

## COURSE DESCRIPTION – ACADEMIC YEAR 2024/2025

<b>Course title</b>	<b>Software Quality and Metrics</b>
<b>Course code</b>	76103
<b>Scientific sector</b>	INF/01
<b>Degree</b>	Master in Software Engineering (LM-18)
<b>Semester</b>	1
<b>Year</b>	1
<b>Credits</b>	6
<b>Modular</b>	No
<b>Total lecturing hours</b>	40
<b>Total exercise hours</b>	20
<b>Attendance</b>	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.
<b>Prerequisites</b>	Students should have a good knowledge of software development in an object-oriented programming language.
<b>Course page</b>	<a href="https://ole.unibz.it/">https://ole.unibz.it/</a>
<b>Specific educational objectives</b>	<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>
<b>Lecturer</b>	<a href="#">Usman Rafiq</a>
<b>Contact LA</b>	Ser I 1.07, <a href="mailto:usman.rafiq@unibz.it">usman.rafiq@unibz.it</a>
<b>Scientific sector of lecturer</b>	INF/01
<b>Teaching language</b>	English
<b>Office hours</b>	During the lecture hours or every Friday from 16:00 to 18:00, please arrange any necessary meetings or discussions in advance via email.
<b>Lecturing Assistant (if any)</b>	Same as lecturer
<b>Contact LA</b>	/
<b>Office hours LA</b>	/
<b>List of topics</b>	<ul style="list-style-type: none"> <li>• Importance of quality</li> <li>• Software product quality</li> <li>• Software process quality</li> <li>• Software quality metrics 5</li> <li>• Measurement techniques for monitoring software quality</li> <li>• Total Quality Management tools and techniques and their application to software</li> </ul>
<b>Teaching format</b>	Frontal lectures and laboratory exercises

<p><b>Learning outcomes</b></p>	<p><b>Knowledge and understanding</b></p> <ul style="list-style-type: none"> <li>D1.1 possess solid knowledge of both the fundamentals and the application aspects of the various fundamental areas of computer science;</li> </ul> <p><b>Applying knowledge and understanding</b></p> <ul style="list-style-type: none"> <li>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;</li> </ul> <p><b>Making judgments</b></p> <ul style="list-style-type: none"> <li>D3.3 ability to define work objectives compatible with the available time and resources.</li> </ul>
<p><b>Assessment</b></p>	<p>Course assessment is based on the written exam, an assignment, and the semester project. The final grade will be based on the average of all these components. Assignments and the project will provide students with the opportunity to practice the knowledge they have acquired, while the final exam will assess their understanding of theoretical concepts as well as their ability to solve problems discussed in lectures and labs. Both attending and non-attending students will be assessed through these assessment components.</p> <p>Note: Passing both the assignment and semester project will be mandatory to attend the final exam. Both of these modules will contribute to all three regular exam sessions.</p>
<p><b>Assessment language</b></p>	<p>English</p>
<p><b>Assessment typology</b></p>	<p>Monocratic</p>
<p><b>Evaluation criteria and criteria for awarding marks</b></p>	<p>For both, attending and non-attending students, the assessment is based on three modules (assignment, project and final exam). The final mark is thus obtained through calculating the average of these three components.</p> <p>Other criteria include effective application of software quality and metrics, good teamwork (particularly for project), clarity of answers, mastery of language, ability to summarize, evaluate, and establish relationships between different concepts.</p>
<p><b>Required readings</b></p>	<p>Lecture notes will be handed out during the course. Research papers may be handed in if needed.</p> <p>Subject Librarian: David Gebhardi, <a href="mailto:David.Gebhardi@unibz.it">David.Gebhardi@unibz.it</a></p>
<p><b>Supplementary readings</b></p>	<p>Software Metrics - A Rigorous &amp; Practical Approach, 3rd edition          Authors: N. E. Fenton, J. Bieman, Publisher: CRC Press, Published: 2015, ISBN: 978-1-4398-3822-8</p> <p>Other reference literatures:</p>

	<ul style="list-style-type: none"><li>- Software Measurement Establish - Extract - Evaluate - Execute, Authors: Christof Ebert, Reiner Dumke, Publisher: Springer, Published: 2007</li><li>- Metrics and Models in Software Quality Engineering, 2nd edition, Author: Stephan H. Kan, Publisher: Addison-Wesley Publishing Company, Published: 2002, ISBN: 0201729156.</li></ul>
<b>Software used</b>	For the approaches discussed in the lectures, the lab introduces tools that support these topics. The tools include Grafana, Visual Studio Code, Jupyter Notebook, Locust and SonarQube. The tool usage also depends on the chosen project or team.