenton, James Bieman: Software Metrics - A Rigorous and Practical Approach, Third Edition. Routledge, 2015
<p>Software used</p>	<p>For the approaches discussed in the lectures, the lab introduces tools that support these topics. The software used includes Microsoft Visual studio Code, Jupyter Notebook, locust, Grafana, SonarQube. The provided examples are written in Java and Python.</p>